PLATE NUMBER RECOGNITION SYSTEMS BASED ON A CONTOURS AND CHARACTER RECOGNITION APPROACH

Chua Jing Yi[1], Rechard Lee[2], Prof. Madya Dr. Abdullah Bade[3]

Faculty of Science and Natural Resources, University Malaysia Sabah,
Kota Kinabalu, Sabah, Malaysia.

*Corresponding author : [1]jingyi5991@gmail.com, [2]rechard@ums.edu.my, [3]abade08@yahoo.com

ABSTRACT. *License plate recognition system (LPR) plays an important role in intelligent traffic control system. However, most of the existing LPR are complex and hard to implement. The aim of this project is to improve the LPR techniques in terms of speed and accuracy by applying the Connected Component Analysis (CCA) and K-Nearest Neighbour algorithm (KNN). The LPR is divided into three stages which are image pre-processing, character segmentation, and character recognition. First, the input plate image will undergo some image property functions such as omission of noise to enhance the quality of the image. The CCA is applied to segment the characters by drawing rectangle boxes on each character, based on contours to extract the characters into smaller images. These images are then used as query images in character recognition stage. The images are fed to a pre-defined KNN classifier to determine the features of each image and to identify them. Five experiments were carried out to validate the proposed system. Ten Malaysia single row plate images and two foreign plate images were used as the input images on these tests. The findings show that the proposed system has an 80.0% success rate in segmentation, 92.21% accuracy rate in recognition, the optimal K value is 1, and the input image must be in a single row and comprises of a black background and white characters namely letters and digits. In conclusion, a prototype for plate number recognition has been developed with a high success rate in segmentation and a high accuracy in character recognition. Suggested future studies include a focus on segmenting double row license plates and recognizing similar characters.*

KEYWORDS. License Plate Recognition, Malaysia license plate surface, Connected Component Analysis, character recognition, K-Nearest neighbour

INTRODUCTION

The immense increment of vehicle ownership has led to the rapid technical growth in the field of computer image processing, which increases the need for an efficient and affordable security system. This has resulted in the evolution of various security methods based on computer image analysis. The *License Plate Recognition* system (LPR) is very useful for large-volume surveillance. However, most of the existing LPR systems are costly and hard to implement [1]. LPR system can be divided into three phases which are image pre-processing, character segmentation, and character recognition. The first phase is to preserve relevant details and to removes irrelevant objects [2] because the noise on the image captured on camera will affect the accuracy of the recognition system [3]. Character segmentation is applied on license plate to remove unnecessary data and to maintain the characters (i.e. letters and numbers) on the license plate. This is achieved by matching the width height ratios with the contours that are detected on the plate images [4]. As for the final phase, many character recognition techniques have been introduced by researchers and scientists. Feature extraction identifies characters based on aspect ratio, concentration of pixels, and number of strokes [5]. Besides that, template matching matches the segmented characters from the image input with the templates stored in a database using cross-correlation method. The best match between an extracted image and templates has the highest cross-correlation coefficient [6]. Hence, several approaches are implemented to develop a simple prototype with high accuracy.

METHODOLOGY

The proposed system design is illustrated in Figure. 1.

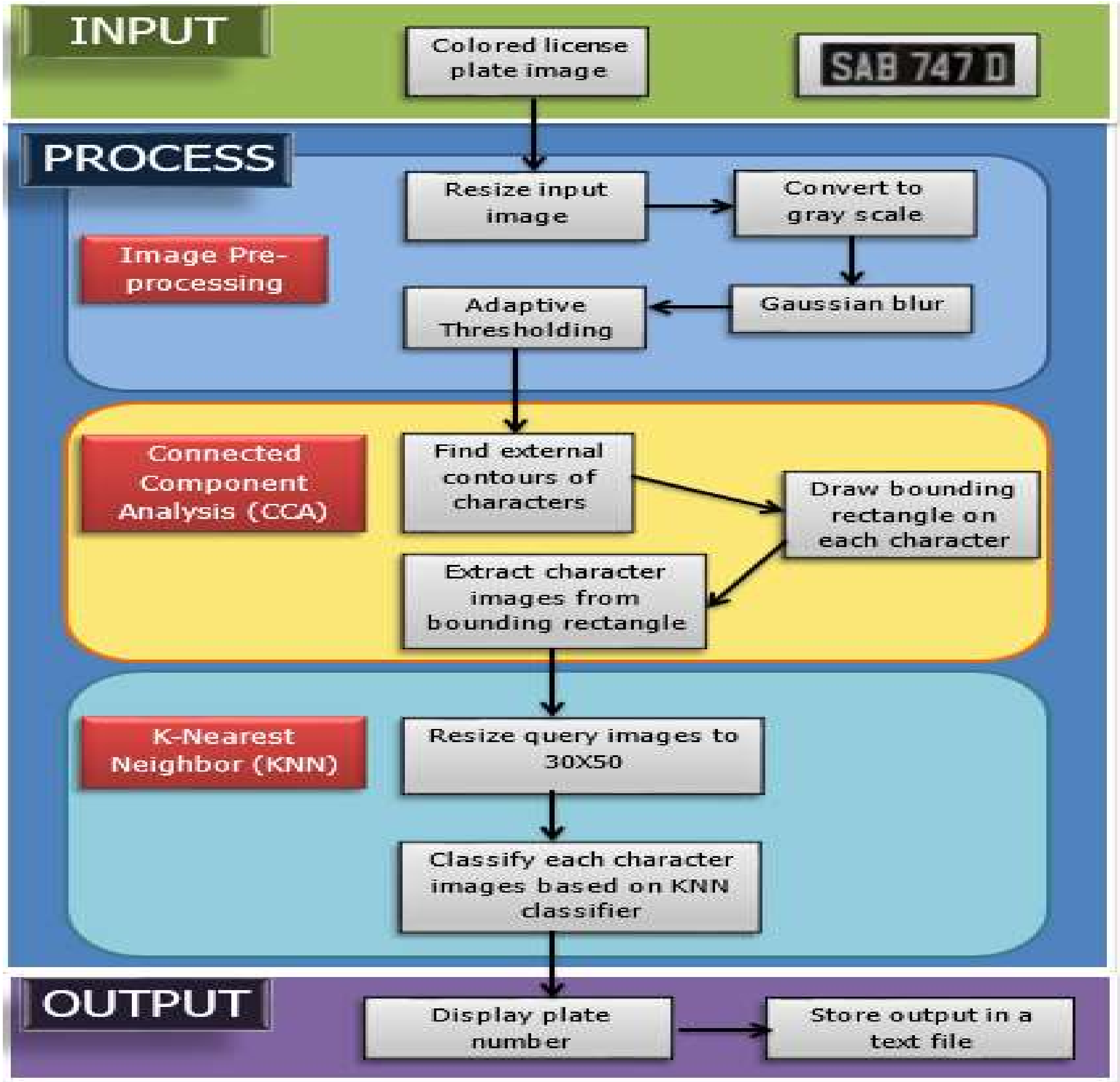


Figure. 1. Overview of Architecture

As shown in Fig. 1 above, the input is a coloured license plate number image that is extracted from a vehicle captured by a camera. There are three stages in developing the license plate recognition system. The first stage is the image pre-processing. In this stage, the coloured input image is resized to the desired image width and height. Next, the image is converted into a grayscale image and the Gaussian blur is applied to reduce noise on the image. After the image is de-noised, adaptive threshold is applied to convert the grayscale image into binary image.

The next stage involves the character segmentation using CCA. First, all the contours are determined in the binary image. This step highlights the external contour of the characters. A bounding rectangle is drawn on each character to extract the character images. These images are then used as the query images for character recognition. The final stage is the character recognition based on the K-Nearest Neighbour (KNN) algorithm. A KNN classifier utilises ten training images (refer to TABLE I). The extracted images from the segmentation stage are assigned to different character classes based on the KNN classifier. Then, the output from the character recognition is displayed in a text form. For storage purposes, a text file is used to store the outputs from the system. The details of the implementation of the process stage are briefly described in the following section.

A. ***Pre-processing***

The algorithm in Fig. 2 is used to remove unnecessary objects in the input image and to enhance the quality of the image. For this research, the following four steps adopts the Grayscale, Gaussian Blur, and Adaptive Threshold.

Step 1: Resize image to 130X27 pixels

Step 2: Convert into gray scale

Step 3: Apply Gaussian blur

Step 4: Set threshold

Figure. 2. Algorithm of image pre-processing

B. ***Connected Component Analysis (CCA)***

The CCA scans and labels the pixels of binary image into components based on pixel connectivity. First, a random pixel in the character is identified then followed by the neighbouring pixels along the edge of the character. Since the edge of a character is a closed path, there is always a chance of returning to the first pixel after visiting all the other pixels along the edge. By using this method, the boundaries of each character are determined and enclosed within the rectangle boxes. The areas inside the boxes is considered the *region of interest* (ROI) and is extracted into several character images for further processing.

C. ***K-Nearest Neighbour (KNN)***

KNN is a template-based method that creates its own database characters using training images. The KNN classifier is trained to classify different types of uppercase letters and digits. A Malaysia license plate is used as the training image in this project (refer to Figure 3).

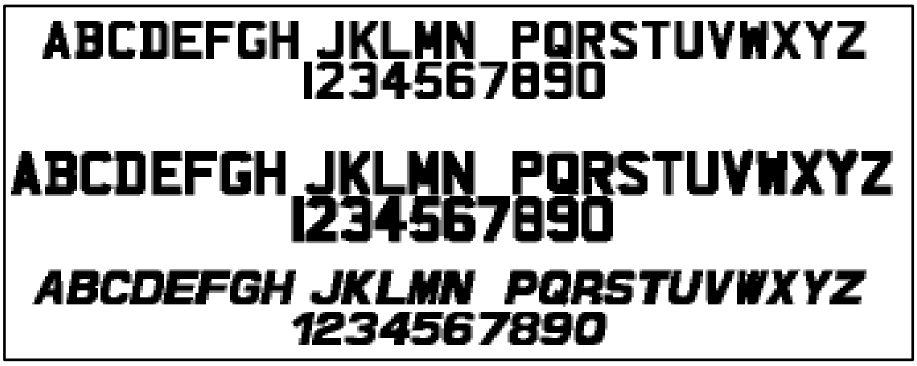


Figure 3: Training image for KNN

The letters 'I' and 'O' are omitted from Malaysia licence plates due to their similarities with the digits 1 and 0. The training data were stored in a feature space, a dimensional vector of numerical features that represents a certain object. These objects have their respective position in the feature space based on their features.

First, the query image goes through a size normalization step to resize the image to 20X30 pixels. Feature extraction is applied to extract the feature vector of the query image. The character of the license plate is then classified based on the feature vector and a proper K value.

IMPLEMENTATION

In Figure 4 below, the implementation begins by obtaining the coloured input image. The image is resized to 130X27 pixels and converted to grayscale image by using the following mathematical function:

Grey = (Red * 0.299 + Green * 0.587 + Blue * 0.114)

Next, the Gaussian blur is applied to reduce noise on the converted image by using the following mathematical formula:

$$\boldsymbol{G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{x^2}{2\sigma^2}}}$$

Where σ is the standard deviation of the function. A convolution matrix is formed by computing the values of the Gaussian function. The matrix is then applied in the image and the new value of each pixel is set to a weighted average of neighbouring pixel. The highest weight is assigned to the original pixels and the smaller weight is assigned to the neighbouring pixels.

The blurred image is converted to a binary image using adaptive threshold and the following mathematical formula:

$$\boldsymbol{b(x,y) = \begin{cases} 0 & if\ I(x,y) \leq T(x,y) \\ 1 & otherwise \end{cases}}$$

where b(x,y) is the binary image, I(x,y) € [0,1] is the intensity of a pixel at location (x,y) of the image I, and T(x,y) is the threshold value at location (x,y).

The next step is to identify whether a contour contains characters or not. If a contour area is smaller than 50 pixels, the contour is invalid and will not be taken into account. If a contour area is larger than 50 pixels, the contour is valid and will be added to list. The valid contours are sorted from left to right.

The character recognition starts by drawing a green rectangle on each valid contour and acquiring the ROI images of the rectangles. The images are resized to 20X30 pixels to ensure the consistency for recognition and storage. To call for the function of KNN, it is necessary to convert the ROI images into float. With the K value set at 1, we can find the nearest neighbour for each character and obtain the plate number.

```
Begin
Train and load KNN data
      Read input image
      //preprocessing
      Resize image to 130X27 pixels
      Convert into gray scale
      Apply Gaussian Blur
      Set threshold
            Adaptive threshold
      Clone image
      //segmentation
      Int i
            FOR i=0 and i smaller than contour.size
                  Contour[i]
                  Draw bounding rectangle
                  Float fltArea = contourArea
                  IF fltArea smaller than 50
                        Return false
                  Else
                        Return true
                        Sort contours from left to right
            //recognition
            FOR i=0 and i smaller than valid contour.size
                        Draw green rectangle on valid contour
                        Get region of interest
                        Resize to 20X30 pixels
                        Convert to float
                        Set K value to 1
                        Find nearest neighbor
                        Get plate number
                  ENDFOR
            ENDFOR
END
```

Figure 4: Pseudo-code for the license plate recognition system

EXPERIMENTAL RESULTS

1. *Segmentation test*

Ten different license plate images were used as the input images to test the segmentation success rate. These images were acquired from a smart phone with 4160X3120 image resolutions. Table I shows that most of the characters on the license plates were segmented correctly, as indicated by the green rectangles. The success rate of the CCA in character segmentation was 80.0%.

TABLE I: Result of segmentation test

Input image	Segmentation	Result
SAA7287 Y	SAA7287 Y	Yes
SAB 4603 D	SAB 4603 D	Yes
SAA 7239 W	SAA 7239 W	Yes
SAC 6890 C	SAC 6890 C	No
SAA 7779 R	SAA 7779 R	Yes
SAB 747 D	SAB 747 D	Yes
SS 8339 X	SS 8339 X	Yes
SAB 7026 G	SAB 7026 G	Yes
SAB 8924 P	SAB 8924 P	Yes
JLB 6738	JLB 6738	No

2. *Accuracy test*

Ten different license plate images were used as the input images to test the accuracy of the character recognition using KNN. The results are shown in Table II. Based on the results, the total number of characters was 77 and the total number of recognized characters was 71. Hence, the accuracy rate of KNN acquired from this test was 92.21%.

Table II: Result of accuracy test

Expected Output	Actual Output	Number of characters	Recognized characters
SAA7287Y	SAA7287Y	8	8
SAB4603D	SA8A603D	8	6
SAA7239H	SAA7239H	8	8
SAC6890C	1SAC6890C	8	8
SAA7779R	SAA7779R	8	8
SAB747D	SAB747D	7	7
SS8339X	SS8339X	7	7
SAB7026G	SA87D26G	8	6
SAB8924P	SAB6924P	8	7
JLB6738	1JL867381	7	6

3. *KNN classifier test*

Ten single-row license plate images were used as the input images in the proposed system with different K values. The results are shown in TABLE III. The optimal K value is 1, based on its highest recognition rate at 92.1%.

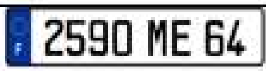
TABLE III: Result of KNN classifier test

K value	Recognition rate (%)
1	92.21
2	84.42
3	81.82
4	72.73
5	75.32
6	72.73
7	66.23
8	57.14
9	57.14
10	61.04

4. *Foreign car licence plate test*

To determine the applicability of the proposed system on other countries, foreign license plate images from France and Singapore were used as the input images. The results shown in TABLE IV shows that the proposed system is not able to recognize the licence plate number. This is due to the high contrast between the background colour and the coloured character used in France license plate.

Table IV: Result of foreign plate test

Country	Input	Result
France		Negative
Singapore		Positive

5. Double-row licence plate number test

To determine the applicability of the proposed system in double-row license plates, two double-row license plate images were captured and used as input images. The results are shown in Table V. Based on the results, both images produced negative results. Hence, the system failed to segment the characters when the input plate image contains two rows of characters.

TABLE V: Result of double row plate test

Input images	Result
	Negative
	Negative

CONCLUSION

Based on these experiments, we conclude that:

- The CCA method works well in character segmentation. However, some of the green boxes are located wrongly. This is due to the failure to eliminate the unwanted regions when determining the valid contours. In order to improve the segmentation method, the requirements for valid contours could be more specific to filter more unnecessary components.
- The KNN algorithm performs well in character recognition. However, the similarity between characters may confuse the KNN classifier while classifying each character based on their features. For example, the letter 'B' looks similar with the digit '8', the letter 'H' with the letter 'W', and the letter 'D' with digit '0'. To eliminate false recognition, the training datasets could include more font types of license plate characters so that the KNN classifier can function better.
- The KNN classifier score the highest recognition rate when the K value is set at 1. This means the KNN classifier chooses only the nearest class that has similar features with the query character and assigns the query character to that class.

- The results from foreign plate test indicate that the proposed system can be applied on licence plates with black background and white characters only.
- The results from double-row licence plate test indicate only single-row license plate is valid as the input image for the proposed system.

REFERENCES

Anagnostopoulos, C. N. E., Anagnostopoulos, I. E., Loumos, V., & Kayafas, E. 2006. A License Plate-recognition Algorithm for Intelligent Transportation System Applications. *IEEE Transactions on Intelligent transportation systems,* 7(3) :377-392.

Ghaffari, B. V., Kitajima, T., & Kouhnavard, M. 2013. A New Localization Method for Malaysian License Plate. *GSTF Journal on Computing (JoC)*, 2(4): 121-124.

Mythili, C., & Kavitha, V. 2011. Efficient Technique for Color Image Noise Reduction. *The research bulletin of Jordan ACM*, 11(11):41-44.

Jain, P., Chopra, N., & Gupta, V. (2014). Automatic License Plate Recognition using OpenCV. *International Journal of Computer Applications Technology and Research*, 3(12,756-61).

Dedgaonkar, S. G., Chandavale, A. A., & Sapkal, A. M. 2012. Survey of Methods for Character Recognition. *International Journal of Engineering and Innovative Technology (IJEIT)*, 1(5):180-189.

Rasheed, S., Naeem, A., & Ishaq, O. 2012. Automated Number Plate Recognition using Hough Lines and Template Matching. *In Proceedings of the World Congress on Engineering and Computer Science*, Vol. 1:24-26.