Determining adaptive thresholds for image segmentation for a license plate recognition system

Siti Norul Huda Sheikh Abdullah  
Center for Artificial Intelligence and Technology (CAIT), Faculty of Information System and Technology, Universiti Kebangsaan Malaysia, 43600 Selangor, Malaysia

Khairuddin Omar  
Center for Artificial Intelligence and Technology (CAIT), Faculty of Information System and Technology, Universiti Kebangsaan Malaysia, 43600 Selangor, Malaysia

Abbas Salimi Zaini  
Center for Artificial Intelligence and Technology (CAIT), Faculty of Information System and Technology, Universiti Kebangsaan Malaysia, 43600 Selangor, Malaysia

Maria Petrou  
Informatic and Telematics Institute (ITI), Center for Research and Technology Hellas, 6th km Charilaou - Thessaloniki road, P.O.Box 60361, GR-57001 Thessaloniki, Greece

Marzuki Khalid  
Centre for Artificial Intelligence and Robotics (CAIRO), Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Jalan Semarak,54100 Kuala Lumpur

Abstract—A vehicle license plate recognition (LPR) system is useful to many applications, such as entrance admission, security, parking control, airport and cargo, traffic and speed control. This paper describes an adaptive threshold for image segmentation applied to a system for Malaysian intelligent license plate recognition (MyiLPR). Due to the different types of license plates used, the requirements of an automatic LPR system are rather different for each country. Upon receiving the input car image, this system (MyiLPR) detects and segments the license plate based on proposed adaptive threshold via image and blob histogram, and blob agglomeration, and finally, it extracts geometric character features and classifies them using neural network. The use of the proposed adaptive threshold increased the detection, segmentation and recognition rate to 99%, 94.98% and 71.08% obtained with the fixed threshold used in the originally proposed system.

Keywords—adaptive threshold; image segmentation; license plate recognition; neural network; computer surveillance;

I. INTRODUCTION

Automatic license plate recognition (LPR) is an important research subject due to its many applications. For local authorities, LPR is required for the purposes of law enforcement, border protection, vehicle thefts, automatic toll collection and perhaps traffic control. For others, automatic LPR can be applied to gain access into housing areas, automatic parking control and as a marketing tool in large shopping complexes, as well as for surveillance.

LPR consists of five major steps: image capturing, image processing, segmentation, feature extraction and pattern classification. Usually, an LPR system includes targeted functions and specifications, such as a fast alphanumeric character recognition system. In addition, 24 hours non-stop operation is expected and the ability to issue alarm messages which are sent out after the recognition phase.

In Malaysia, perhaps 95% of the vehicles’ license plates are in the form of a single or a double line with normal fonts (Figure 1(a)). The other 5% are made of special fonts, such as those shown in Figure 1(b). The halo or illumination and unstandardized Malaysian plate issues have made license plate recognition continues to face big challenges [5]–[8]. On 20th January 2015, the signing ceremony for memorandum of understanding (MoU) took place.

Fig. 1: (a) Samples of common license plates and (b) Samples of special license plates in Malaysia.

An important issue in image preprocessing for a real time application, like license plate recognition, is the selection of the best threshold value for image binarization. Choosing the wrong threshold can cause failure in either the detection or the segmentation process, as the thresholded image could either totally hide the required objects or cause distraction in the region of interest due to variable illumination. Several researchers have proposed different ways for selecting the best threshold. For example Yoshimori et al. [1] used a real-coded genetic algorithm and a so called hereditary threshold determination method, Kim et al. [2] used an enhanced neural network, and Lee et al. [3] used adaptive local binarization. Another technique referred to as the Otsu automatic threshold [4] has also been extensively used in LPR and other applications.

www.ijacsa.thesai.org 510
of agreement (MoA) between UKM Tech (M) Sdn. Bhd. and Synergy Saver (M) Sdn. Bhd. was held successfully. According to Synergy Saver CEO Mr. Shariffuddin Basiron;

Up to date, the license plate recognition technology for Malaysian license plate cases are still an open issue. Our company purchased and tested several existing LPR products, yet we failed to achieve high accuracy rate. It is due to lack of enforcement towards unstandardized license plate format practice from road user. We believed that a collaboration between researchers (UKM) and industry will be able to solve the issues regarding Malaysia license plate recognition.

The objective of this paper is to propose an adaptive thresholding method for thresholding both the number plate from the car image and the individual characters within the number plate region, and to compare its results with those obtained by a fixed threshold, another state of the art thresholding method, as well as Otsu’s automatic thresholding method, in relation to a Malaysian license plate recognition system.

This paper is organized as follows. Section II reviews the state of the art in automatic threshold selection. Section III explains the proposed method and Section IV presents the state of the art in automatic threshold selection. Section V explains the proposed method and Section VI presents and analyses the experimental results. Finally, conclusions are presented in Section VII.

II. STATE OF THE ART

Thresholding is a straightforward technique in transforming a gray scale image into a binary image that can facilitate the segmentation process. If \( h(x, y) \), is the input image, the value of the output image, \( \varphi \), at position \((x, y)\), given a threshold \( \Omega \), is:

\[
\varphi(x, y) = \begin{cases} 
1 & \text{if } h(x, y) \geq \Omega, \\
0 & \text{else } h(x, y) < \Omega.
\end{cases}
\]

Recent most popular global binarization used on license plate recognition is Otsu thresholding method for its simplicity and fast performance. In addition, one of the most recently introduced methods for threshold determination known as adaptive contrast is also based on Otsu method. Meanwhile, Niblacks and Sauvolas methods become an alternative to solve shadows and illumination problem on license plate images. Global binarization methods are dependent on distinct image foreground and background such that the preprocessing and post-processing on binarization can contribute towards optimal results. Some recent work such deblurring, image filtering algorithms and super-resolution are good example for preprocessing. Although post-processing approach are not popular compared to preprocessing, some recent work such dirt effect elimination show decent result in character segmentation.

One of the most recent introduced methods for threshold determination is based on the local entropy. This approach has been widely used for measuring the local information content or uncertainty and the information content of a probability distribution. First, the co-occurrence matrix of the image is constructed considering pairs of pixels that are either one below the other or one next to the other. For \( N \) grey level values, the co-occurrence matrix is \( N \times N \) in size. Selecting a threshold value \( M \) divides the matrix into four quadrants (see Figure 2). The entropy of each quadrant can be computed and the threshold that maximises the sum of the entropies of the quadrants that correspond to the object and the background is selected. If \( p_{ij} \) is an element of the co-occurrence matrix, we normalise it so that the sum of all values inside each quadrant is 1.

Assuming a bright object, the elements of the matrix in quadrant C (see figure 2) are divided by \( P_B \), and the elements of the matrix in quadrant A are divided by \( P_O \),

\[
P^O_{ij} = \frac{p_{ij}}{P_O}, \quad p^B_{ij} = \frac{p_{ij}}{P_B},
\]

where,

\[
O = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} p_{ij}, \quad B = \sum_{i=0}^{M} \sum_{j=0}^{M} p_{ij}.
\]

Then the Shannon entropy, \( E_O(M) \) and \( E_B(M) \) for object and background respectively, is:

\[
E_O(M) = -\frac{1}{2} \sum_{i=0}^{M} \sum_{j=0}^{M} p^O_{ij} \log_2 p^O_{ij},
\]

\[
E_B(M) = -\frac{1}{2} \sum_{i=M+1}^{N} \sum_{j=M+1}^{N} p^B_{ij} \log_2 p^B_{ij}
\]

The threshold that maximises the total entropy,

\[
E_T(M) \equiv E_O(M) + E_B(M)
\]

is considered to be the optimal threshold. Fuzzy and honey bee optimization has been incorporated into the method of threshold identification. The local entropy computation time is similar to that for Otsu’s automatic threshold determination method. In this paper, we shall compare the results of the entropy-based and Otsu’s methods with the results of a heuristic method presented in the next section for an LPR system.

In some cases, a single threshold is not applicable for global segmentation. Hence, multiple threshold values are identified to segment an image globally. A statistical recursive approach proposed in calculates first the mean \( \mu \) and standard deviation \( \sigma \) of the image. Two thresholds are then defined:

\[
S1 = \mu + \kappa_1 \times \sigma
\]
where \( \kappa_1 \) and \( \kappa_2 \) are free parameters.

Pixels with value below the low threshold constitute one segment and pixels with value above the high threshold another segment. In the output image, each such pixels is assigned the average grey value of the segment to which it belongs. Pixels with values in between the two thresholds are thresholded further, in the same way, by calculating their mean and standard deviation, to form two new segments in the output image, carrying grey values equal to their respective means, and a central segment, which may be further subdivided in the same way. The method is repeated recursively, until the central segment diminishes, or a prespecified number of thresholds has been defined, or the Peak Signal to Noise Ratio (PSNR) reaches a certain value. PSNR computes noise introduced in the image by replacing the true grey values of the pixels with their average grey value of the segment to which they belong:

\[
PSNR = 20 \times \log_{10} \left( \frac{255}{RMSE} \right),
\]

where

\[
RMSE = \sqrt{\frac{1}{S \times T} \sum_{x=1}^{S} \sum_{y=1}^{T} \left[ g(x, y) - h(x, y) \right]^2}
\]

III. The Proposed Method

The proposed thresholding method consists of three steps: (1) identify the type of image according to its histogram; (2) calculate blob distributions for various threshold values and (3) select threshold value(s) based on a simple heuristic decision rule. Each of the steps is explained next.

1) Identify the type of image according to its histogram: At this stage the contrast of the image is assessed and the image is categorized as dark, of medium brightness or bright, according to the shape of its histogram. We simply partition the 8-bit image into three levels by dividing 256 into 3 predetermined categories and sum those pixel frequencies for each category. Hence, we used the following heuristic rules to assess the shape of the histogram of the image:

\[
\Theta = \begin{cases} 
\text{dark} & \text{if } \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{255} \rho_g} \right) \geq \left( \frac{\sum_{g=86}^{170} \rho_g}{\sum_{g=171}^{255} \rho_g} \right) \land \left( \frac{\sum_{g=86}^{170} \rho_g}{\sum_{g=171}^{255} \rho_g} \right) \geq \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{170} \rho_g} \right), \\
\text{medium} & \text{if } \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{170} \rho_g} \right) \geq \left( \frac{\sum_{g=86}^{170} \rho_g}{\sum_{g=171}^{255} \rho_g} \right) \land \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{170} \rho_g} \right) \geq \left( \frac{\sum_{g=86}^{170} \rho_g}{\sum_{g=171}^{255} \rho_g} \right), \\
\text{bright} & \text{if } \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{170} \rho_g} \right) \geq \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=86}^{170} \rho_g} \right) \land \left( \frac{\sum_{g=86}^{170} \rho_g}{\sum_{g=171}^{255} \rho_g} \right) \geq \left( \frac{\sum_{g=0}^{85} \rho_g}{\sum_{g=171}^{255} \rho_g} \right). 
\end{cases}
\]

Here \( \rho_g \) is the number of pixels in the image with grey value \( g \) and the image is assumed to be an 8-bit grey image. The histograms of images ‘ADG676’, ‘ACP92’ and ‘ACN65’ (Figure 3) are shown in Figure 4.

2) Calculation of blob distributions for various threshold values: We transform source image into binary images using as a series of 25 thresholds, starting from threshold 0 and increasing its value in steps of 10. For each threshold, the number of the connected components created is identified. In Figure 5 we plot the number of the connected components identified this way for each of the example images of Figure 3.

versus the threshold used. We then consider the thresholds that created the peaks in these graphs. This sub-process determines the peak as the adaptive thresholds as the more number of blobs gained, therefore the higher potential for the important objects (foreground) to be segmented or separated from the background of an image.

3) Selection of the thresholds: Some preliminary experiments showed that if the type of image is of ‘medium brightness’, a threshold of 130 sometimes leads to good results. So for this type of image, threshold 130 is always used in addition to the other thresholds identified as explained next. For all types of image, all thresholds that correspond to the peaks of the blob distributions had to be used. Open circles identify the peaks of the blob distributions as in in Figure 5 and dictates in a listing as in Table I for the example images of Figure 3. These thresholds produce images as shown in Figure 6.

We notice that in Figure 6(a) and 6(c), which correspond to the images in the category of ‘dark’, and ‘bright’, respectively, there is a clear character separation in the number plate of thresholds \( \Omega_{Select} = 40 \) for the ‘ACN65’ image, and \( \Omega_{Select} = 60, 90, 110, 170 \) for the ‘ADG676’ image. So, in the subsequent steps, the thresholds selected at this stage will be used to identify all white blobs. The flow chart of the proposed method for threshold selection is shown in Figure 7.
Fig. 3: Examples of (a) ‘dark’, (b) ‘medium’ and (c) ‘bright’ license plate images.

Fig. 4: The histogram distribution, $\rho(0,1,...,255)$, for the ‘ADG676’, ‘ACP92’ and ‘ACN65’ car image from top to bottom, respectively.
Fig. 5: Each threshold used for (a) ADG676, (b) ACP92, and (c) ACN65 images created distributions for number of blobs. Identified circles represent the local maxima in each distribution.

TABLE I: The thresholds that create the peaks in the distributions of Figure 5

<table>
<thead>
<tr>
<th>Peak index, (p)</th>
<th>Peak threshold values, (\Omega_{p} \in \Omega_{0, 1, \ldots, p-1})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADG676</td>
</tr>
<tr>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>1</td>
<td>90</td>
</tr>
<tr>
<td>2</td>
<td>110</td>
</tr>
<tr>
<td>3</td>
<td>170</td>
</tr>
<tr>
<td>4</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>240</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
</tr>
</tbody>
</table>
Fig. 6: Examples of the thresholding results using as thresholds the grey values that correspond to the peaks of the graphs in Figure 5 for (a) ‘ADG676’, (b) ‘ACP92’ (also includes the result for the 130 threshold value), and (c) ‘ACN65’, respectively.
Heuristic Threshold Process

Obtain image category

- Get Original image, \( \text{imgO} \)
- Set as dark
- Set as medium
- Set as bright

Obtain peak threshold(s)

- Set \( \text{thresh} = 0 \)
- Identify peaks, \( p \) and threshold value, \( \text{thresh} \)
- Calculate number of blobs of \( \text{imgTthresh} \)
- Apply CCL and Blob Analysis on \( \text{imgTthresh} \)
- If \( \text{thresh} < 256 \)
  - Threshold value candidates

Select threshold values

- Get \( p \) and threshold values, \( \text{thresh}_{\text{peak}(0,1...p)} \)
- Select = 130 and all peaks, \( \text{thresh}_{\text{peak}(0,1...p)} \)
- Select all peaks, \( \text{thresh}_{\text{peak}(0,1...p)} \)
- Select = 120 and all peaks, \( \text{thresh}_{\text{peak}(0,1...p)} \)

Fig. 7: The flow chart of the proposed threshold selection process.
A. Blob Agglomeration

Further processing will only consider all white blobs identified by the selected threshold applied to an image. Hence, the minimum enclosing rectangle will represent each blob. Eventually, we remove noise by considering each blobs’ height and width at a certain size. In an LPR operational system, the scale of the images captured is assumed to be known. So, these size thresholds may easily be selected and fixed from the beginning. In our case, blobs narrower than 5 pixels and shorter than 15 pixels were treated as due to noise and removed. Then, the agglomeration process selects the remaining blobs into several clusters, starting from the top left, corner of the top left blob and agglomerates blobs that did not differ very much in their vertical position in the image will build up the cluster, as well as in height. In particular, if $H_{\text{current}}$ is the height of the minimum enclosing box of the cluster, and $\bar{Y}$ is the vertical coordinate of the top left corner of the minimum enclosing box of the cluster, the new blob is added to the cluster if,

$$Y_{\text{current}} - \bar{Y} \leq \alpha \times H_{\text{current}}$$

and

$$H_{\text{current}} - \bar{H} \leq \alpha \times H_{\text{current}}$$

where $\bar{H}$ is the height of the blob and $\alpha$ is some parameter (See Figure 8).

The cluster with the maximum number of blobs is considered to be the identified number plate. Then we consider the blobs in the cluster, one at a time. The system as explained in [28] will identify the characters inside the blobs.

IV. EXPERIMENTAL EVALUATIONS

The proposed thresholding method was tested within the LPR system presented in [28]–[30]. This system uses geometric features [28], [30] and either a back-propagation multilayer perceptron (MLP-BP) or a support vector machine (SVM) to classify the characters. In this paper we use only the version with MLP-BP as it gave better results than SVM. This system uses OpenCV library support to implement the thresholding method at its back end. While at its front end a fixed threshold (equal to 130) interface also provided to binarise the captured image [29]. In this paper we supplement the front end of the system with automatic selection of the threshold using

1) the local entropy method [21];
2) Otsu’s method [4];
3) Multi level method [27];
4) the proposed method,

and compare the results obtained with those of the original system. CAIRO and CAIT members collected a dataset consist of 1216 images and it was used in the experiments.

The experiments were performed using the dataset on Intel Dual Core PC with 1.33Ghz processor speed. If all clusters identified in an image consist of a single blob, then we say that no number plate was identified in this image. The first row of Table II shows the percentage of images in which the number plate was not identified. Note that at this stage we do not consider cases where a number plate region had been found but it was totally wrongly placed. These cases will be identified only at the character recognition stage.

Of the clusters that are considered as candidate number plates, we consider first whether the number of blobs corresponds correctly to the true number of characters in the number plate. If it does, we consider that the segmentation was right. Table II also lists the percentage of images for which one, two, or more characters were not segmented, as well as the cases when more blobs than characters in the number plate were identified. Finally, we report the results of the recognition stage. Of the correctly segmented blobs, Table III gives the percentage of wrongly recognised characters for each of the tested methods. The same results are presented also in Figure 9.

Finally, the overall performance of the LPR system is reported in Table IV for the various thresholding methods, for license plate.

![Fig. 8: The blob on the right is added to the cluster on the left, only if $u + v \leq \alpha \times H_{\text{current}}$.](image)

<table>
<thead>
<tr>
<th>TABLE II: Segmentation results.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Thresholding method</strong></td>
</tr>
<tr>
<td>Not found</td>
</tr>
<tr>
<td>Miss 1</td>
</tr>
<tr>
<td>Miss 2</td>
</tr>
<tr>
<td>Miss &gt;2</td>
</tr>
<tr>
<td>Extra</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III: Classification results with MLP-BP.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Thresholding method</strong></td>
</tr>
<tr>
<td>Wrong 1</td>
</tr>
<tr>
<td>Wrong 2</td>
</tr>
<tr>
<td>Wrong &gt;2</td>
</tr>
</tbody>
</table>
of correctly identified number plates (LPD) includes all images for which a number plate region was identified and at least 60% with the true number plate region was overlapped. The percentage of the correctly segmented characters (LPS) includes all images for which the number of blobs identified inside the candidate number plate region is at least equal to the number of characters in the number plate. Finally, the percentage of correctly identified number plates (LPR) includes only those images for which the full car number was recognised. These results are presented in Table V and in Figure 10.

Table V and Figure 10 are constructed based on three types of accuracy rate: LPD, LPS and LPR. The LPD, LPS and LPR accuracy rates are calculated as follows [31]:

\[
LPD \ rate = \frac{(1216 - \text{Notfound} - \text{Smalloverlap})}{1216}, \tag{12a}
\]

\[
LPS \ rate = \frac{(1216 - \text{Miss1} - \text{Miss2} - \text{Miss>2})}{1216} - LPD, \tag{12b}
\]

\[
LPR \ rate = \frac{(1216 - \text{Wrong1} - \text{Wrong2} - \text{Wrong>2})}{1216} - LPS. \tag{12c}
\]

Even though the fixed threshold [29] system has poor performance, it has the lowest recognition time (973.485ms) compared with the Otsu [4] (3549.31ms), Local Entropy [21] (2630.81ms), Multilevel [27] (3789.18ms) and the proposed threshold (4026.14ms). This is due to its lower complexity.
Some results of success and failures of the Otsu threshold LPR system are shown in Figure 11. Some results of success and failures of the fixed threshold LPR system are shown in Figure 12. Some results of success and failures of the multilevel threshold LPR system are shown in Figure 13. Examples of successful and failed experiments of the system using the proposed thresholding method are shown in Figure 14.

V. CONCLUSIONS

Specific applications have their own peculiarities. License plate recognition systems are commercially available nowadays. However, generic LPR systems are not appropriate for all countries, due to variations in the lay out and the font of the characters used in the number plates. A Malaysian LPR system developed recently uses a fixed threshold to segment the number plate and the characters. In this paper, we showed that with the help of a taylor-made thresholding method we can increase the performance of the algorithm significantly. However, the proposed approach is computationally intensive and it cannot be used in its present form for a real time application. It may, however, be used for off-line processing of images with license plates. Another advantage of the proposed approach, is that the adaptive threshold values can adapt to the environment where the system is to be used, for example when there is a high or low contrast situation, such as during the night, mid-day, underground and in a rainy day.

ACKNOWLEDGMENT

Special thanks to: Shahnorbanum Sahran, Sharifah Norziatul Said Samsudin, Sharifah Norzulhida bt Said Samsudin, Noor Faridatul Ainun Bt Zainal, Fadhilah Bt Mustapha, Rosmizan Mat Nong and Nor Hidayah Kasim who had worked hard in giving ideas and collecting the image data in various states. The FRGS/1/2014/ICT07/UKM/02/5 grant entitled OVERLAPPED IRREGULAR SHAPE DESCRIPTOR BASED ON NON-LINEAR APPROACH and also DIP-2015-023 grant entitled OBJECT DESCRIPTOR VIA OPTIMIZED UNSUPERVISED LEARNING APPROACHES are used to fund this project.

REFERENCES


www.ijacsa.thesai.org
Fig. 11: Examples of screen shots of Otsu threshold LPR system for successful and unsuccessful detection. (a) Successfully guessed as ‘WJD2519’, (b) unsuccessfully guessed as ‘W25’. (c) Successfully guessed as ‘AFH5321’, and (d) unsuccessful detecting the location of the license plate.


Fig. 12: Examples of screen shots of Fixed [29] threshold LPR system for successful and unsuccessful detection. (a) Successfully guessed as ‘WJD2519’, (b) unsuccessfully guessed as ‘W4WX25000’, (c) unsuccessful to detect the location of the license plate, and (d) unsuccessful to detect the characters.
Fig. 13: Examples of screen shots of Multilevel [27] threshold LPR system for successful and unsuccessful detection. (a) Successfully guessed as ’TS9867’, (b) successfully guessed as ’WBM5181’. (c) unsuccessfully guessed as ’555’, and (d) unsuccessful guessed as ’N87’.
Fig. 14: Examples of screen shots of the output of the LPR system that uses the proposed thresholding method for (a) (b) successful, and (c) (d) fail detection.