License Plate Detection and Recognition in Complex Scenes Using Mathematical Morphology and Support Vector Machines

Ayman Rabee and Imad Barhumi
Faculty of Engineering, UAE University, P.O.Box 15551, Al Ain, UAE
{aymanrabee, imad.barhumi}@uaeu.ac.ae

***Abstract*—In this paper we propose a highly reliable license plate detection and recognition approach using mathematical morphology and support vector machines (SVM). The approach is composed of three main stages including license plate detection, character segmentation and recognition. A preprocessing step is applied to improve the performance of license plate localization and character segmentation in case of severe imaging conditions. The first and second stages utilize edge detection, mathematical morphology followed by connected component analysis. While SVM is employed in the last stage to construct a classifier to categorize the input numbers of the license plate into one of 9 classes. The algorithm has been applied on 208 car images with different backgrounds, license plate angles, distances, lightning conditions, and colors. The average accuracy of the license plate localization is 97.60%, 90.74% for license plate identification, and 97.89% for number recognition.**

***Keywords*—License Plate Identification, Support vector machines, Digital image processing**

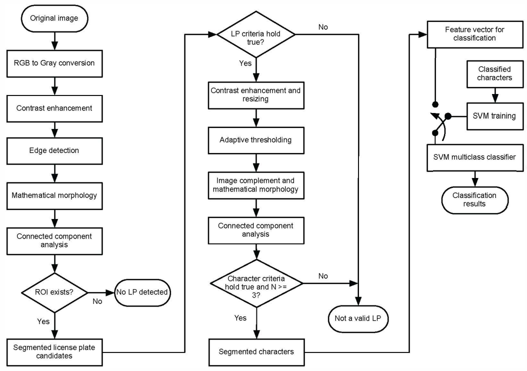


Fig. 1. Flowchart of the proposed method

I. Introduction

Automatic license plate recognition system (ALPRS) employs image processing techniques and character recognition algorithms to recognize vehicles by automatically localizing their license plates (LPs) and then identifying the characters on the LP. Such systems, has important role in many nowadays life applications such as parking management systems, traffic control, speed checking, security and public safety.

ALPRS usually consists of image acquisition, LP localization, character segmentation, and character recognition. While all of these steps are important to identify a vehicle, the overall accuracy of such a system depends crucially on how accurate the system can detect the LP location. In this step, the algorithm has to get rid of different types of noises that would lead to low accuracy rates for LP localization or character recognition.

Over the past few years, many researchers developed different techniques for automatic LP analysis. Some of these methods developed for LP localization operate on LP color [1], [2], texture descriptor [3], Hough transform and contours [4], plate boundary detection [5], [6], neural networks and HSI values [7]. Other methods developed to establish character classifiers based on neural networks [1] and support vector machines [8].

In this paper, we present a new robust ALPRS as shown in Fig. 1. The design is considered for the specific characteristics of Emirates of Abu Dhabi new license plate. The vehicle images were obtained with different backgrounds, illumination, LP angles, distance from camera to vehicle, light conditions (night and day) and different size and type of LPs. The rest of the paper is organized as follows: section 2 presents the proposed method for LP localization, section 3 presents character segmentation, section 4 shows the LP recognition, section 5 demonstrations the experimental results and section 6 concludes the paper.

II. Proposed LP Localization

The flowchart that describes the steps in our proposed ALPRS is shown in Fig.1. The main steps used in LP localization are RGB to Gray conversion and contrast enhancement (preprocessing), edge detection and mathematical morphology.

A. *Preprocessing*

This step is very important for the whole algorithm performance. At first we convert the color image to gray scale using weighted sum method as:

$$I_G = 0.2989 * R + 0.587 * G + 0.114 * B \qquad (1)$$

Then contrast enhancement is applied. This enhancement is essential to extract the LP from images with poor illumination or images that have complex background. Assuming that I_G is the gray scale image, $S_{m\times n}$ is a structuring element with size $m \times n$, and $\ominus$ and $\oplus$ are the erosion and dilation operations respectively. Then the following morphological operations are

defined:
Opening: $I_G \circ S_{m\times n} = (I_G \ominus S_{m\times n}) \oplus S_{m\times n}$
Closing: $I_G \bullet S_{m\times n} = (I_G \oplus S_{m\times n}) \ominus S_{m\times n}$
Top-hat: $I_G \nabla S_{m\times n} = I_G \bullet S_{m\times n} - I_G$
Bot-hat: $I_G \Delta S_{m\times n} = I_G - I_G \circ S_{m\times n}$
The enhancement is obtained by using both bottom-hat and top-hat transformation according to (2).

$$I_{EN} = (I_G + I_G \nabla S_{m\times n}) - I_G \Delta S_{m\times n} \quad (2)$$

B. Edge Detection

One of the most important features of the LP is the edge as it contains many characters. Hence, using this feature helps in reducing the candidate regions considerably. The choice of the gradient operator to be used in finding the vertical edges depends on the capability of the operator to detect edges that are not connected to the surrounding parts. This problem could happen when there is partial contrast which may lead to that some characters edges are connected to the surroundings. The enhancement step explained earlier, solves this issue.

After many tests on different gradient operators like Sobel operator, Prewitt operator, Mexican hat operator, and Robert operator, Sobel operator produced the best results. The resulting edge image could have unwanted long edges, therefore all edges are traced out and those which are greater than the largest LP edges are filtered out.

(a) (b)

Fig. 2. (a) Enhanced image (b) Output image of the morphological step

C. Mathematical Morphology

Mathematical morphology are non-linear filtering operations that provide different tools for image processing based on geometrical and topological concepts to extract information from the processed image. In this paper we are using some morphological operations described in section II-A to highlight and detect LP candidates. After obtaining the edge image, a closing operation is used with structuring element $S_{m\times n}$ whose width larger than the largest space between two consecutive characters on the LP to guarantee that the LP is not eliminated. This operation will connect the LP characters and fill the whole plate region.

In some cases, one closing operation is not enough to fill all the gaps, therefore a second closing operation is used. Then an erode operation followed by an open operation is utilized to filter out all objects that are smaller than the LP. To compensate for the effect of the erosion, a dilation operation is applied and followed by 7×7 median filter. Fig. 2(b) shows the output image after applying all the morphological operations described above.

Fig. 3. Sample car image with the detected candidates

D. LP Segmentation and Verification

In the previous step, the resulting image contains many candidate LP regions. All regions are segmented using eight-connected component algorithm [9], [10] and labeled with different numbers. To decide which region is a potential LP, the following criteria are used. Let CC_i represent a connected component whose size is $w \times h$.

- *Rule 1*: the shape should be rectangular with aspect ratio w/h between 1.5 to 7.8.
- *Rule 2*: the CC_i should not be too small such that $w \geq 30$ and $h \geq 11$.
- *Rule 3*: the CC_i should not be connected to the image boundary.
- *Rule 4*: the CC_i orientation should be horizontal up to $30°$.

If the above rules are applied and there are no candidate regions extracted, the algorithm will report no LP is detected. However, these rules do not guarantee that all false candidates are eliminated especially if there are text blocks on the car body that look like a LP. Therefore, an additional rule explained in section III is used for further verification.

Fig. 3 shows the input car image with the candidates detected by the algorithm in this step. As can be seen in this sample image, there are two candidates in addition to the true LP. These two false candidates can be filtered out using character criteria. Basically the algorithm will count the number N of correctly segmented characters from the LP, if $N \geq 3$ then it is a valid LP. Otherwise it is a false candidate and will be rejected.

III. Character Segmentation

After the LP localization step, candidate regions are analyzed to segment their character. Additionally, the number of correctly segmented characters according to the character criteria are considered as the fifth rule to verify LP.

A. Candidate Preprocessing

Each candidate LP is cropped and processed separately for better performance. The first preprocessing step enhances the image using the same enhancement methods used in (2). Then all the candidate LPs are resized to a width of 228 pixels, the height is calculated to keep the aspect ratio intact.

B. Adaptive Thresholding

After obtaining the gray enhanced image, it is converted to a binary image. The simplest method to do this task is through thresholding. There are basically two types of thresholding: global approach and local approach.

In the global approach, one threshold value is used for all pixels. The problem here is that the illumination could change across different parts of the LP image causing some parts of the plate to be darker or brighter than the characters to be detected. In this case, this type of threshold fails and the resulting character could be connected to other unwanted objects. On the other hand, in the local approach the gray scale plate image is divided into subimages using windowing and a threshold value is chosen based on the histogram of each subimage. The problem with this approach is that it is time consuming. Instead, the local threshold is statistically calculated for each pixel based on the intensity value of its local neighborhood. This approach is referred to as adaptive thresholding. The image is then complemented since the characters are black on a white background.

Fig. 4 shows a sample LP processed with global thresholding. It is clear that this approach is not sufficient to produce satisfactory results where the characters 3, 0, and 1 are connected to each other because of the shadow in the original image. The same image has been processed with adaptive thresholding as shown in Fig. 5(a). The characters in this case are separated and can be forwarded to the next stage.

(a) (b)

Fig. 4. (a) binarized image using global thresholding (b) Complemented image

(a) (b)

Fig. 5. (a) binarized image using adaptive thresholding (b) Image after applying morphological operations

C. Morphological Operations

The binary image obtained from the previous step may have some undesired parts and in some cases these parts are connected to the LP characters due to shadows, screws, dirt, etc.. These parts can be removed by two morphological operations: area opening and erosion. The area opening is an operation for removing objects that are smaller in area than a predefined threshold. Then erosion operation is applied with structuring element designed such that only small objects are removed without affecting the LP characters. After that a dilation operation is applied to compensate for the erosion operation. Fig. 5(b) shows the output of this step.

D. Connected Component Analysis

Following the binarization step, the objects of the LP are segmented using connected component analysis and labeling.

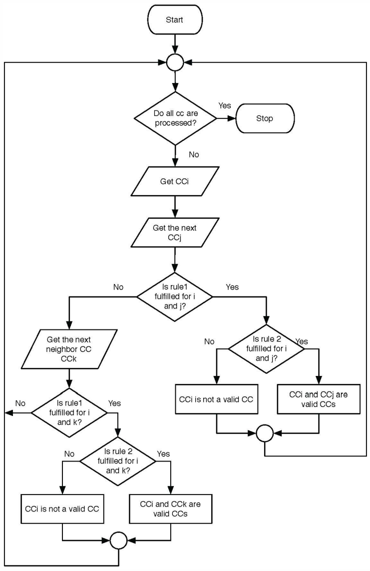
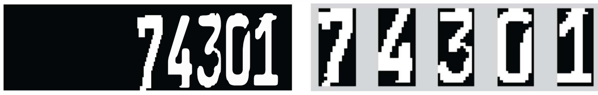


Fig. 6. character verification criteria



(a) (b)

Fig. 7. (a) Image after applying character criteria (b) segmented characters

The height and width of every object are calculated, the objects whose height h and width w do not satisfy: $85 \geq h \geq 11$, $60 \geq w \geq 7$ are rejected.

It was noticed that some false objects may still satisfy the height and width criteria. A rejection algorithm is proposed as explained in the flowchart of Fig. 6, where rule 1 and rule 2 are as follows:

RULE 1. The horizontal distance between two objects ≤ 7
RULE 2. The vertical distance between two objects ≤ 11

By applying the previous character criteria, false objects were removed from the LP. The ALPRS main algorithm will decide whether the detected object is a correct LP or not based on the number of correct characters (LP should contain N characters such that $N \geq 3$). A sample final plate image that shows the output of applying character criteria is shown in Fig. 7(a) and the segmented characters are shown in Fig. 7(b).

IV. CHARACTER RECOGNITION USING SVM

In support vector machines (SVMs) [11], the basic idea is to find the optimal hyper-planes between data points of different classes. For a given training data (x_i, y_i) for $i = 1 \dots N$, the optimization problem for the SVM is formulated as follows.

$$\min_{w,\zeta,b} J(w,\zeta) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \zeta_i, \tag{3}$$

such that

$$y_i(w^T \varphi(x_i) + b) \geq 1 - \zeta_i, \qquad i = 1, \dots, N \tag{4}$$

$$\zeta_i \geq 0, \qquad i = 1, \ldots, N, \tag{5}$$

where C is a positive regularization constant which is chosen empirically, w is the weight vector for training parameters, ζ_i is a positive slack variable indicates the distance of x_i with respect to the decision boundary, and φ is a nonlinear mapping function used to map input data point x_i into a higher dimensional space. SVMs can be written using Lagrange multipliers $\alpha \geq 0$ for (4). The solution for Lagrange multipliers is obtained by solving a quadratic programming problem. The SVM decision function can be expressed as

$$g(x) = \sum_{x_i \in SV} \alpha_i y_i K(x, x_i) + b \tag{6}$$

where $K(x, x_i)$ is the kernel function and defined as

$$K(x, x_i) = \varphi(x)^T \varphi(x_i) \tag{7}$$

In this work, the Gaussian radial basis function [12] is used. It is defined as $K(x, z) = e^{-\gamma ||x-z||^2}$, and the SVM classifier was implemented using LIBSVM [13], which is one-against-one multi-class classifier.

V. Experimental Results

The proposed algorithm described in the previous sections has been applied to a database of car images that contains 208 images of Emirates of Abu Dhabi new license plates. The images were downsized to 860×567 pixels. The plates are composed of Arabic numbers only and the sample images were taken at day and night with different sizes, distances (1m - 5m) and under various illumination conditions. The database is divided into two sets. The first set is composed of 40 images with total of 200 characters and is used for training the SVM character classifier. While the second set is composed of 168 images and are used for testing the performance of the classifier. However, for LP localization the whole database was used.

Once the character is extracted from character segmentation step, it is resized to 25×13 pixels. Then the feature vector is formed from direct pixel values with size of 325. Each character is classified into one of the 9 classes implemented in the SVM character classifier that corresponds to the ten digits from $0 - 9$.

The kernel used is the Gaussian RBF which can be tuned using a single parameter γ. To get the best performance of the SVM classifier, the optimal combination of the regularization constant C and γ should be found. Therefore, a grid search with exponential growing sequences was used. Each combination is checked using cross validation. The parameters with the best cross validation accuracy are selected. In this study, C and γ are found as $C = 4$ and $\gamma = 0.03125$.

Table I shows the percentage of successful LP detection, LP identification (all characters are correctly recognized), and character recognition rates. The algorithm achieved 97.60% accuracy for license plate detection (203 out of 208 license plate detected), license plate identification accuracy of 90.74% computed based on full character recognition in the license plate, and character recognition accuracy of 97.89%.

TABLE I
Results of LP detection and Recognition

LP Det. Acc.	LP Iden. Acc.	Char. Recog. Acc.
97.60%	90.74%	97.89%

VI. Conclusion

In this paper, we presented an ALPRS for the new plates of Emirates of Abu Dhabi. The algorithm employs edge detection along with mathematical morphology and component analysis to detect and segment LPs. Utilizing the number of characters and their geometrical properties help a lot to filter out false plates and characters. The use of adaptive thresholding prevents the effect of illumination change across the plate and therefore gives better binarizations. The final stage of the system is character recognition which was implemented using SVM.

References

[1] S.L. Chang, L.S. Chen, Y.C. Chung, and S.W. Chen,"Automatic License Plate Recognition," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 5, No. 1, pp.42-53, 2004.

[2] W. Jia, H. Zhang, and X. He,"Region-based license plate detection," *Journal of Network and Computer Applications*, Vol. 30, No. 4, pp. 1324-1333, 2007.

[3] C.D. Nguyen, M. Ardabilian, L. Chen, "Robust Car License Plate Localization Using a Novel Texture Descriptor," *IEEE International Conference on Advanced Video and Signal Based Surveillance*, Vol., pp.523-528, 2009.

[4] T.D. Duan, and D.A. Duc, "Combining Hough Transform and Contour Algorithm for detecting Vehicles License-Plates," *Proceedings of International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp.747-750, 2004.

[5] Y.M. Alginahi, "Automatic Arabic License Plate Recognition," *International Journal of Computer and Electrical Engineering*, Vol. 3, No. 3, pp.454-460, 2011.

[6] H. Chen, D. Rivait, and Q. Gao, "Real-time license plate identification by perceptual shape grouping and tracking," *The 9th International IEEE Conference on Intelligent Transportation*, Vol. 4, No., pp.1352-1357, 2006.

[7] S.H. Park, K.I. Kim, K. Jung, and H.J. Kim, "Locating car license plate using neural networks," *Electronics Letters IEE*, Vol. 35, No. 17, pp.1475-1476, 1999.

[8] K.K. Kim, K.I. Kim, J.B. Kim, and H.J. Kim, "Learning-Based Approach For License Plate Recognition," *Proceedings of the 2000 IEEE Signal Processing Society Workshop*, Vol. 2, pp.614-623, 2000.

[9] R.M. Haralick, and L.G. Shapiro, "Computer and Robot Vision," , Vol. I, pp.28-48, 1992.

[10] S.E. Umbaugh, "Computer Vision and Image Processing," , Prentice Hall, New Jersey, pp.133-138, 1998.

[11] Christopher J. C. Burges,"A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, Vol. 2, pp.121-167, 1998.

[12] S. Keerthi and C. Lin, "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel," *Neural Computation*, Vol.15(7), pp.1667-1689, 2003.

[13] C. Chang and C. Lin, "LIBSVM: a Library for Support Vector Machines,"2001.