

Implementation of a High Speed Technique for Character Segmentation of License Plate Based on Thresholding Algorithm

Bahram Rashidi¹

¹ Isfahan University of Technology, Isfahan, IRAN
E-mail: b.rashidi@ec.iut.ac.ir

Bahman Rashidi²

²Iran University of Science and Technology, Tehran, IRAN
E-mail: b_rashidi@comp.iust.ac.ir

Abstract— This paper presents, complete step by step description design and implementation of a high speed technique for character segmentation of license plate based on thresholding algorithm. Because of vertical edges in the plate, fast Sobel edge detection has been used for extracting location of license plate, after stage edge detection the image is segmented by thresholding algorithm and the color of characters is changed to white and the color of background is black. Then, boundary's pixels of license plate are scanned and their color is changed to black pixels. Afterward the image is scanned vertically and if the number of black pixels in a column is equal to the width of plate or a little few, then the pixels of that column is changed to white pixel, until create white columns between characters, in continue we change pixels around license plate to white pixels. Finally characters are segmented cleanly. We test proposed character segmentation algorithm for stage recognition of number by code that we design. Results of experimentation on different images demonstrate ability of proposed algorithm. The accuracy of proposed character segmentation is 99% and average time of character segmentation is 15ms with thresholding algorithm code and 0.7ms only segmentation character code that is very small in comparison with other algorithms.

Index Terms— Character Segmentation, Sobel Edge Detection, Thresholding Algorithm, License Plate

I. INTRODUCTION

With the rapid development of intelligent transport systems, automatic identification of vehicles using license plates has played an important role in a lot of applications. The License Plate Recognition (LPR) technology uses image enhancement, feature extraction and classification techniques to search, locate and identify license plate in an image, segment and recognize characters present and classify the vehicle on the basis of a database working with regional car records present at a backend server. The system can also be made available on a range of networks to harmonize

records of vehicles moving on regional basis. The area has already explored intermediate level expects of enhancing image quality and reducing irrelevant information from an image while still leaving enough information for a system to successfully recognize license plate area, characters present, segmented individual characters, define rules based on area or country specific standards and finally the identification of vehicle on the basis of information retrieved [1]. Autonomous segmentation has been categorized as one of the most difficult tasks in digital image processing. Unfortunately, the success or failure of the recognition modules in any application is extremely dependent on the accuracy and reliability of the previous image segmentation process [2]. In the car license plate recognition system, correct characters segmentation is the basis of characters recognition. The level of characters segmentation accuracy has great impact for characters recognition. In the same conditions, more level the accuracy of characters segmentation is, more higher the probability of characters recognition [3]. The segmentation of a line of characters is an important problem emerging in the LPR systems. The objective is to partition image into segments with isolated characters which serve as an input of the Optical Character Recognition (OCR) system. The problem is challenging due to noise in the image, low resolution, space marks, illumination changes, shadows and other artifacts present in real images. Despite a large effort dedicated to the LPR systems a robust method for character segmentation remains to be an open problem. Segmentation techniques of machine printed characters have been studied for a long time. Common approaches to character segmentation specially designed for the LPR systems are based on the projection method, the Hough transform and intensity thresholding with connected component analysis [4]. In this work presents, a high speed and effective method was offered for diagnosing the car number plates and segmentation Persian and non-Persian characters. In this proposed method, primarily, with the aid of a Sobel edging operator, the vertical edges of the image are detached.

Since the resulting image has some disconnections, the result is extended by 3*3 window in such a way that the pixels around the white one is whitened with respect to the width and length of the window that in continue description proposed method with more detail. We propose an accuracy, speed, and performance trade off analysis for character segmentation of license plate using a high speed technique based on thresholding algorithm.

II. RELATED WORK

In recent years many methods for license plate recognition systems have been proposed. In [3], for overcome the problems of quality of image that be shot in the natural environment is poor generally. Use projecting the image to horizontal direction can detects the approximate edge of single license character in the vertical direction. Then design a template on basis of prior knowledge about the license plate, and take similarity measure to match the image segmented by projection. As a result, they can get the template that its similarity is best license characters re-segmentation can be implemented by the best similar template standard. The algorithm is simple. Template matching avoids the complicated floating point operations. It is only the use of arithmetical operations. The whole algorithm not only reduces the complexity, but also has great robust. in [5], their algorithm performs preprocessing on the license plate, such as size normalization, uneven illumination correction, contrast enhancement, incline correction and edge enhancement; then, locates the character segments according to the vertical projection and merges the character segments that belong to the same character or splits the wider character segment according to the prior knowledge; finally, segments the characters according to the number and the width of the character segments. Proposed algorithm is more efficiency under the condition that the license plate images are degraded, such as declining, faded, distorted, fogged, etc, quality degradation license plate. Then a new method for character segmentation and recognition, correcting the image of license plate which uses the improved method of Hough Transform, then employs the normalization method to segment the characters of the license plate and finally utilizes a modified template matching method to recognize the characters of the license plate. But they only consider using Hough Transform to correct the horizontal and vertical tilt brought by relatively small shooting angles, while other forms of deformation have not been taken into account so far [6]. Also [7] a hybrid license plate segmentation approach based on neural network is proposed, which is designed to work under complex acquisition conditions including unrestricted scene and lighting and a wide range of camera car distances. The approach consists of four stages: preprocessing, candidate regions detecting, real vehicle license plate extracting, and character segmenting. Experiments have proved the robustness and accuracy of the approach. The authors present their work in the mainly two stages of this approach; in the

first stage, candidate regions are located using scanning line checking and vertical cutting algorithm. In the second stage, Back Propagation (BP) neural network are used to verify the real license plates, after multiple features of the license plate are extracted based on the two method of principal component analysis (PCA) and texture analysis. A new method is presented based on the vehicle license location In [8], the segment method of vertical projection information with prior knowledge is used to slit characters and extract the statistical features. Then the Radial Basis Function (RBF) neural network is used to recognize characters with the feature vector as input. The design uses RBF as plate character classifier. RBF is a feed-forward neural network and has simple structure. It also has other self features such as optimal approximation and no local minimum, faster convergent speed and simpler topology structure. The main contributions of [9] are: the pipeline of modified techniques, which reduces the number of analyzed elements in a fast manner, and the qualification process that selects the license plate. The presented methodology is based on a connected component analysis. In order to improve the accuracy of license plate character segmentation in [10], the authors, strengthens pre-processing's before segmentation. In character binarization, use improved Bernsen method (this method is a kind of common local threshold method. It obtains threshold by neighborhood grayscale and maintains stroke of text image well) to reduce noise interference; use line-shape filter to remove plate frames. Then character segmentation approach combines projection and inherent characteristics of character, can better solve character adhesion and fracture problem. For tilted characters only in horizontal direction, even without tilt correction, this method can achieve better segmentation. The authors present an efficient algorithm [11], for character segmentation on a license plate. The algorithm follows the step that detects the license plates using an Adaptive Boosting (AdaBoost) algorithm. It is based on an efficient and accurate skew and slant correction of license plates, and works together with boundary (frame) removal of license plates. Various well-known techniques, including Adaptive Boosting algorithm, Hough transform, edge detection, horizontal and vertical projections, and image binarization, are applied to come out with the innovative algorithm. In [12], Firstly, the binarization image is pre-processed, in which those regions that areas are excessively small and connected with the boundary are eliminated; the left and right boundaries of the first character (Chinese character) and the last one (possibly Chinese character) in the Vehicle License Plate (VLP) image are obtained, and the auxiliary lines are added to make the character become a connected region. Secondly, the image with the auxiliary lines added is done by distance transformation based on city block to get the corresponding gray image, and then it is segmented by Watershed Algorithm (WA). Finally, the boundaries, including left, right, top and bottom ones in the segmented characters, are precisely clipped. This

algorithm can address the following problems. First, each part of the Chinese character has disconnected. Second, there is some tilt in the VLP image. Third, there are multi-Chinese characters in the image. In [13], their work is on the basis of a novel adaptive image segmentation technique (Sliding Concentric Windows-SCW) used for faster detection of Regions of Interest (RoI). And an algorithmic sequence able to cope with plates of various sizes and positions. More than one license plate can be segmented in the same image. Additionally with the above contributions, a trainable optical character recognition system based on neural network technology is presented, and connected component analysis in conjunction with a character recognition neural network. An algorithm based on discrete wavelet transform (DWT), is presented [14], the License Plate (LP) can be extracted from different quality of source images under complex environments by using two frequency subbands. We first use the HL subband to search the features of LP and then verify the features by checking whether in the LH subband there exists a horizontal line around the feature or not. In [15], a novel algorithm applied to sliding window analysis, namely Operator Context Scanning (OCS), is proposed and tested on the license plate detection module of a License Plate Recognition (LPR) system. In the LPR system, the OCS algorithm is applied on the sliding concentric windows pixel operator and has been found to improve the LPR system's performance in terms of speed by rapidly scanning input images focusing only on regions of interest, while at the same time it does not reduce the system effectiveness. Additionally, a novel characteristic is presented, namely, the context of the image based on a sliding windows operator. This characteristic helps to quickly categorize the environmental conditions upon which the input image was taken. A two-stage approach consisting of Character coarse segmentation method and Character precision segmentation process is adopted in [16]. It can increase the accuracy of the segmentation and has good segmentation speed.

The remainder of the paper is organized as follows: Section (3) focuses on license plate location algorithm Section (4) emphasizes on character segmentation and comparison and experimental results of proposed method are at section (5). Finally section (6) provides the conclusion of this paper.

III. PROPOSED ALGORITHM LICENSE PLATE LOCATION ALGORITHM

In order to go over this stage, we design an approach that is similar to Method [17]. But proposed method is different from [17]. In proposed method in order to decrease the effect of intense light during daytime, the image contrast is increased and in order to join the external edges of Sobel edge detection we using of with 3*3 window replace star extension, which itself helps increase precision. These are to be explanation below. On the stage of the whitened license plates, the ratio of

white pixels to black pixels was changed 65%. In such a way, the rate of discerning the plates was escalated. In this part the method is explained. The aim here is to show how the license plate of a car is recognized in a picture. The outcome of this part of the photograph includes the plate only. Since the context of the thing is needed, edge detection is used in this part. Now the edge detection algorithm is used, which itself has less complexity and more speed and also provides the desired specifications. For this reason Sobel edge detection Algorithm is applied, since the outcome image of this algorithm has some disconnections. The outcome is extended by 3*3 window in such a way that the pixels around the one white pixel are whitened with respect to the width of length of the window. Now that the lines are nearer to each other, one can fill in the enclosed areas. Of course, this can make some filled things which themselves should be numbered. Now, one should look for some things with some certain specifications. These certain specifications are as follows:

- 1) Having an area of at least 4500 (in pixel scale)
- 2) Having at least a ratio more than 65% between the white pixels to black pixels.

At last, since it is possible in some conditions that more than two things have such specifications, the one with the larger area is chosen. If nothing was found with the specifications, the biggest thing in the photograph is to be chosen. Now, by having the dimensions and characteristics, we separate it from the rest of the image. The equal of this selected part is founded in the colored image. Since the focal aim is to discern the plate and the specified area is rectangular and has a certain proportion (of width to length), it can be recognized irrespective of the background color. In such a way that by differentiation the image, the edges of the image or its enclosed areas are specified and by whitening this enclosed area we will have some white filled things. With respect to the rectangular shape of the number plate and the rectangular shape of the selected areas, the ratio of white pixels to black pixels will be 65%. This is not the case only when the number plate is seen with an angle. In such cases the ratio will decrease in practice to at least 52%. Therefore, the desired item will be found with regard to the given specifications. For example some of the images that were tested are shown in Fig. 1. Also Fig. 2 shows the items on the whitening stage and Fig. 3 shows the separation of whitened areas from the former stage image. In the same way, Fig. 4 depicts the final outcome image.



Fig. 1. Samples of different pictures.



Fig. 2. Pictures in stage of doing white place of plate in proposed method.



Fig. 3. Separation of whitened areas from the former stage images in proposed method.



Fig. 4. Equivalent extraction plate of gray image in proposed method.

IV. PROPOSED CHARACTER SEGMENTATION

Character separating is a procedure in which the number plates' characters are separated. Nowadays, different methods are used for character separating, for example:

- 1) Character separation according to cluster analysis.
- 2) According to vertical illumination.
- 3) The algorithm of analyzing according to light way and the algorithm of categorizing according to merging areas.

However, there are various factors that can negatively affect the separation of characters, the quality of separated plate, noise, weather conditions and inappropriate light, to name a few. What comes below is the explanation of character separation of plates. For this reason, it is better to have a new algorithm based on thresholding in accordance with the specifications of the plate in which the characters are counted as foreground and the rest as background. The proposed character segmentation algorithm is presented in below algorithm form in Fig. 5.

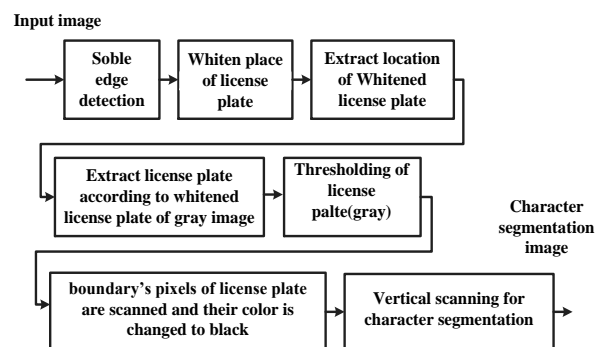


Fig. 5. Proposed character segmentation algorithm.

First we give a summary explanation of thresholding algorithm:

A. Thresholding Algorithm

Segmentation involves separating an image into regions (or their contours) corresponding to objects. As described in [18], we usually try to segment regions by identifying common properties. Or, similarly, we identify contours by identifying differences between

regions (edges). The simplest property that pixels in a region can share is intensity. So, a natural way to segment such regions is through thresholding, the separation of light and dark regions. Thresholding creates binary images from grey-level ones by turning all pixels below some threshold to zero and all pixels about that threshold to one.

If $g(x, y)$ is a thresholded version of $f(x, y)$ at some global threshold T .

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{otherwise} \end{cases}$$

The major problem with thresholding is that we consider only the intensity, not any relationships between the pixels. There is no guarantee that the pixels identified by the thresholding process are contiguous. We can easily include extraneous pixels that aren't part of the desired region, and we can just as easily miss isolated pixels within the region (especially near the boundaries of the region). These effects get worse as the noise gets worse, simply because it's more likely that pixel intensity doesn't represent the normal intensity in the region. When we use thresholding, we typically have to play with it, sometimes losing too much of the region and some-times getting too many extraneous background pixels. (Shadows of objects in the image are also a real pain not just where they fall across another object but where they mistakenly get included as part of a dark object on a light background.) Fig. 6 shows problems with thresholding.

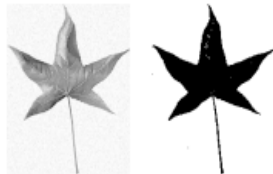


Fig. 6. Problems with thresholding

B. Local Thresholding

Another problem with global thresholding is that changes in illumination across the scene may cause some parts to be brighter (in the light) and some parts darker (in shadow) in ways that have nothing to do with the objects in the image. We can deal, at least in part, with such uneven illumination by determining thresholds locally. That is, instead of having a single global threshold, we allow the threshold itself to smoothly vary across the image.

C. Automated Methods for Finding Thresholds

To set a global threshold or to adapt a local threshold to an area, we usually look at the histogram to see if we can find two or more distinct modes one for the foreground and one for the background. Recall that a histogram is a probability distribution:

$$p(g) = n_g / n \quad (1)$$

That is, the number of pixels n_g having grey scale intensity g as a fraction of the total number of pixels n . Here are five different ways to look at the problem: Known distribution if we know that the object we're looking for is brighter than the background and occupies a certain fraction $1/p$ of the image, we can set the threshold by simply finding the intensity level such that the desired percentage of the image pixels are below this value. This is easily extracted from the cumulative (sum) histogram:

$$C(g) = \sum_{i=0}^g p(i) = \sum_{i=0}^g n_i / n \quad (2)$$

$C(g)$ is the frequencies (the number of pixels) at each intensity value are accumulated. The algorithm of cumulative histogram is shown in below:

Algorithm1. Cumulative Histogram

```
// build a cumulative histogram as LUT
for ( i=0; i < HISTOGRAM_SIZE; ++i )
{
    Sum += histogram[i];

    sum Histogram[i] = sum;
}
// transform image using sum histogram as a LUT (Look Up Table)
for ( i = 0; i < pixelCount; ++i )
{
    outImage[i]= sumHistogram[image[i]] * MAX_INTENSITY
    / pixelCount;
}

```

Simply set the threshold T such that $C(T) = 1/p$. (Or, if we're looking for a dark object on a light background, $C(T) = 1 - 1/p$)

D. Finding Peaks and Valleys

One extremely simple way to find a suitable threshold is to find each of the modes (local maxima) and then find the valley (minimum) between them. While this method appears simple, there are two main problems with it:

- 1) The histogram may be noisy, thus causing many local minima and maxima. To get around this, the histogram is usually smoothed before trying to find separate modes.
- 2) The sum of two separate distributions, each with their own mode, may not produce a distribution with two distinct modes.

E. Clustering (K-Means Variation)

Another way to look at the problem is that we have two groups of pixels, one with one range of values and one with another. What makes thresholding difficult is that these ranges usually overlap. What we want to do is to minimize the error of classifying a background pixel as a foreground one or vice versa. To do this, we try to

minimize the area under the histogram for one region that lies on the other region's side of the threshold. The problem is that we don't have the histograms for each region, only the histogram for the combined regions. Understand that the place of minimum overlap (the place where the misclassified areas of the distributions are equal) is not necessarily where the valley occurs in the combined histogram. This occurs, for example, when one cluster has a wide distribution and the other a narrow one. One way that we can try to do this is to consider the values in the two regions as two clusters. The idea is to pick a threshold such that each pixel on each side of the threshold is closer in intensity to the mean of all pixels on that side of the threshold than the mean of all pixels on the other side of the threshold. In other words, let $\mu_B(T)$ be the mean of all pixels less than the threshold and $\mu_O(T)$ be the mean of all pixels greater than the threshold. We want to find a threshold such that the following holds:

$$\text{If } g < T : |g - \mu_B(T)| < |g - \mu_O(T)|$$

The basic idea is to start by estimating $\mu_B(T)$ as the average of the four corner pixels (assumed to be background) and $\mu_O(T)$ as the average of everything else. Set the threshold to be halfway between $\mu_B(T)$ and $\mu_O(T)$ (thus separating the pixels according to how close their intensities are to $\mu_B(T)$ and $\mu_O(T)$ respectively). Now, update the estimates of $\mu_B(T)$ and $\mu_O(T)$ respectively by actually calculating the means of the pixels on each side of the threshold. This process repeats until the algorithm converges. This method works well if the spreads of the distributions are approximately equal, but it does not handle well the case where the distributions have differing variances.

F. Clustering (The Otsu Method)

As described in the Otsu Method [19], another way of accomplishing similar results is to set the threshold so as to try to make each cluster as tight as possible, thus (hopefully) minimizing their overlap. Obviously, we can't change the distributions, but we can adjust where we separate them (the threshold). As we adjust the threshold one way, we increase the spread of one and decrease the spread of the other. The goal then is to select the threshold that minimizes the combined spread. We can define the within-class variance as the weighted sum of the variances of each cluster:

$$\sigma^2_{within}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T) \quad (3)$$

Where

$$n_B(T) = \sum_{i=0}^{T-1} p(i) \quad (4)$$

$$n_O(T) = \sum_{i=T}^{N-1} p(i) \quad (5)$$

$\sigma^2_B(T)$ = the variance of the pixels in the foreground(below threshold)

$\sigma^2_O(T)$ = the variance of the pixels in the foreground(above threshold)

And $[0, N - 1]$ is the range of intensity levels. Computing this within-class variance for each of the two classes for each possible threshold involves a lot of computation, but there's an easier way. If you subtract the within-class variance from the total variance of the combined distribution, you get something called the between-class variance:

$$\sigma^2_{Between}(T) = \sigma^2 - \sigma^2_{within}(T) \\ = n_B(T) [\mu_B(T) - \mu]^2 + n_O(T) [\mu_O(T) - \mu]^2 \quad (6)$$

Where σ^2 is the combined variance and μ is the combined mean. Notice that the between class variance is simply the weighted variance of the cluster means themselves around the overall mean. Substituting $\mu = n_B(T)\mu_B(T) + n_O(T)\mu_O(T)$ and simplifying, we get

$$\sigma^2_{Between} = n_B(T)n_O(T)[\mu_B(T) - \mu]^2 \quad (7)$$

So, for each potential threshold T we

- 1) Separate the pixels into two clusters according the threshold.
- 2) Find the mean of each cluster.
- 3) Square the difference between the means.
- 4) Multiply by the number of pixels in one cluster times the number in the other.

This depends only on the difference between the means of the two clusters, thus avoiding having to calculate differences between individual intensities and the cluster means. The optimal threshold is the one that maximizes the between-class variance (or, conversely, minimizes the within-class variance). This still sounds like a lot of work, since we have to do this for each possible threshold, but it turns out that the computations aren't independent as we change from one threshold to another. We can update $n_B(T)$, $n_O(T)$, and the respective cluster means $\mu_B(T)$ and $\mu_O(T)$ as pixels move from one cluster to the other as T increases. Using simple recurrence relations we can update the between class variance as we successively test each threshold:

$$n_B(T+1) = n_B(T) + n_T \quad (8)$$

$$n_O(T+1) = n_O(T) + n_T \quad (9)$$

$$\mu_B(T+1) = \frac{\mu_B(T)n_B(T) + n_T(T)}{n_B(T+1)} \quad (10)$$

$$\mu_O(T+1) = \frac{\mu_O(T)n_O(T) + n_T(T)}{n_O(T+1)} \quad (11)$$

This method is sometimes called the Otsu method, after its first publisher.

G. Mixture Modeling

Another way to minimize the classification error in the threshold is to suppose that each group is Gaussian-distributed. Each of the distributions has a mean (μ_B and μ_O respectively) and a standard deviation (σ_B and σ_O respectively) independent of the threshold we choose:

$$h_{\text{mod } eI}(g) = n_B e^{-(g-\mu_B)^2 / 2\sigma_B^2} + n_O e^{-(g-\mu_O)^2 / 2\sigma_O^2} \quad (12)$$

Whereas the Otsu method separated the two clusters according to the threshold and tried to optimize some statistical measure, mixture modeling assumes that there already exists two distributions and we must find them. Once we know the parameters of the distributions, it's easy to determine the best threshold. Unfortunately, we have six unknown parameters ($n_B, n_O, \mu_B, \mu_O, \sigma_B, \text{ and } \sigma_O$), so we need to make some estimates of these quantities. If the two distributions are reasonably well separated (some overlap but not too much), we can choose an arbitrary threshold T and assume that the mean and standard deviation of each group approximates the mean and standard deviation of the two underlying populations. We can then measure how well a mix of the two distributions approximates the overall distribution:

$$F = \sum_{g=0}^{N-1} [h_{\text{mod } eI}(g) - h_{\text{image}}(g)]^2 \quad (13)$$

Choosing the optimal threshold thus becomes a matter of finding the one that causes the mixture of the two estimated Gaussian distributions to best approximate the actual histogram (minimizes F). Unfortunately, the solution space is too large to search exhaustively, so most methods use some form of gradient descent method. Such gradient-descent methods depend heavily on the accuracy of the initial estimate, but the Otsu method or similar clustering methods can usually provide reasonable initial estimates. Mixture modeling also extends to models with more than two underlying distributions (more than two types of regions).

H. Thresholding Along Boundaries

If we want proposed thresholding method to give stay fairly true to the boundaries of the object, we can first apply some boundary-finding method (such as edge detection techniques) and then sample the pixels only where the boundary probability is high. Thus, our threshold method based on pixels near boundaries will cause separations of the pixels in ways that tend to preserve the boundaries. Other scattered distributions within the object or the background are of no relevance. However, if the characteristics change along the boundary, we're still in trouble. And, of course, there's still no guarantee that we'll not have extraneous pixels or holes. In implementation of proposed thresholding algorithm we first for reduction noise using med-filter. Now, the numbers are separated from the background by the algorithm that we developed optimally (according to

Fig. 9). Now, the background color turns black and the numbers white. This method has been developed out of this very color difference between the numbers and background. Because of the deficiency in precise discerning of the edges of the plate in the previous stage, these areas of the image are not monolithic in color (black & white), but are rather noise-like, these areas are not in characters. The area that in the plate framework are converted with black color by some controlled scanning of vertical scanning and horizontal scanning. Fig. 7 shows proposed vertical scanning and horizontal scanning for license plate as seen from the Fig. 7 by these two method plate boundary have been blackened.

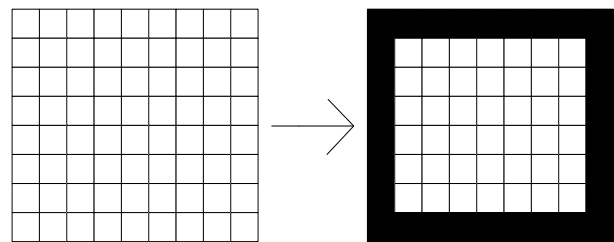


Fig. 7. Proposed vertical scanning and horizontal scanning of License Plate until whose boundary have been blackened.

Now the outcome image will be ready for later stage processes. In next stage, numbers and letters are differentiated by vertical scanning. This stage is carried out in such a way that scan is started from the left corner of the image in a vertical mode from top to down, and the black and white pixels are counted. If the number of black pixels is equal to the width of the image or a little few, the pixels of that column are whitened, otherwise they are left intact. Vertical scanning is carried out up to the last column. Since the area is on the black numbers, this area is taken as a column. Its black pixels are more than the white ones, and if we whiten the black areas among the numbers, we can discern the numbers. On the area where the numbers are located, the number of white pixels in a vertical fashion is higher than that of the black ones. These are left intact, and in the end, with the very primary scan that was used to blacken the edges of the plate they will be whitened so that the outcome consists of numbers and letters. Fig. 8 shows an example of image whose boundary have been blackened and also Fig. 9 shows an example of image that by vertical scanning characters are segmented and boundary have been whitened.

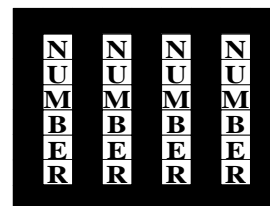


Fig. 8. Example of image whose boundary has been blackened in proposed method.

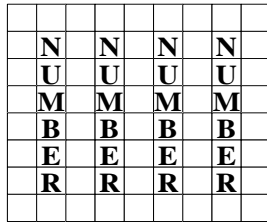


Fig. 9. Example of image that by vertical scanning characters are segmented and boundary have been whitened.

Fig. 10 shows thresholded images of the plates in proposed method.

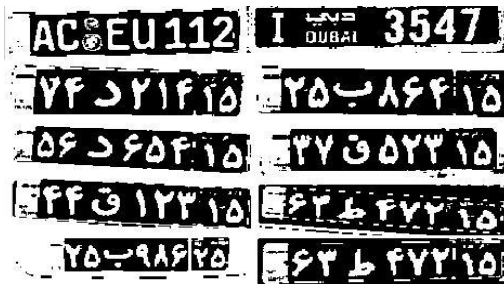


Fig. 10. Thresholded images of the plates in proposed method.

Fig. 11 shows an image whose boundary has been blackened according to the algorithm, and finally, Fig. 12 depicts the final image from the separated characters.



Fig. 11. Samples images that whose boundary has been blackened according proposed vertical scanning and horizontal scanning of Plates.

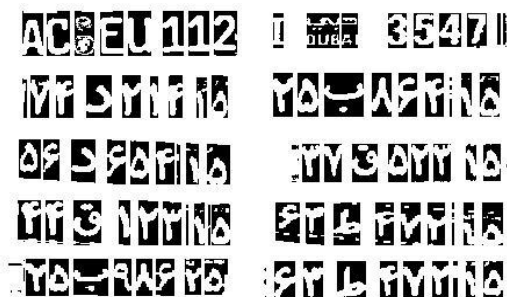


Fig. 12. Final images from the characters segmentation in proposed method.

V. COMPARISON AND EXPERIMENTAL RESULTS

In this paper, the aim of proposed method is design and implementation character segmentation algorithm with high speed. One of the main reasons high speed in proposed method is used thresholding algorithm, because after apply thresholding algorithm we have a binarization image include black pixels and white pixels also separation objects (character on plate with white pixels) from background with black pixels. So working on this image (output image of thresholding algorithm) is comfortable, now we based on segmented image select simple algorithm based on vertical scanning and horizontal scanning pixels, this proposed algorithm is simple and high speed. Below we have calculated formula of the execution time:

$$Time = t_{\text{Soble edge detection}} + t_{\text{Withen place of license plate}} + t_{\text{Extract location of image}} + t_{\text{Thresholding}} + t_{\text{Boundar pixels have been blackened}} + t_{\text{Vertical scanning}} + t_{\text{Horizontal scanning}}$$

$t_{\text{Thresholding}}$: Execution time for thresholding algorithm.

$t_{\text{Boundary pixels have been blackened}}$: Execution time for plate boundary have been blackened.

$t_{\text{Vertical scanning}}$: Execution time for vertical scanning characters are segmented.

$t_{\text{Horizontal scanning}}$: Execution time for horizontal scanning plate boundary has been whitened.

Result of experimentation demonstrates that execution time for boundary pixels have been blackened, vertical scanning and horizontal scanning are very low. Because these three operation only done on segmented image that is equal with size of license plate and size of image is small, thus execution time reducing and it can has good segmentation speed. In order to verification performance the proposed method, 200 images that scaled 640*480 and were different in backgrounds also in light were collected. The computer system used for the investigation had the specifications of 2.20GHz CPU, 2.00GB RAM and the MATLAB version 7.4 was applied for the study. The investigation showed that 198 out of the 200 images were recognized correctly on character separation stage experiment result shows that 2 out of the 200 image were recognition error on the proposed algorithm because these images are very crooked with more angle (out of limit) nevertheless in two images one or two character are include error, and the average time for character segmentation was 13ms. Table I show the results of different methods and proposed method.

Table I. The results of different methods and proposed method.

Character segmentation	Average response time(ms)	Size of pictures	Sample Accuracy Rate	Platform & Processor
[3]	10	640*480	98.8%	Not reported
[7]	225	640*480	95%	Pentium IV (2400MHZ) and 256MB RAM
[11]	200	648*486	98.82%	PC with Pentium 2.8GHz CPU
[6]	90	640*480	100%	Not reported
[12]	163	Not reported	100%	VC++6.0
[12]	1521	Not reported	100%	Matlab7.0
[10]	40	Not reported	97%	PC with Intel CPU (T2130,1.86GHZ) and 1GB RAM, VC++6.0
[13]	111	1024*768	96.5%	Visual C++ 6.0, Pentium IV at 3.0 GHz with 512 MB RAM
[14]	180	400*300	97.33%	IBM X61 notebook PC with 2.4 GHz CPU, 2 GB RAM
[16]	192	Not reported	92.8%	Not reported
Proposed method	13	640*480	99%	PC with 2.20GHz CPU, 2.00GB RAM, MATLAB 7.4

VI. CONCLUSION

In this paper, a high speed and effective method was offered for diagnosing the car number plates and segmentation Persian and non-Persian characters. In this method, primarily, with the aid of a Sobel edging operator, the vertical edges of the image are detached. Since the resulting image has some disconnections, the result is extended by 3*3 window in such a way that the pixels around the white one is whitened with respect to the width and length of the window. Now we can fill the enclosed areas, and in order. to diagnose the plate we should look for the areas that have spaces of greater than 4500 pixels and have white-black ratios of higher than 65%. Therefore, in this way, the number plate can be recognized. After detaching the plate location, by the aid of the thresholding algorithm, and with detaching the characters from the background based on the whitening of character colors and on blackening the edges of the plate in order to uniform the color of these areas, we should begin to whiten the completely black columns by vertical scanning. In the end, the external edges of the separated characters are whitened. In such a way, an acceptable contrast is created between the characters. The proposed segmentation method, have high quality and rapidity in separating of characters, makes the image more appropriate for the stages of character recognition and reducing the rate of calculation in this stage. Speed and high precision are two features of proposed method over other existing works.

REFERENCES

- [1] Syed Adnan Yusuf, "Identification Of Saudi Arabian License Plates", a Thesis Master of Science in Information and Computer Science, King Fahd University of Petroleum & Minerals, January, 2005.
- [2] Beatriz D áz Acosta," Experiments In Image Segmentation for Automatic Us License Plate Recognition", a Thesis Master of Science in Computer Science Virginia Polytechnic Institute and State University, June 18, 2004.
- [3] Deng Hongyao and Song Xiuli , "License Plate Characters Segmentation Using Projection and Template Matching", IEEE International Conference on Information Technology and Computer Science, pp. 534-537, 2009.
- [4] Vojtěch Franc and Václav Hlaváč," License Plate Character Segmentation Using Hidden Markov Chains", Springer-Verlag Berlin Heidelberg, pp. 385-392, 2005.
- [5] Xiaodan Jia, Xinnian Wang, Wenju Li, Haijiao Wang "A Novel Algorithm for Character Segmentation of Degraded License Plate Based on Prior Knowledge", This work is supported by doctoral scientific research foundation of Liaoning Province of China Grant ,2006.
- [6] Jin Quan, Quan Shuhai, Shi Ying, Xue Zhihua "A Fast License Plate Segmentation and Recognition Method Based on the Modified Template Matching", IEEE 2nd International Congress Image and Signal Processing , pp.1-6, 2009.
- [7] Xianchao Zhang, Xinyue Liu, He Jiang, "A Hybrid Approach to License Plate Segmentation under Complex Conditions", IEEE Third International Conference Natural Computation, pp. 68-73, 2007.
- [8] Baoming shan, "License Plate Character Segmentation and Recognition Based on RBF Neural Network", IEEE Second International Workshop Education Technology and Computer Science, pp. 86-89, 2010.
- [9] José María Lezcano Romero and Abilio Mancuello Petters and Dr. Horacio A. Legal Ayala and Dr. Jacques Facon, " Moving License Plate Segmentation" , IEEE 17th International Conference on Systems Signals and Image Processing, pp. 256-259, 2010.
- [10] Shuang Qiao and Yan Zhu and Xiufen Li and Taihui Liu and Baoxue Zhang, "Research of improving the accuracy of license plate character segmentation", IEEE Fifth International Conference on Frontier of Computer Science and Technology, China, pp. 489-493, 2010.
- [11] Xiangjian He and Lihong Zheng and Qiang Wu and Wenjing Jia and Bijan Samali and Marimuthu Palaniswami, "Segmentation of Characters on Car

License Plates”, IEEE 10th Workshop Multimedia Signal Processing , pp.399-402, 2008.

- [12] Lei Chao-yang and Liu Jun-hua ,” Vehicle License Plate Character Segmentation Method Based on Watershed Algorithm”, IEEE International Conference Machine Vision and Human-Machine Interface, pp. 447-452, 2010.
- [13] C.N. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, “A License Plate-Recognition Algorithm for Intelligent Transportation System Applications”, IEEE Transactions on Intelligent Transportation Systems, Vol (3), pp. 377–392, Sept. 2006.
- [14] Yuh-RauWang, “A sliding window technique for efficient license plate localization based on discrete wavelet transform”, Elsevier Expert Systems with Applications, 38(4) pp. 3142-3146, 22 October 2010.
- [15] Ioannis Giannoukos, Christos-Nikolaos Anagnostopoulos, Vassili Loumos, Eleftherios Kayafas, “Operator context scanning to support high segmentation rates for real time license plate recognition”, Elsevier Pattern Recognition, 43(11), pp. 3866–3878, 2010.
- [16] JIAN-XIA WANG, WAN-ZHEN ZHOU, JING-FU XUE, XIN-XIN LIU, “The research and realization of vehicle license plate character segmentation and recognition technology”, IEEE International Conference on Wavelet Analysis and Pattern Recognition, pp.101–104, 2010.
- [17] Abas yaseri, samira torabi, homeira bagheri,” License Plate Recognition (LPR) With Neural Networks”,
<http://www.4shared.com/zip/Wv5c8p8V/tashkhis-pelak-khodro-worldboo.html>.
- [18] Bryan S. Morse, “Thresholding”, Brigham Young University, 1998–2000 Last modified on Wednesday, January 12, 2000.
- [19] N. Otsu, “A threshold selection method from gray–level histogram,” IEEE Transactions on System Man Cybernatics, Vol. SMC-9, No.1, pp. 62-66, 1979.

VERILOG, He now continues on his interest in digital circuits with research in embedded processor systems and VLSI digital chip design.



Bahman Rashidi, received his B.SC. Degree in Computer Engineering from the Science & Technology Sepahan Isfahan University, Iran, in 2009 and he is now M.SC. In the Iran University of Science and Technology, Tehran, IRAN, respectively. He has accepted and published 2 refereed conference papers. His research interests include Computer Architecture, Computer vision, Distributed System, Cloud Computing.



Bahram Rashidi, was born in 1986 in Boroujerd-Lorestan, Iran. He received his B.SC. Degree in Electrical Engineering from the Lorestan University, Iran, in 2009 and he received his M.SC. In the Tabriz university, Iran in 2011 also he is now Ph.D. student in Isfahan University of technology, respectively. His research interests include digital signal processing, DSP processors, computer vision, modeling with hardware description languages VHDL and