Application-Oriented Farsi License Plate Recognition using Deterministic Clustering Algorithm and MSER Detector

Mohammad Mehdi Zeinali[1], Sedigheh Ghofrani[2]
1-Computer Engineering Department, Islamic Azad University, South Tehran Branch, Tehran, Iran
E-mail: st_m_zeinali@azad.ac.ir
2-Electrical and Electronic Engineering Department, Islamic Azad University, South Tehran Branch, Tehran, Iran
E-mail: s_ghofrani@azad.ac.ir

Received: June 2015 Revised: July 2015 Accepted: August 2015

ABSTRACT:
The main contribution of this paper is proposing a novel deterministic and non-iterative clustering algorithm for automatic Farsi license plate recognition (ALPR). In fact, after discarding some regions with low probability in terms of being license plate, the edge points inside every separately remained region are considered as a cluster and a Gaussian component is estimated using expectation-maximization (E-M) algorithm, for every cluster. Candidate regions are obtained by applying application-oriented thresholds for size, aspect ratio and orientation to Gaussian components. Then the license plate regions are identified by counting regions like to numeric characters, exploiting maximally stable extremal regions (MSER) detector whereas numeric characters are extracted by using a proposed algorithm, for discarding non-character regions. So, the algorithm is able to segment the license plate numeric and alphabetic characters simultaneously. Finally for character recognition, a new simple, fast and robust algorithm, which uses feature extraction and template matching technique, is proposed. The method is evaluated for detection and recognition, by using an Iranian image database that includes access control (AC), law enforcement (LE) and road patrol (RP) applications. The method is robust to rotation, skew, and multiplicity of license plate and low quality and complex background images.

KEYWORDS:

1. INTRODUCTION

In general, ALPR system segments license plates and recognizes the characters in images which are captured by using color, black and white or infrared cameras. Diversities in license plate types including background color, characters font, size and image capturing conditions like illumination, hardware performance, distance and angle are two main challenges. Monitoring input-output gates, automatic toll collection and monitoring highways are different ALPR applications. In general, ALPR systems include three modules named license plate detection (LPD), character segmentation and character recognition.

An ALPR method based on analyzing the vertical edges was proposed in [1] where the new algorithm named VEDA extracts the vertical edges in binary images. Although VEDA is 7 to 9 times faster than the Sobel operator, the main challenge is finding an appropriate binarization algorithm. The block based method [2] supposed the regions with high edge density as license plates. Therefore the block based algorithms may find many regions as license plate in an image with complex background. Although methods based on edge statistics, are simple and fast, the edge connectivity is necessary. In [3-5], the unwanted edges were removed and the license plate region is segmented by using the morphology filter. In [6], the license plate region was detected roughly at first phase and precisely at second phase by detecting vertical edges and using mathematical morphology filtering. A weight allocation schema as a criterion for computing edge density of candidate regions was used in [7-9].

The main objective for character segmentation is extracting the license plate characters individually. Horizontal and vertical mapping of intensity levels [10], character contours [11] and binarizing license plate region and using geometrical features [12] are some proposed character segmentation techniques, whereas sensitivity to noise, being time consuming and sticking characters to plate frame due to choosing incorrect thresholds, are their drawbacks. License plate characters were detected by using the scan-line technique and counting peaks of changes in levels of illumination [13].

Third module of ALPR system is character (alphabetic or numeric) recognition. Different size and font of characters, unclean or noisy characters and fragmented characters are challenges. Template matching [14], horizontal and vertical mapping [15], segmenting the character image to small blocks and counting character pixels in each block [16], extracting the character skeleton and counting parts with 0, 45, 90 and 135 degrees angle [17], Gabor filter [18] and extracting topological features such as number of holes and number of end points [19], are some examples of feature extracting techniques. To classify the extracted features, neural networks [20] and support vector machines [21] are common.

In this paper a new application-oriented method for Farsi ALPR is proposed. Three general applications named AC, LE and RP with different settings for license plate size, aspect ratio and orientation are considered. After preprocessing the input image for resizing and sharpening, some regions with low probability in terms of being license plate are omitted based on density of extracted vertical edges and applying morphology operators. A novel deterministic and non-iterative clustering algorithm, combining morphology results and Gaussian component estimation [22] is proposed where the consumed time is reduced significantly and the results in terms of candidate regions are deterministic. For obtaining candidate regions, the application-oriented settings are applied for Gaussian components to discard out of range ones. Then the license plate region(s) is identified by counting regions like numeric characters extracted using MSER detector [23] whereas non-character regions are discarded using the proposed algorithm. So the character segmentation is done simultaneously. Finally a simple, fast and robust method combining the feature extraction and the template matching techniques is proposed for character recognition.

The rest of this paper is organized as follows. In Section 2, the application-oriented license plate recognition method is reviewed. In Section 3, the proposed method named application-oriented Farsi ALPR is explained in detail. The six modules of our approach are resizing and sharpening, vertical edge detection, sliding window and modification, Gaussian components estimation, candidate region segmentation and character recognition. The experimental results for detection and recognition phases of AC, LE and RP applications are reported in Section 4. Finally the conclusion is given in Section 5.

2. APPLICATION-ORIENTED LICENSE PLATE RECOGNITION

The application-oriented license plate recognition method based on edge clustering by k-means algorithm and estimating Gaussian mixture model (GMM) by E-M algorithm [22] were proposed for AC, LE and RP applications. As an example, the entrance gate monitoring for AC, road monitoring for LE and searching stolen cars for RP applications can be referred. Each application needs setting parameters according to the license plate size, orientation and aspect ratio in an image. The GMM [22] for distribution of extracted vertical edge point's using standard Sobel operator is,

$$p(x_i) = \sum_{j=1}^{N_c} \omega_j \phi(x_i \mid \mu_j, \mathbf{V}_j) \tag{1}$$

Where x_i refers to an edge point, $p(x_i)$ is the probability of the edge point, ϕ is the *j-th* Gaussian with mean of μ_j and covariance $\mathbf{V}_j$, N_c is the predetermined number of components or clusters, ω_j is the weighting factor and $\sum_{j=1}^{N_c}\omega_j = 1.$

The E-M algorithm is starting with initial values as mean, covariance and weight of every Gaussian component and then for every iteration, the values are updated until the convergence is satisfied. The edge points are grouped into $k = N_c$ clusters by using the k-means algorithm. Then the mean and covariance of every cluster are considered as the initial values,

$$\begin{aligned} \mu_j &= \frac{1}{N}\sum_{i=1}^{N} x_{ij} \\ \mathbf{V}_j &= \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij}-\mu_j)(x_{ij}-\mu_j)^T \end{aligned} \tag{2}$$

Where x_{ij} denotes the edge point *i* in cluster *j*, *N* is the number of edge points in cluster *j* and $(.)^T$ is the transpose operator. Also the initial value for ω_j is obtained as the number of edge points in cluster *j* to total number of edge points ratio.

For every iteration and for every cluster, $j = 1,\ldots,N_c$, the application-oriented algorithm consists of two steps named expectation (E) and maximization (M),

- In expectation step, the algorithm computes

$$n_j^{(m)} = \sum_{i=1}^{N} \gamma_{ij}^{(m)}$$

where

$$\gamma_{ij}^{(m)} = \frac{\omega_j^{(m)}\varphi(x_i \mid \mu_j^{(m)}, \mathbf{V}_j^{(m)})}{\sum_{l=1}^{N_c}\omega_l^{(m)}\varphi(x_i \mid \mu_l^{(m)}, \mathbf{V}_l^{(m)})}, \qquad i = 1,\ldots,N \tag{3}$$

- In maximization step, the algorithm updates the weighting factor, the mean and the covariance matrix,

$$\omega_j^{(m+1)} = n_j^{(m)} / N$$
$$\mu_j^{(m+1)} = \sum_{i=1}^{N} \gamma_{ij}^{(m)} x_i / n_j^{(m)} \qquad (4)$$
$$\mathbf{V}_j^{(m+1)} = \sum_{i=1}^{N} \gamma_{ij}^{(m)} (x_i - \mu_j^{(m+1)})(x_i - \mu_j^{(m+1)})^T / n_j^{(m)}$$

For a predetermined threshold value α, the algorithm terminates if $\left| l^{(m+1)} - l^{(m)} \right| < \alpha$ where $l^{(m+1)}$ is,

$$l^{(m+1)} = \frac{1}{N} \sum_{i=1}^{N} \log(\sum_{j=1}^{N_c} \omega_j^{(m+1)} \varphi(x_i \mid \mu_j^{(m+1)}, \mathbf{V}_j^{(m+1)})) \qquad (5)$$

After stopping the GMM and representing components as ellipses, some components according to thresholds for size (major diagonal length to image width ratio), aspect ratio and orientation, are discarded. The length and orientation of major and minor diagonals of each ellipse are obtained based on square roots of eigen values and eigen vectors of estimated covariance matrix. The corresponding inequalities for above mentioned thresholds are:

$$s_m < \frac{\lambda_{i,M}}{W_I} < s_M$$
$$r_\lambda < \frac{\lambda_{i,M}}{\lambda_{i,m}} < R_\lambda \qquad (6)$$
$$\theta_m < \tan^{-1}(\frac{\mathbf{v}_{i,M}}{\mathbf{v}_{i,m}}) < \theta_M$$

Where W_I is the image width and $(\lambda_{i,m}, \lambda_{i,M})$ are the smallest and the largest eigen values of covariance matrix $\mathbf{V}_j$ with associated eigen vectors $(\mathbf{v}_{i,m}, \mathbf{v}_{i,M})$. The minimum and the maximum application-oriented thresholds are $[s_m, r_\lambda, \theta_m]$ and $[s_M, R_\lambda, \theta_M]$. The edge clustering, GMM estimation and Gaussian components discarding are performed for AC, LE and RP applications according to the pre-determined number of iterations. In general, the application-oriented algorithm is assuming a small value as the number of components and increasing it at every iteration. For example for AC, the iteration numbers is considered 6 and the clusters are varied among 5 to 10. Also from AC to RP, the number of iterations and the number of components are increased to cover more scales of license plate in images. The algorithm identifies license plate regions as regions with the most similar components (in terms of size, location and orientation).

Due to non-deterministic behavior of k-means and E-M algorithms, different results as the license plate region candidates may be got whenever the algorithm is run. In this paper the novel deterministic and non-iterative clustering algorithm is proposed. In fact, after discarding some regions with low probability in terms of being license plate based on density of extracted vertical edges and applying morphology operators, the edge points inside every separately remained region are considered as a cluster. So, in comparison with [22], the proposed algorithm needs no iterations and the clustering result is deterministic. This property reduces the computation time significantly. In addition, it is not required to define rules for identifying the similar Gaussian components.

3. APPLICATION-ORIENTED FARSI ALPR

In this paper, the proposed method shown in Fig. 1 includes six modules. The first five parts are detecting a license plate and segmenting the characters and the last module is recognizing the extracted characters.

3.1. Image Resizing and Sharpening

The input image is resized without changing the aspect ratio (*ar*) to decrease the computation time. The assumption image size is $n = 480 \times 640$ and the aspect ratio is 4:3. The input image with any arbitrary aspect ratio *'ar'* is resized,

$$x = \text{round}(\sqrt{\frac{n}{ar}}) \qquad (7)$$
$$y = \text{round}(\sqrt{n \times ar}) \qquad (8)$$

Where $x \times y$ is the output image size. To increase the contrast level of a resized image, the mask

$$\begin{bmatrix} C & C & C \\ C & W & C \\ C & C & C \end{bmatrix} \qquad (9)$$

Is used for sharpening [12], where $W = 9A - 1$, $C = -(A/2)$, $A = 2 - \sigma$ and σ denotes the standard deviation of the resized image. After applying the filter, the image is rescaled into range $[0, 1]$ and enhanced by using the histogram equalization. In Fig. 2, three original images of AC, LE and RP applications and their corresponding sharpened images are shown.

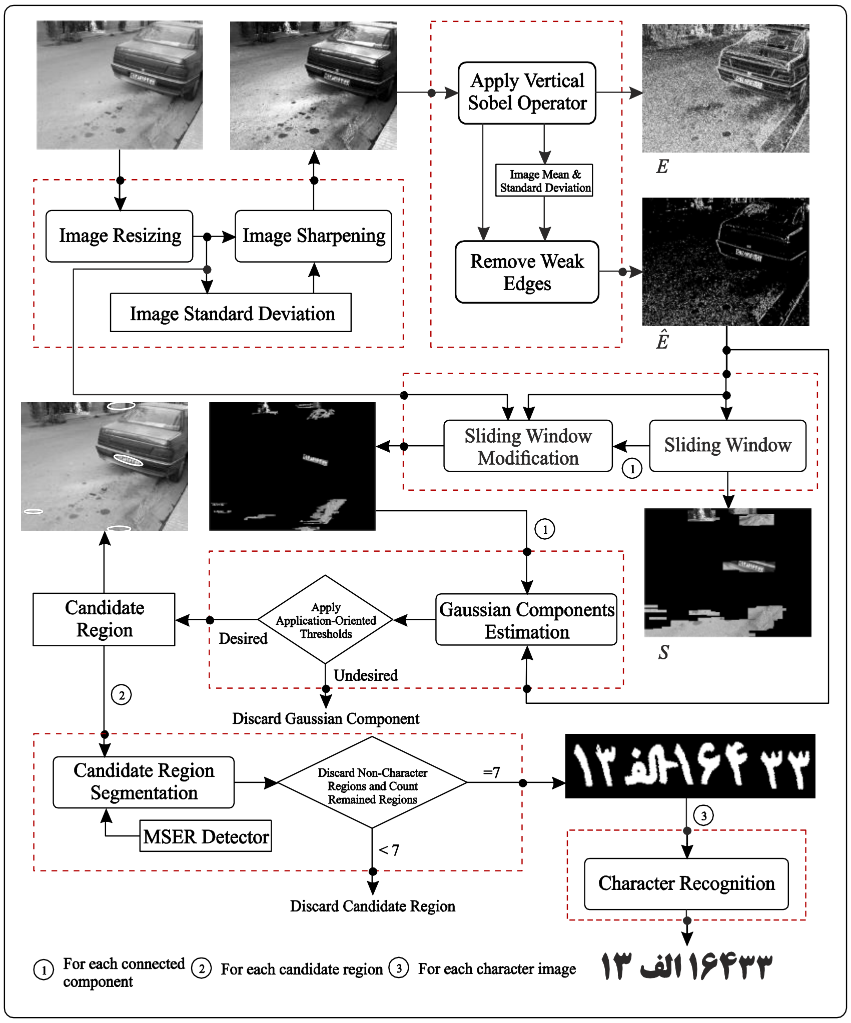

Apply Vertical Sobel Operator
Image Mean & Standard Deviation
Remove Weak Edges
E
Ê
Image Resizing
Image Sharpening
Image Standard Deviation
Sliding Window Modification
Sliding Window
S
Candidate Region
Apply Application-Oriented Thresholds
Desired
Undesired
Gaussian Components Estimation
Discard Gaussian Component
Candidate Region Segmentation
MSER Detector
Discard Non-Character Regions and Count Remained Regions
=7
< 7
Discard Candidate Region
Character Recognition
① For each connected component
② For each candidate region
③ For each character image
۱۳ الف ۱۶۴۳۳

Fig. 1. The block diagram of the proposed method applications (top to bottom)

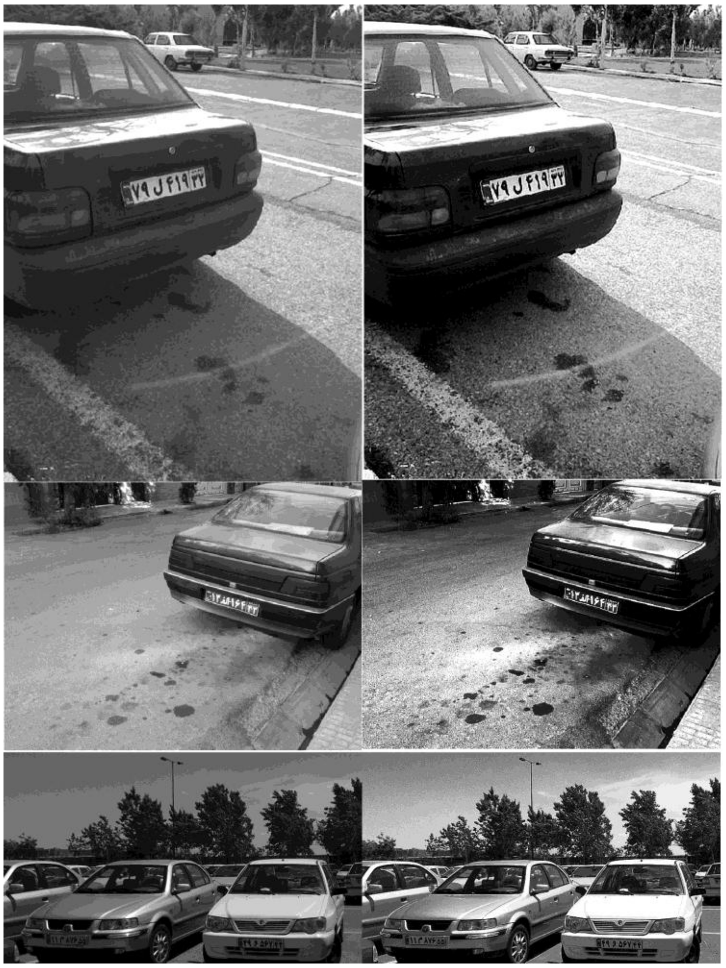


Fig. 2. Three sample original (left-column) and sharpened (right-column) images for AC, LE and RP

3.2. Vertical Edge Detection

In this paper, Sobel mask as the vertical edge detector is used,

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{10}$$

The output values of Sobel mask may be positive or negative as moving from darker intensity levels to lighter ones, or vice versa. Since both edge types are important, the absolute value of output image is used in following. The result of applying Sobel operator is enhanced by using the histogram equalization. Although many edges are detected, the strong edge points are kept,

$$\hat{\mathbf{E}}(i,j) = \begin{cases} \mathbf{E}(i,j) & ; \quad \mathbf{E}(i,j) \geq Th \\ 0 & ; \quad \mathbf{E}(i,j) < Th \end{cases} \tag{11}$$

Where $Th = \mathrm{Max}(\mathbf{E}) - (\frac{m_{\mathbf{E}}\sigma_{\mathbf{E}}}{2})$ is the global threshold value [12], $\mathbf{E}$ and $\hat{\mathbf{E}}$ are the input and output edge images and $(m_{\mathbf{E}}, \sigma_{\mathbf{E}})$ are the mean and the standard deviation of $\mathbf{E}$. Fig. 3 shows the input edge image, $\mathbf{E}$, and the output edge image, $\hat{\mathbf{E}}$.

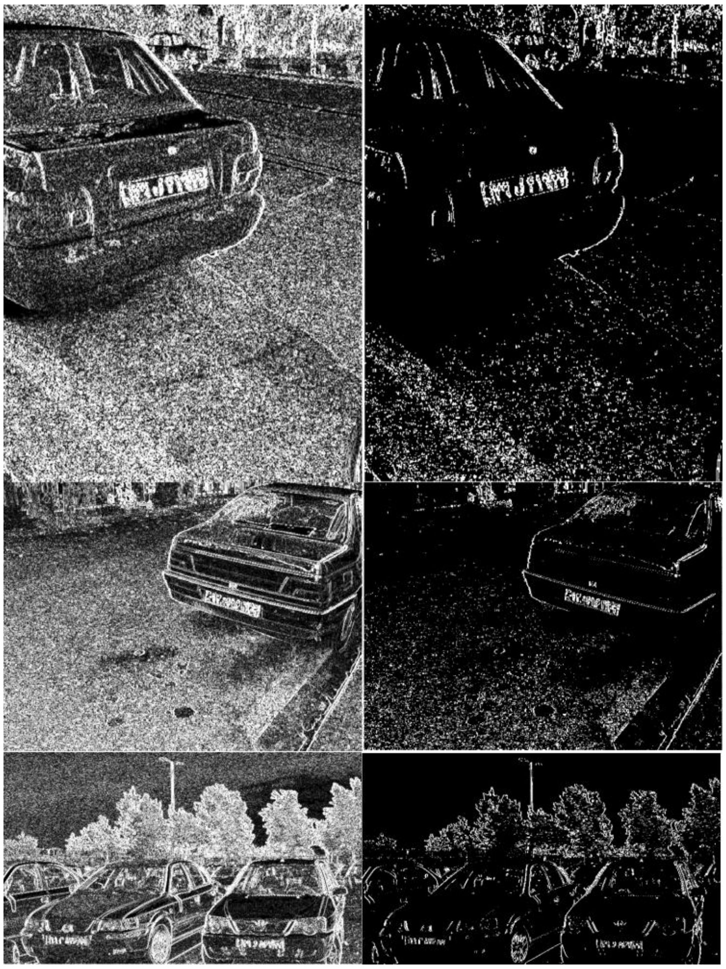


Fig. 3. The output of Sobel vertical edge detector without using a threshold (left-column) and with using the threshold value based on (11) (right-column).

3.3. Sliding Window and Modification

By applying sliding window and modification [12] procedures, some regions of image with low probability in terms of being license plate, are omitted based on density of vertical edges and using morphology operators. Considering $\mathbf{S}$ as the sliding window output binary image, in which the values 0 indicate the discarded regions and the values 1 indicate the remained regions and supposing the initial values 0 for all pixels, the sliding window procedures are,

1. Consider an initial window size.
2. Move the window horizontally and then vertically on $\hat{\mathbf{E}}$ and compute the edge density value of pixels inside the window.
3. Set the values of corresponding pixels of inside window in $\mathbf{S}$ to 1 if the edge density value satisfies the predefined range.
4. Increase the window size and then repeat steps 2-4 if the window width is smaller than the image width named *'y'* in (8).

As the standard Farsi license plate is 10×50 centimeters, the window size for every iteration, $t = 1,2,\ldots,\lfloor y/50 \rfloor$, is considered as,

$$window-size_t = 10t \times 50t \tag{12}$$

For every iteration, suppose the window consists of $p \times q$ pixels, the edge density of pixels inside the window is,

$$\text{edge}-\text{density} = \frac{1}{p \times q}\sum_{i=1}^{p}\sum_{j=1}^{q}\hat{\mathbf{E}}(i,j) \tag{13}$$

The window is moved in vertical and horizontal directions with steps $\lfloor p/3 \rfloor$ and $\lfloor q/3 \rfloor$ pixels respectively. According to [12] the predefined range for desired edge density value is [0.22, 0.35].

As seen in Fig. 4 (left-column), some connected components including the license plate region are kept in the image output of sliding window. The sliding window modification [12] using morphology operators, omits some regions with textures unlike to license plate characters. For every connected component, the sliding window modification procedures are,

1. Count the edge points of rows and columns in $\hat{\mathbf{E}}$ and obtain two corresponding vectors named $\mathbf{sum_r}$ and $\mathbf{sum_c}$ with mean values of $\overline{\mathbf{sum_r}}$ and $\overline{\mathbf{sum_c}}$.
2. Use the closing and the opening morphology filters for the resized grayscale original image (before sharpening). The morphology filter structure is considered a rectangle with size of:

$$
\begin{aligned}
s_x &= \mathrm{round}(0.3\times\overline{\mathbf{sum_c}}) \\
s_y &= \mathrm{round}(0.6\times\overline{\mathbf{sum_r}})
\end{aligned}
\qquad (14)
$$

3. Subtract the output images of opening and closing morphology filters. Then, binarize the subtraction image by using the global threshold value equal to the mean value of its intensities.
4. Replace the connected component in $\mathbf{S}$ with obtained binary image.

As seen in Fig. 4 (right-column), after sliding window modification, the size of some connected components became smaller. In addition, the connected components unlike to license plate are omitted according to the area and the aspect ratio. The characters occupy about 55% of standard Farsi license plate total area. Supposing one pixel per square centimeter for the license plate area in image as the worst case for standard Farsi license plate, the minimum allowed area for a connected component, is 220 pixels (44% of 10×50). So the components with smaller area are removed. Furthermore components with an area greater than one sixth of total image area are also omitted. According to [12], the connected components with aspect ratio out of range [1.8, 13.7] should be discarded.

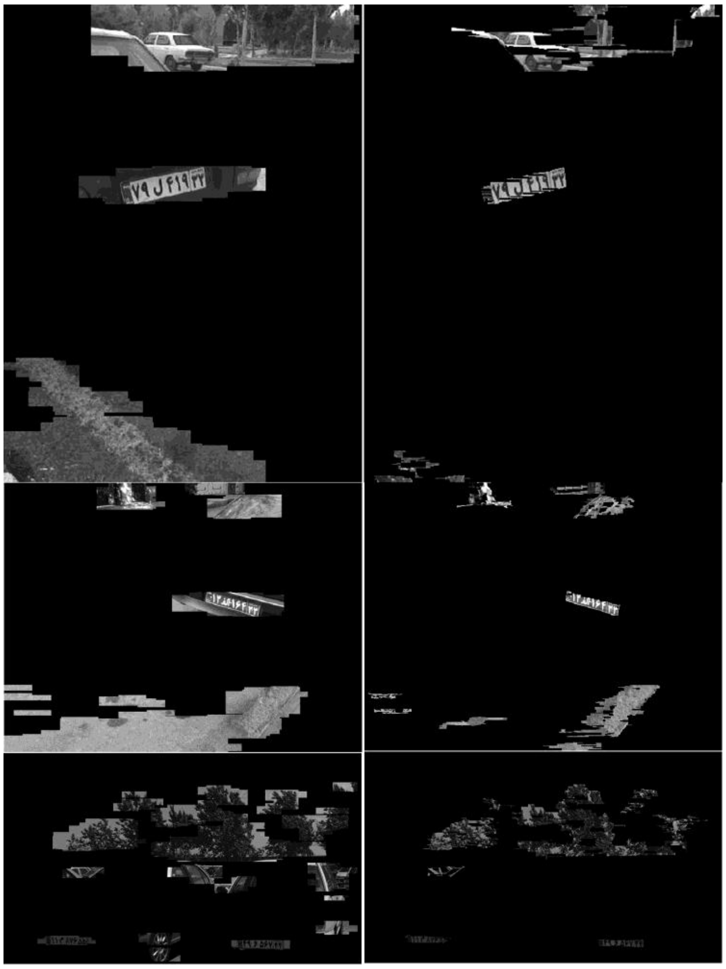


Fig. 4. The remained regions after using sliding window (left-column) and modification (right-column)

3.4. Gaussian Components Estimation

In this paper, the method in [22] is revised. So the novel deterministic and non-iterative clustering algorithm to obtain the license plate candidate regions is proposed. The proposed method is considering every remained connected component as a cluster boundary and the corresponding edge points in $\hat{\mathbf{E}}$ as an edge point cluster. Then for every cluster, only one Gaussian component is estimated. In fact, the clustering result is deterministic because of using the sliding window and modification algorithms. On the other word, no needs for iterations reduce significantly the consumed time. According to [22], by representing each Gaussian component as an ellipse (refer to Section 2) and applying application-oriented thresholds to,

- Ellipse major axis length to image width ratio
- Ellipse aspect ratio
- Ellipse orientation (the angle between major axis and horizontal axis)

some components are discarded to obtain candidate regions as corresponding horizontal surrounding rectangle of remained components. The experimentally chosen ranges for every application are:

AC: $[s_m, s_M]$ = [0.085, 0.25][r_λ, R_λ] = [4, 9] [θ_m, θ_M] = $[-13^\circ, 13^\circ]$ (15)

LE: $[s_m, s_M]$ = [0.05, 0.22] $[r_\lambda, R_\lambda]$ = [3.3, 9.25] $[\theta_m, \theta_M]$ = $[-22^\circ, 22^\circ]$ (16)

RP: $[s_m, s_M]$ = [0.035, 0.27] $[r_\lambda, R_\lambda]$ = [3.34, 11.1] $[\theta_m, \theta_M]$ = $[-30^\circ, 30^\circ]$ (17)

Where [s_m, r_λ, θ_m] and [s_M, R_λ, θ_M], as introduced in (6), are application-oriented minimum and maximum allowed threshold values. We believe that the experimentally ranges of parameters for three applications AC, LE and RP have been obtained globally. It means that they are applicable for a new image or another Iranian license plate database as well. In Fig. 5, results of estimating Gaussian components after discarding undesired ones are shown.

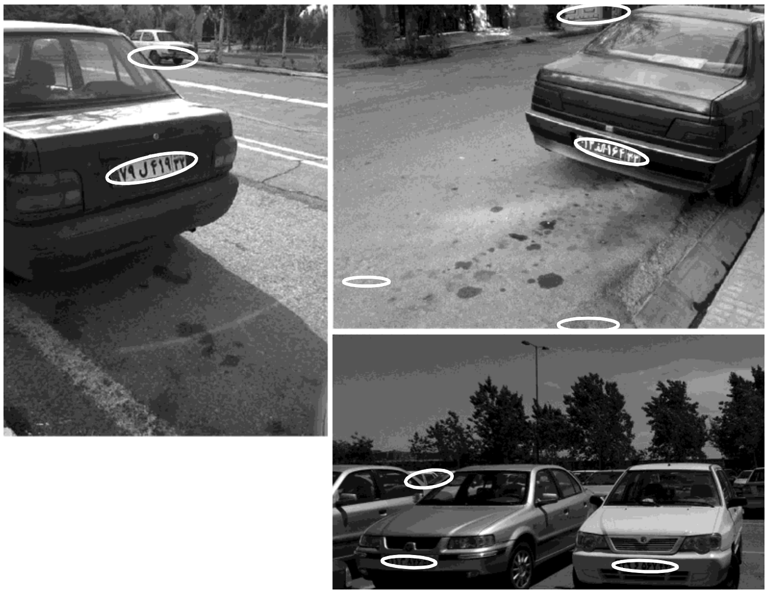


Fig. 5. The estimated Gaussian components (white ellipses) as license plate candidates

3.5. Candidate Regions Segmentation

The candidate region segmentation extracts the numeric characters by using MSER detector [23] to identify the license plate region(s) among candidate regions. A candidate region is considered as a license plate region if the number of its extracted MSER regions as numeric characters is exactly equal to the number of standard Farsi license plate numeric characters. Considering a range of consecutive threshold levels (e.g. 100 to 150 for an 8 bit grayscale image) for binarizing a grayscale image, a MSER region is a connected region where the area and thickness have little changes in binary images, from start to the end of thresholds range. In binary images, a MSER region is either white or black. Furthermore, the three options for MSER detector are desired range of regions area, maximum area changes and difference of two consecutive thresholds.

In this paper, four types among eleven types of standard Farsi license plates are chosen. The four types shown in Fig. 6 are private, public, taxi and state standard Farsi license plates consisting of seven numeric characters and one alphabetic character. For extracting numeric characters, after detecting MSER regions with experimentally chosen range [0.0075, 0.1] for the ratio of regions area to area of surrounding rectangle of Gaussian component as a candidate region, and obtaining corresponding ellipses (based on covariance matrix of constitutive pixels), since non-numeric regions including alphabetic characters may still remain among MSER regions, they are discarded according to,

1. Ellipse properties such as the ratio of major diagonal length to minor diagonal length of Gaussian component, aspect ratio and the angle of major diagonal relative to horizontal axis. The experimentally chosen ranges for these three criteria are [0.427, 1], [1.43, 9] and $[50^{\circ},130^{\circ}]$.
2. Nested regions areas. Nested regions are regions with similar shape whereas their areas, thicknesses and locations have little changes. An example of nested regions for a numeric character is shown in Fig. 7a. Only the region whose area has the median value is kept.
3. Find the median value for the area of all remained regions and obtain the ratio of regions area to the median value. Consider the experimentally chosen range [0.3, 1.85] and omit the regions out of the range. An example is shown in Fig. 7b where the region surrounding two last numeric characters would be removed.
4. The horizontal distance of a region center from neighbors centers. The alphabetic character is localized by finding the maximum horizontal distance between adjacent regions. The maximum distance is between the second and third numeric characters if the alphabetic character was discarded in previous steps. Otherwise the maximum distance is between the alphabetic character and the second or third numeric character. After localizing the alphabetic character, it is removed and the horizontal distances between each two neighbor regions are obtained and compared to the ranges of distances in standard Farsi license plates. Then the regions with out of range distances are removed. Examples of this type of regions are shown in Fig.7c-d.

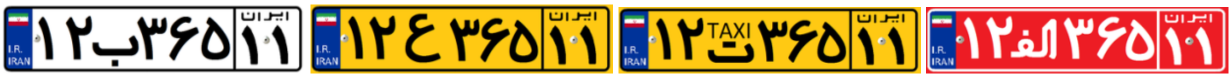


Fig. 6. Four types of Farsi license plates; (the first column) private and taxi, (the second column) public and state.

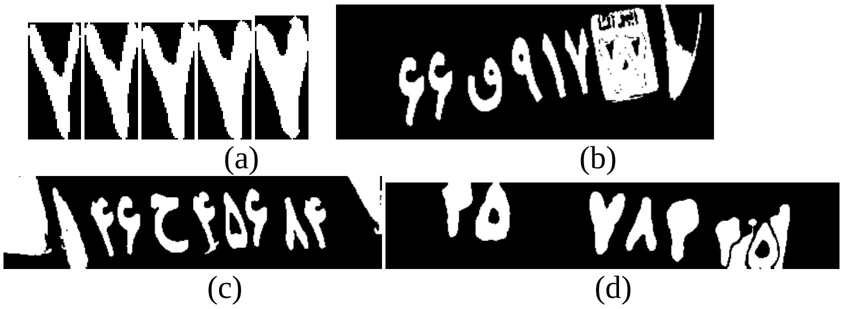

(a) (b)
(c) (d)

Fig. 7. Examples of

a nested regions with little changes in terms of area and thickness,

b non-character regions discarded based on region areas and,

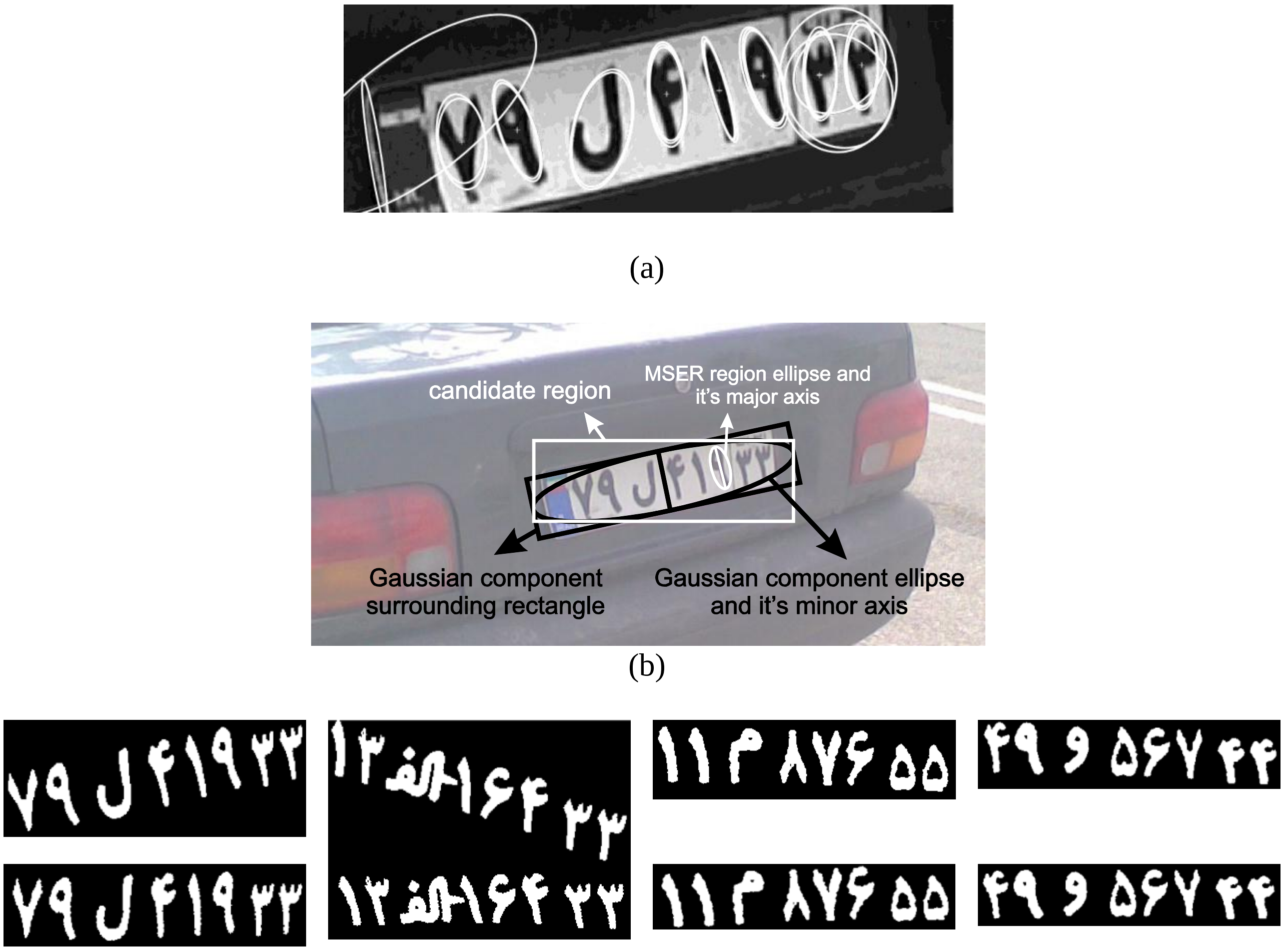
c-d horizontal distances between adjacent regions.

For every explained step above, if the number of regions after discarding non-characters is:

- less than seven, the candidate region is removed.
- equal to seven, the horizontal distances between adjacent regions are checked out. If the maximum distance position is in accordance with the standard Farsi license plate (between second and third numeric characters), the candidate region is considered as a license plate region.
- greater than seven, the region discarding procedures is continued.

At the end, the remained candidate regions are considered as license plates. As seen in Fig. 6, the alphabetic character in an Iranian standard license plate is located in a rectangular shape region between the second and the third numeric characters. In this paper, the Otsu thresholding method for region binarization and alphabetic character extraction is used. We notify that the characters are segmented simultaneously for license plate identification. Furthermore, the MSER detector extracts both black and white regions, so the candidate region segmentation is working properly independent of the license plate types, either the characters are dark or bright in comparison with the background (see Fig. 6).

As an example, the license plate region of AC sample image with white ellipses of MSER regions after discarding non-characters in step 1 is shown in Fig. 8a. Also for classification, we showed the AC sample image and the guide indicators in Fig. 8b. After extracting characters, it is necessary to compensate the rotation angle for the rotated license plates. So the angles between centers of each pair of numeric characters are calculated and the mean value of them is considered as the rotation angle. The extracted license plate regions of sample images and the result of rotation compensation are shown in Fig. 8c. The size of each character image after rotation compensation is normalized to 30×30 pixels.


candidate region
MSER region ellipse and it's major axis
Gaussian component surrounding rectangle
Gaussian component ellipse and it's minor axis

(a)

(b)

(c)

Fig. 8. a) the regions ellipses found by MSER detector, b) the guide indicators for better understanding of subjects, c) the extracted Iranian license plate before (first row) and after (second row) rotation compensation.

3.6. Character Recognition

As seen in Fig. 6, the alphabetic character is always located in order between the second and the third numeric characters, so the numeric and alphabetic characters are recognized individually. For this purpose, a simple, fast and robust method combining feature extraction and template matching is used. Although, we have compensated the rotation, characters may be sheared due to either horizontal or vertical angle between camera and license plate surface, see the second column in Fig. 8c. As said above, the size of each character image is normalized to 30×30 pixels, the number of connected components and the ratio $C_{R_f} / \sum_{f=1}^{30} C_{R_f}$ are considered as two features for each row, R_f , where $f = 1,\ldots,30$ and

C_{R_f} is the number of character pixels in a row. Therefore, two feature vectors named $\mathbf{n_{cc}}$ and $\mathbf{r_p}$ both with size 30×1 are extracted for every character image. We consider the shearing as horizontal moving of image rows with different distances. So the extracted vector features are robust against shearing. In this paper, the Euclidean distance as a simple classifier is used for character recognition. Fig. 9 shows the block diagram of the proposed method for character recognition.

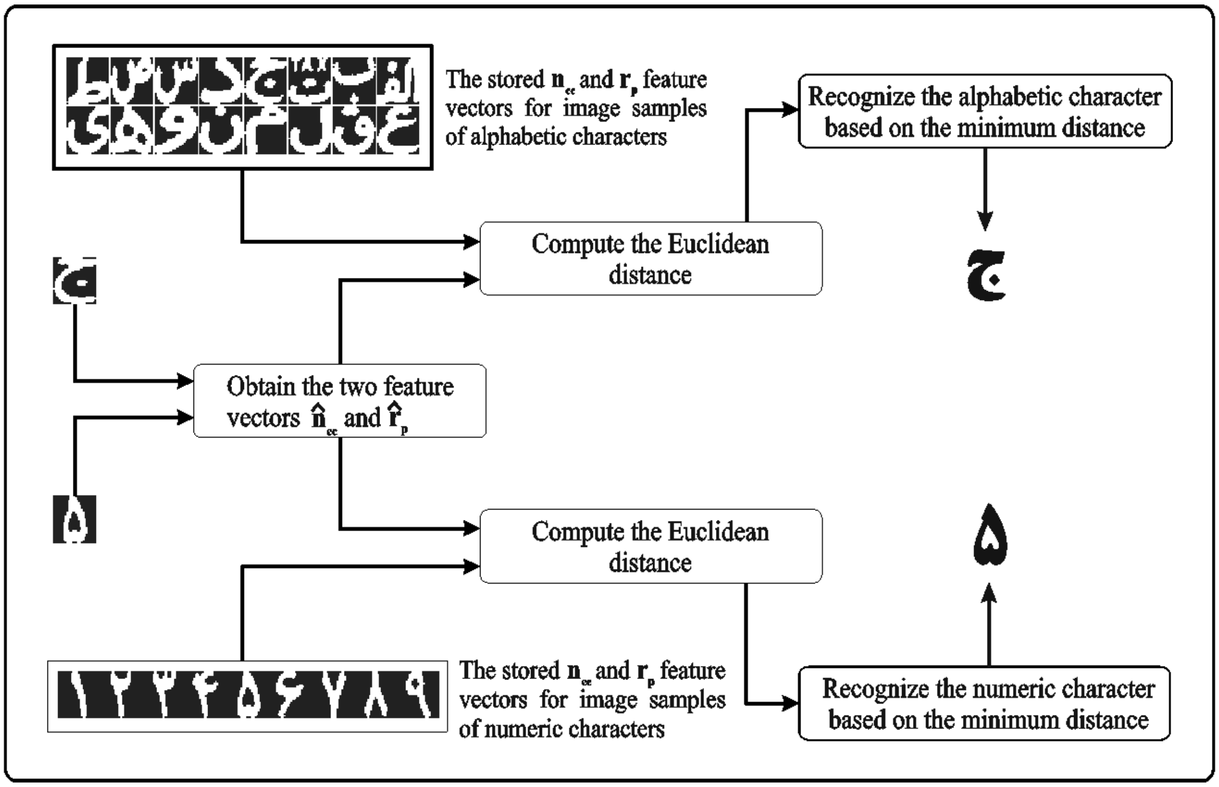

The stored $\mathbf{n}_{cc}$ and $\mathbf{r}_p$ feature vectors for image samples of alphabetic characters
Recognize the alphabetic character based on the minimum distance
Compute the Euclidean distance
ج
Obtain the two feature vectors $\hat{\mathbf{n}}_{cc}$ and $\hat{\mathbf{r}}_p$
Compute the Euclidean distance
۵
۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹
The stored $\mathbf{n}_{cc}$ and $\mathbf{r}_p$ feature vectors for image samples of numeric characters
Recognize the numeric character based on the minimum distance

Fig. 9. Block diagram of the proposed method for character recognition.

4. EXPERIMENTAL RESULTS

In this paper, in order to evaluate the performance of our proposed method, an Iranian license plate database including 390 images is used. Some sample images are shown in Fig. 10. The database includes low contrast, low quality, and blurred images and furthermore there are images with complex background containing multiple license plates. We grouped images into AC, LE and RP applications where in order they consist of 81, 211 and 390 images. All images in AC and LE are duplicated in RP, because as previously mentioned, from AC to RP application the ranges of changes become wider. So, the parameters used for RP covers other applications as well.

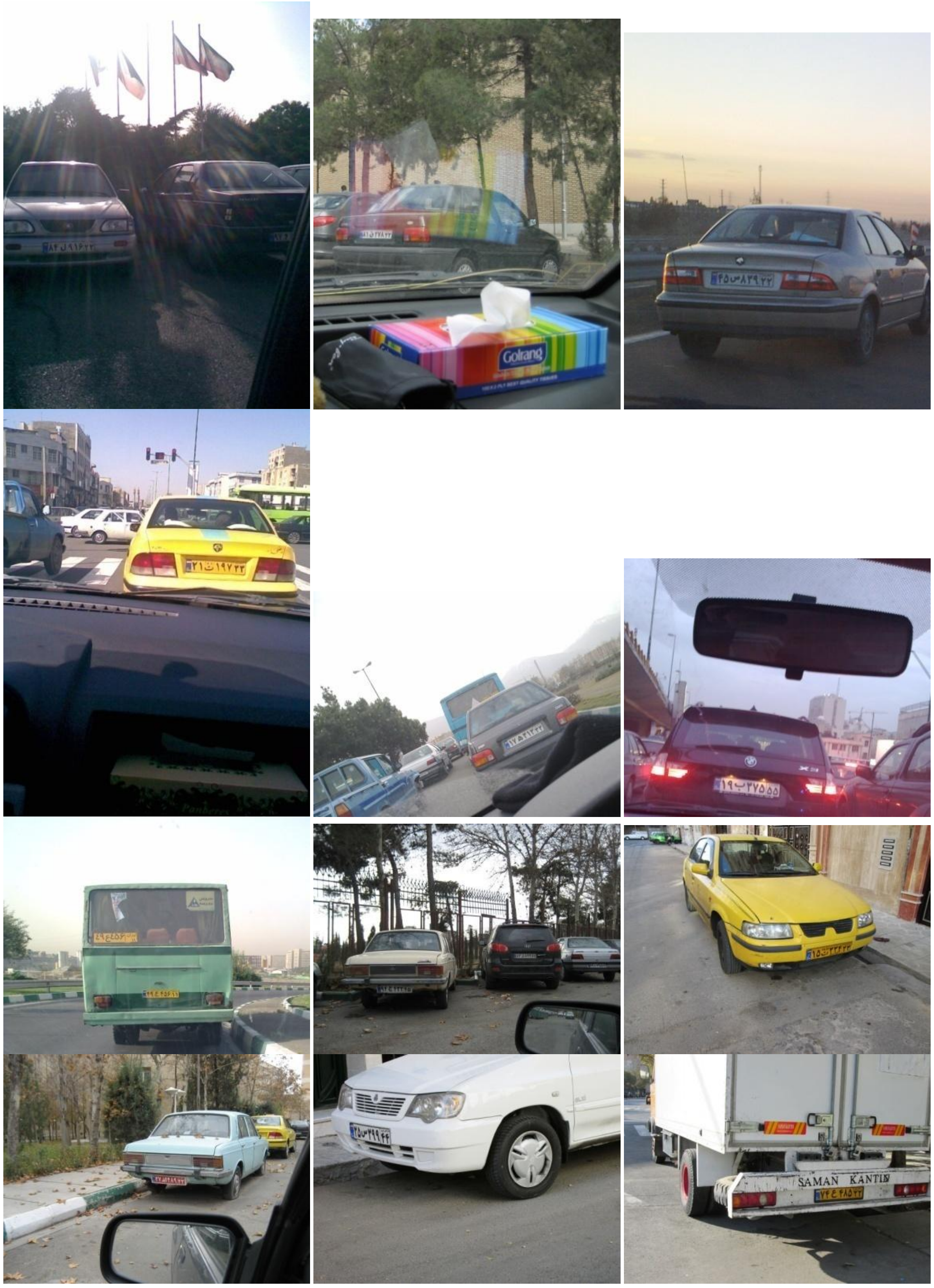


Fig. 10. Some sample images of our database.

The method is implemented using MATLAB 2012 on a PC with Core i7 CPU and 8GB of RAM. The obtained license plate detection accuracy and the consumed time of applying AC, LE and RP settings for three group images are shown in Fig. 11. As seen in Fig. 11a, the RP setting in (17) works properly for the three AC, LE and RP applications as well. In contrast as seen in Fig. 11b, the average consumed time for RP setting is more than both AC and LE. In other words, although the detection accuracy of RP setting is appropriate, it is time consuming in comparison with AC and LE settings.

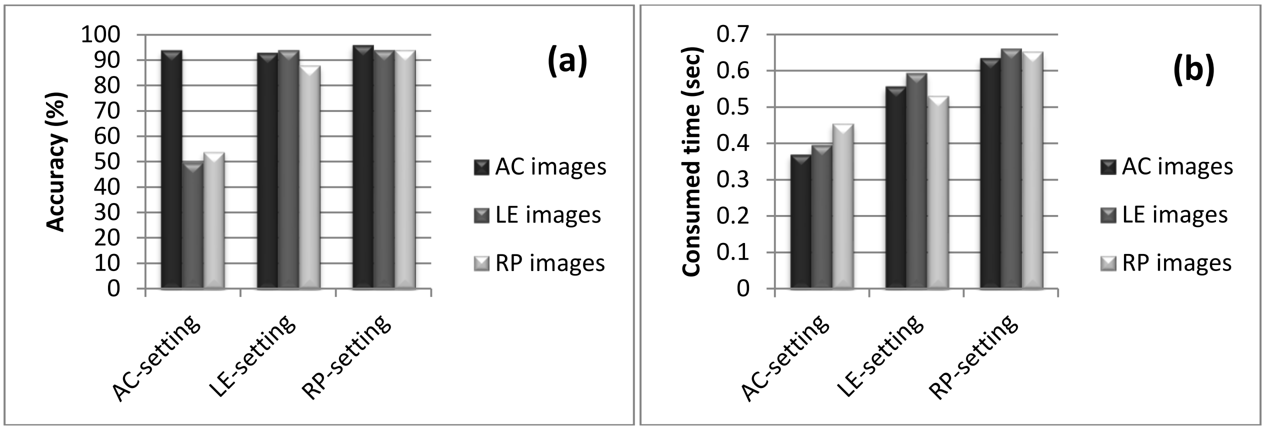


Fig. 11. a) The accuracy and b) the average consumed time for AC, LE and RP applications, according to three different settings.

The proposed method in comparison with some recently published algorithms for Farsi license plates detection and recognition is evaluated according to the achieved accuracy. The results are written in Table 1. Although the accuracy of our method is less than others except in one case, and their databases were not accessible, we believe that the other image databases except [27] were prepared under controlled conditions. It means that the used images in comparison with ours have high quality or having less complexity background. In fact, in [24] the images belong to 20 cars with fixed locations, varying illumination and angles, in [25] the distance between camera and car is 1.5 meters, in [26] the images are captured from a highway high-speed lane. The proposed character recognition algorithm by extracting the two feature vectors and using the Euclidean distance is evaluated irrespective of three AC, LE and RP applications and got 97% accuracy rate with 9 ms consumed time for both alphabetic and numeric characters. Comparing to some other character recognition methods or classifiers that use more complicated techniques, the accuracy of the proposed method is appropriate whereas its processing time is low.

Table 1. Comparison of detection accuracy between our proposed method and some recently proposed methods for Farsi license plate detection.

Method	Accuracy (%)	Time (sec)	Number of Images
[24]	99	-	2000 of 20 cars
[25]	94 (day) 96 (night)	-	150
[26]	96	0.61	500
[27]	92	-	425
The proposed method	94	0.37 (AC) 0.59 (LE) 0.65 (RP)	390

5. CONCLUSION

A reliable and robust method for application-oriented Farsi license plate recognition was proposed in this paper. The new deterministic and non-iterative clustering algorithm by using edge density and morphology filters was introduced where in comparison with [23], the effect of non-deterministic behavior of k-means and E-M algorithms are eliminated. Also, the computation cost is reduced significantly because of being a non-iterative algorithm. In addition, using the MSER detector for license plate region(s) identification segments the characters simultaneously. A simple and fast algorithm for character recognition was proposed with high accuracy even for sheared characters. The method is robust against rotation, skew and multiplicity of license plates, and low quality and having complex background images with very diverse capturing conditions. The proposed method has been tested on an application-oriented classified image database and the experimental results show that the average achieved accuracy is appropriate. Although the proposed method is used for Farsi license plates, the parameters can be set for other license plates as well. Furthermore, the algorithm can be implemented for real time applications by using efficient programming techniques.

REFERENCES

[1] A. M. Al-Ghaili, S. Mashohor, A. Ramli, and A. Ismail, **“Vertical edges-based car license plate detection method”**, *IEEE Transactions on Vehicular Technology*, Vol.62, pp. 26-38, 2013.

[2] H. J. Lee, S. Y. Chen, and W. S. Zheng, **“Extraction and recognition of license plates of motorcycles and vehicles on highways”**, *IEEE*

International Conference on Pattern Recognition, Vol. 4, pp. 356-359, 2004.

[3] B. Hongliang, and L. Changping, **"A hybrid license plate extraction method based on edge statistics and morphology"**, *IEEE International Conference on Pattern Recognition*, Vol. 2, pp. 831-834, 2004.

[4] N. Ballas, S. M. Chai, M. Dwyer, and D. Linzmeier, **"FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators"**, *IEEE International Conference on Parallel and Distributed Processing Symposium*, 2006.

[5] H. P. Wu, H.H. Chen, and R. J. Wu, **"License plate extraction in low resolution video"**, *IEEE International Conference on Pattern Recognition*, Vol. 1, pp. 824-827, 2006.

[6] C. T. Hsieh, Y. S. Juan, and K. M. Hung, **"Multiple license plate detection for complex background"**, *IEEE International Conference on Advanced Information Networking and Applications*, Vol. 2, pp. 389-392, 2005.

[7] V. S. L. Nathan, J. Ramkumar, and S. K. Priya, **"New approaches for license plate recognition system"**, *IEEE International Conference on Intelligent Sensing and Information Processing*, pp. 149-152, 2004.

[8] V. Seetharaman, A. Sathyakhala, and N. L. S. Vidhya, **"License plate recognition system using hybrid neural networks"**, *IEEE International Conference on Fuzzy Information*, Vol. 1, pp. 363-366, 2004.

[9] R. Bermananth, A. Chitra, and V. Seetharaman, **"A robust video based license plate recognition system"**, *IEEE International Conference on Intelligent Sensing and Information Processing*, pp.175-180, 2005.

[10] Z. Qin, S. Shi, J. Xu and H. Fu, **"Method of license plate location based on corner feature"**, *Intelligent Control and Automation*, Vol. 2, pp. 8645-8649, 2006.

[11] K. B. Kim, S. W. Jang and C. K. Kim, **"Recognition of car license plate by using dynamic thresholding method and enhanced neural networks"**, *Computer Analysis of images and patterns*, Vol. 2756, pp. 309-319, 2003.

[12] M. Rasooli, S. Ghofrani, and E. Fatemizadeh, **"Farsi license plate detection based on element analysis and characters recognition"**, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol.4, No. 1, 2011.

[13] Y. S. Soh, B. T. Chun, and H. S. Yoon, **"Design of real time vehicle identification system"**, *IEEE International Conference on Humans, Information and Technology*, Vol. 3, pp. 2147-2152, 1994.

[14] P. Comelli, P. Ferragina, M. N. Granieri and F. Stabile, **"Optical recognition of motor vehicle license plates"**, *IEEE Transactions on Vehicular Technology*, Vol. 4, No.4, pp.790-799, 1995.

[15] C. A. Rahman, W. Badawy and A. Radmanesh, **"A real time vehicle's license plate recognition system"**, *IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 163-166, 2003.

[16] F. Aghdasi and H. Ndungo, **"Automatic license plate recognition system",** *AFRICON Conference*, Africa, Vol. 1, pp. 45-50, 2004.

[17] T. Nukano, M. Fukumi and M. Khalid, **"Vehicle license plate character recognition by neural networks"**, *IEEE Intelligent Signal Processing and Communication Systems*, pp. 771-775, 2004.

[18] M. K. Kim and Y. B. Kwon, **"Recognition of gray character using Gabor filters"**, *International Conference on Information Fusion*, Vol. 1, pp. 419-424, 2002.

[19] 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.

[20] J. Jiao, Q. Ye, and Q. Huang, "**A configurable method for multi scale license plate recognition"**, *Pattern Recognition*, Vol. 42, No. 3, pp. 358-369, 2009.

[21] K. K. Kim, K. I. Kim, J. B. Kim, and H. J. Kim, **"Learning-based approach for license plate recognition"**, *IEEE Signal Processing Society Workshop*, Vol. 2, pp. 614-623, 2000.

[22] G. S. Hsu, J. C Chen, and Y. Z Chung, **"Application-Oriented license plate recognition"**, *IEEE Transactions on Vehicular Technology*, Vol. 62, No. 2, pp. 552-561, 2013.

[23] J. Matas, O. Chum, M. Urban, and T. Pajdla, **"Robust wide-baseline stereo from maximally stable extremal regions"**, *Image and Vision Computing*, Vol. 22, No. 10, pp. 761-767, 2004.

[24] M. Zahedi and S. M. Salehi, **"License plate recognition system based on SIFT features"**, *World Conference on Information Technology*, Vol. 3, pp. 998-1002, 2011.

[25] M. Ayoubi Mobarhan et al, **"A license plate detection algorithm is using edge features"**, *Advances in Intelligent and Soft Computing*, Vol. 166, pp. 413-421, 2012.

[26] A. H. Ashtari and M. Fathy, **"An Iranian License Plate Recognition System based on Color Features"**, *IEEE Transactions on Intelligent Transportation Systems*, 2014, Vol. 15, No. 4, 2014.

[27] M. Ashoori Lalimi, S. Ghofrani and D. McLernon, **"A vehicle license plate detection method using region and edge based methods"**, *Special Issue on Image and Video Processing*, Vol. 39, No. 3, pp. 834-845, 2013.

[28] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, **"Automatic License Plate Recognition (ALPR): A state-of-the-art review"**, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 23, No. 2, 2013.