An Iranian License Plate Recognition System Based on Color Features

Amir Hossein Ashtari, *Graduate Student Member, IEEE*, Md. Jan Nordin, and Mahmood Fathy

***Abstract*—In this paper, an Iranian vehicle license plate recognition system based on a new localization approach, which is modified to reflect the local context, is proposed, along with a hybrid classifier that recognizes license plate characters. The method presented here is based on a modified template-matching technique by the analysis of target color pixels to detect the location of a vehicle's license plate. A modified strip search enables localization of the standard color-geometric template utilized in Iran and several European countries. This approach uses periodic strip search to find the hue of each pixel on demand. In addition, when a group of target pixels is detected, it is analyzed to verify that its shape and aspect ratio match those of the standard license plate. In addition to being scale and rotation invariant, this method avoids time-consuming image algorithms and transformations for the whole image pixels, such as resizing and Hough, Fourier, and wavelet transforms, thereby cutting down the detection response time. License plate characters are recognized by a hybrid classifier that comprises a decision tree and a support vector machine with a homogeneous fifth-degree polynomial kernel. The performance detection rate and the overall system performance achieved are 96% and 94%, respectively.**

***Index Terms*—Color template matching, image recognition, license plate detection, license plate localization, license plate number identification, license plate recognition (LPR).**

I. Introduction

Automatic license plate recognition system (ALPR) has become a high-priority research in recent years. Because the license plate is a unique ID for a vehicle, its automatic recognition has many uses. For example, the license plate recognition (LPR) system can be used in smart parking areas or smart toll stations to open gates for vehicles bearing authorized license plates or to calculate the average speed of a vehicle between two stations by recognizing its license plate at both stations. In addition, by installing LPR systems on roads, particularly in traffic zones and at junctions that need police patrolling, prohibited vehicles can be recognized and their movement monitored. Nowadays, speed control stations on highways take color photos of vehicles that break the speed limit. Usually, on highways in Iran, these control stations use local memory because they are not connected to a central database [1]. The photos are taken by speed control cameras in high resolution, which consume a lot of disk space; thus, after a while, most of the stations encounter a low-disk-space problem. This problem can be solved by the ALPR system, which converts huge data of images into a series of bits. The ALPR system installed at speed control stations uses high-resolution images to recognize license plates. After recognition, the images are compressed into small and low-resolution images and then transferred through normal and low-band connection devices such as short messaging or multimedia messaging by a GSM board. Instead of mere recognizing and sending traffic tickets to the offenders after a few days, penalizing the offenders at the time and place of the incident would be more effective in preventing accidents and deterring drivers from repeat offenses. Thus, an ALPR system based on color images from normal/high-resolution surveillance cameras is highly useful. That is why several studies have applied color features for localizing license plates. Most of their algorithms, which apply color features for license plate localization, are country specific [2]. For localization, they usually convert all image pixels to other color spaces, such as hue, saturation, and value (HSV) or hue, saturation, and intensity (HSI), and find background or foreground color using tiling and hue histogram [3]. Some algorithms apply color features and fuzzy sets to localize license plates [4], [5]. Color edge detection is widely used for license plate detection; in addition, it has been applied as an aided feature in Persian license plate detection [2], [4], [6]. Most of the implemented LPR systems for Iranian license plates are based on infrared photos and, hence, require special photographic equipment for infrared photography [1].

The proposed system, which is based on developments in image processing and optimized template-matching methods, proves that a fast LPR system based on color features is possible and practical. It recognizes license plates in color images without any resizing and conversion, thus reducing the response time. It can be used in automatic toll stations, tunnels, highways, intelligent parking, and traffic zones for LPR by using surveillance cameras. The results based on images from speed control cameras on highways demonstrate the system's capability and reliability.

The functioning of the ALPR system includes two steps. In the first step, the system performs license plate localization, which involves identifying the license plate location in the original image and cropping it. In the second step, the system extracts and classifies all the available characters in the license plate, namely, numbers and letters.

Manuscript received June 4, 2013; revised September 29, 2013 and December 24, 2013; accepted January 20, 2014. This work was supported in part by Universiti Kebangsaan Malaysia under Grant ERGS/1/2012/STG07/UKM/02/9. The Associate Editor for this paper was S. S. Nedevschi.

A. H. Ashtari and M. J. Nordin are with the Department of Computer Science, Pattern Recognition Research Group, National University of Malaysia, 43600 Bangi, Selangor, Malaysia (e-mail: ashtari@ieee.org; jan@ftsm.ukm.my).

M. Fathy is with the Department of Computer Engineering, Iran University of Science and Technology, Tehran 16846-13114, Iran (e-mail: mahfathy@iust.ac.ir).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2014.2304515

In the first step (detection) of the proposed ALPR system, processing is done on the original image without any resizing or transformation. The license plate is detected by using a geometric template on colonies of target pixels. A colony is defined as connected component pixels with the same color. The new approach for detection avoids time-consuming algorithms, such as Hough transform, wavelet transform, and curvelet transform, gray scale or binary conversion in large scale, or even edge detection. Nowadays, due to the developments in professional photography equipment and high-speed camera shutters, it is no longer necessary to carry out high-level preprocessing for noise reduction and sharpening/deblurring of the original image; hence, the preprocessing done at low level ensures faster recognition. Some weather factors (foggy or rainy) and light factors (dark nights or vehicle lights) or broken, smeared, and unreadable license plates can cause vision problems, rendering recognition difficult. Using color features in the proposed system makes the system more flexible and aids in generating fairly reliable results in all the weather conditions because of the system's similarity to human vision and decision [7].

In the second step (recognition), morphological functions and image processing filters improve the quality of the license plate image to optimum posture. All the characters in the license plate are extracted by a connected component or a projection analysis and then sent to the classification section for recognition. A hybrid classifier, comprising a decision tree and a support vector machine (SVM) with a homogeneous fifth-degree polynomial kernel, recognizes the extracted letters and numbers.

II. License Plate Detection

A. Review of License Plate Detection Methods

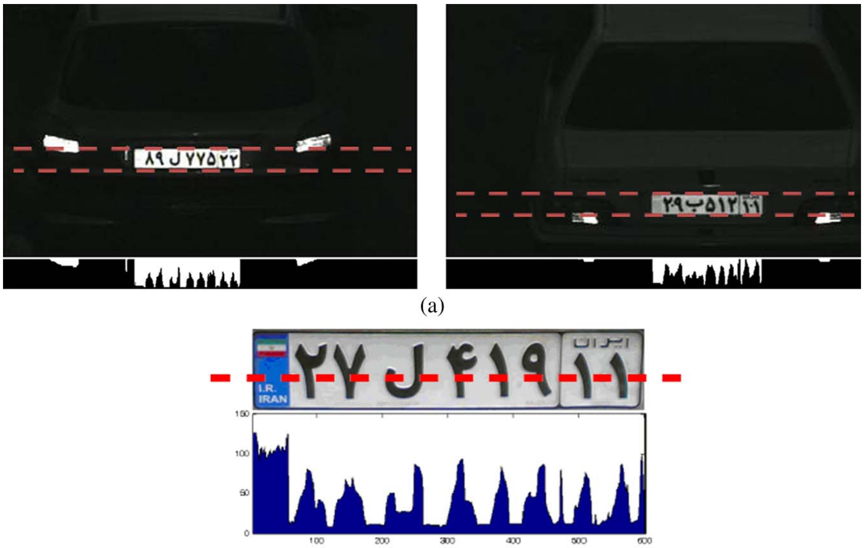
Color features play an important role in the detection and recognition system because the color of an object is the first feature that a healthy human vision looks for, and then, the shape of object is considered [7]. There are many methods for the localization of a license plate, but most of them ignore color features and straightaway convert images into gray-scale or binary images. In the absence of applying color features, the available methods are robust and accurate; thus, using color methods that are usually time consuming was not reasonable [2]. Several methods of detection use color as the basic or aided feature, and usually, they are country specific [2], [4]–[6], [8], [9]. Some localization methods try to apply human intuition and perception for shape localization. Using the aforementioned rules and fuzzy sets, they try to localize a license plate [2], [4], [5], [10]. Jia *et al.* [3] converted all image pixels to another color space such as HSI and used tiled histogram to find background and foreground colors to localize license plate texture. Abolghasemi and Ahmadyfard [6] used color edge detection, such as black–white, red–white, and green–white, for localization. Chang *et al.* [4] used color edge and the set of hue, saturation, and intensity to make an image fuzzy map for license plate localization. Usually, the methods for color analysis are time consuming or involve elaborate processing; in addition, they localize more than one candidate instead of one, which makes them not suitable for real-time LPR systems [1], [2], [6].

Most of the available methods use gray-scale or binary images for license plate detection. Many features are used to select the regions of interest as license plate location such as corner points. One of the methods for detecting the region of interest is to use the Hough transform for extracting vertical and horizontal lines in an image and then mixing them to find some rectangles that correspond to the aspect ratio of the license plate [11], [12].

The simplest way of detecting a license plate is by using morphological functions to find white areas with some block objects (white vehicle's license plate with block characters in it). License plate candidates would be detectable by finding white areas and by using a threshold range to select them [13], [14]. When the images come from an infrared camera, the tiling histogram algorithm or morphological function is very effective for rapid detection [1]. Morphological algorithms give excellent performance in infrared photos, although not in normal color images. They usually detect many candidates of color images; therefore, finding the real candidate not only would be time consuming but also would require other algorithms. Fig. 1(a) shows two sample images that were taken by an infrared camera in daylight. Obviously, these images can be converted by a low-level threshold for binarization into black and white. Then, finding the white areas to detect the location of the license plate would be fast and accurate. Using edge detection and projection to detect a license plate is a well-known method (an edge in gray-level photos is a sudden change in value from white to black or vice versa); when a part of the projection plot or the edge density seems to be similar to a license plate, it is considered as a candidate [3], [6], [15]–[20]. Projection is the most favored algorithm for license plate detection. It is based on alterations of edges in limited areas; thus, transforming images to other domains, such as frequency, wavelet, or curvelet, would be an effective way of detecting the correct location of the license plate [2], [21], [22]. Transforming to other domains involves additional processing and requires more time; therefore, achieving real-time LPR would be difficult, although possible with precise syntax and algorithms. Fig. 1(b) illustrates the concept of projecting algorithms in a strip of a license plate image. Projection algorithms deliver excellent and accurate performance when importing images taken by infrared cameras. In infrared images, the location of the license plate can be easily detected by using simple horizontal projection strips. Usually, in the projection methods, the correct location of a license plate is found in a single-search process [2].

For detecting the location of the license plate, artificial intelligence methods, particularly neural networks, are used. A neural network detects the location of the license plate by blurring and sweeping the image surface via a dynamic window [23], [24]. Template matching is another method, but it is highly static, and its performance is low when the exact license plate's image, its characters, or the plate signature is applied as a template [2], [25], [26]. There are other methods and algorithms that use plate signature [15], fuzzy logic [4], [10], [27], geometric parameters, and window movement [28], [29]. Morphological functions also give acceptable results if the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)

(b)

Fig. 1. (a) Samples of infrared images in daylight. (b) License plate detection through horizontal projection in an image.

license plate is close to the camera [18], [20], [30]. Many previous studies used hybrid methods to achieve a more accurate and faster license plate detection, for example, combining fuzzy logic and neural network [2], [27].

As aforementioned, most algorithms in license plate localization ignore color features and use gray-level or binary processing to localize license plates. Using color feature for vehicle license plate localization is nearer to human vision and decision [7]. Therefore, we propose a new algorithm based on color feature that requires low-level processing to find minimum candidates.

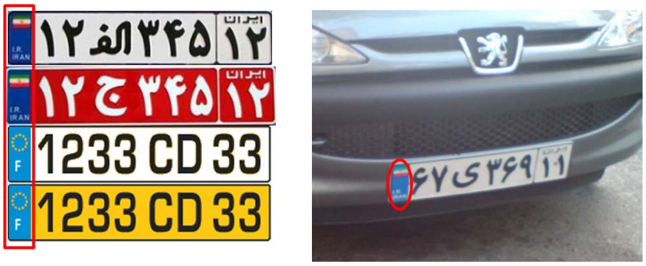


Fig. 2. Samples of license plates from Iran and some European countries.

B. New Approach for License Plate Detection

The proposed method for license plate localization is an optimized template-matching algorithm. It is basically an algorithm that sweeps the entire image surface to find the object via a template or its description [25], [31]. Usually, a template-matching system is not flexible and expandable enough to accommodate imported images of any size [25]. The proposed method, on the other hand, offers optimized template matching that is scale invariant and orientation invariant so that it can detect license plates of any scale or size and in any direction. For using a template-matching algorithm, a template or its description is necessary. Thus, defining a salient and standard template for Persian license plates is important. A suitable feature is the blue rectangle that appears on the side of Iranian and some European license plates. The license plate can be detected by finding the blue rectangle in terms of its standard aspect ratio. Its detection is facilitated by its color-geometric description as a mask. Fig. 2 shows the blue rectangle on license plates of Iran and some European countries.

Because color information is used to detect the location of a license plate, converting the whole image into gray level, black and white, or binary level is not necessary. Detecting an object within its image has always been difficult in image processing [25]. Generally, template-matching methods sweep an image, pixel by pixel or window by window, for comparing a template with a part of the image using convolution. The part with the lowest error rate is taken as the candidate for matching. This whole process imposes a heavy overhead on the system [31]; therefore, it is necessary to find an approach that is scale and rotation invariant and that can sweep the entire image surface at high speed.

The proposed method uses color feature to localize a license plate. The RGB channel format is a natural feature that correlates all the channels with one another for representing real-world colors. HSV is a component of the color space that includes brightness and saturation of the basic hue. Hue indicates the dominant wavelength of color, and saturation indicates its purity. Highly saturated colors contain a very narrow set of wavelengths that are much more pronounced than similar but less saturated colors [31]. Fig. 3 illustrates the spectrum for highly saturated and less saturated shades of blue. In the RGB color space, because of lack of orthonormality in HSV, the color does not separate from the value, and therefore, the detection of target color becomes difficult.

In the primary step of localization, blue pixel colonies are sought by using hue. Color spaces with separated hue, such as HSV and HSI, are quite useful for seeking. The proposed method seeks blue pixel colonies to find blue rectangles in a license plate's aspect ratio. Converting all image pixels from RGB to HSV is not necessary, but periodic strip line pixels

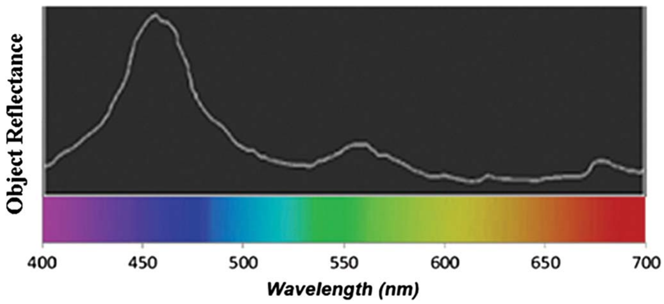


Fig. 3. Example spectrum with a bluish hue.

TABLE I
LIMITATION LEVELS FOR CHOOSING BLUE PIXELS

Level	Limitation type	L	H	φ
1	Rigid	0.6	0.7	0.27
2	Semi-rigid	0.55	0.75	0.24
3	Medium	0.55	0.77	0.21
4	Semi-medium	0.52	0.79	0.18
5	Semi-low	0.5	0.8	0.15
6	Low	0.47	0.83	0.11

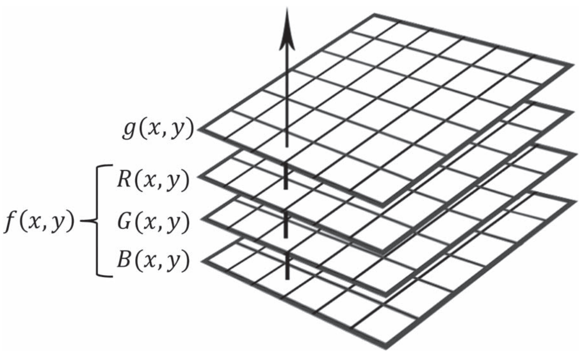


Fig. 4. Finding $g(x, y)$ using three layers of RGB.

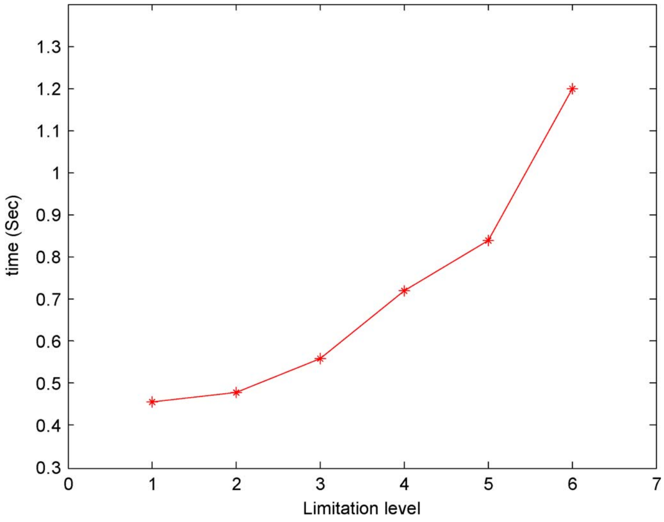


Fig. 5. Effect of limitation levels on processing time.

are converted on demand for a vertical or a slope search using (1) and γ condition. Equation (1) finds bluish of a pixel; if γ conditions are not true, the pixel is not blue, and it is assumed as "do not care." Equation (1) is shown at the bottom of the page, where R, G, and B are the three RGB matrices, respectively; m is the number of columns (width), and n is the number of rows (length) in an $m \times n$ image resolution. φ determines the low-saturation pixels (white pixels) and the low-value pixels (black pixels) that need to be disregarded. φ is the initial parameter that is adjustable by the user in special cases; it is 0.21 by default. Different levels of φ are initially defined in Table I.

Equation (2) is the limitation for selecting pure blue; $g(x, y)$ is a matrix of pure blue pixels that applies to license plate localization. Fig. 4 shows how $g(x, y)$ is achieved by using the three layers of RGB. A rigorous limitation to selecting pure blue enables faster search for localization.

If a detection system with severe limitations cannot detect the blue rectangle, then unclosed limit will have to be applied. Table I shows different limitation levels for selecting blue pixels. Fig. 5 illustrates the effect of limitation levels on processing time. Obviously, using rigid level facilitates fast localization in the proposed system. Fig. 6 indicates that semilow level ensures the best detection rate. Consequently, medium level gives best results for detecting more license plates in less response time. Thus

$$g(x, y) = \begin{cases} 1, & L \times 6 < B_p(x, y) < H \times 6 \\ 0, & B_p(x, y) \le L \times 6 \\ 0, & B_p(x, y) \ge H \times 6 \end{cases}$$
$$x \in \{0, \ldots, (n-1)\}, y \in \{0, \ldots, (m-1)\}. \quad (2)$$

Periodic sweeping, in the vertical or the slope strip of image pixels, is proposed to detect the location of a license plate in $f(x, y)$. The dashed line in Fig. 7(a) shows how to carry out vertical periodic search. When the search index in the

$$B_p(x, y) = \left\{ \left[\frac{R_{(x,y)} - G_{(x,y)}}{\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) - \min(R_{(x,y)}, G_{(x,y)}, B_{(x,y)})} + 4 \right] \middle| \gamma \right\}$$
$$\forall\, x \in \{0, \ldots, (n-1)\}, \forall\, y \in \{0, \ldots, (m-1)\}$$
$$\gamma = \left\{ [\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) == B_{(x,y)}, B_{(x,y)} \ge \varphi] \wedge \left| \frac{\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)}) - \min(R_{(x,y)}, G_{(x,y)}, B_{(x,y)})}{\max(R_{(x,y)}, G_{(x,y)}, B_{(x,y)})} \right| \ge \varphi \right\} R, G, B, \varphi \in [0, 1], B_p \in [0, 6), \quad (1)$$

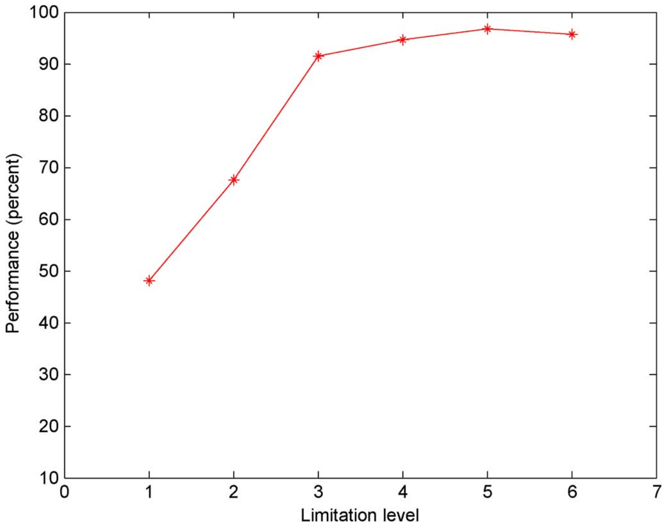
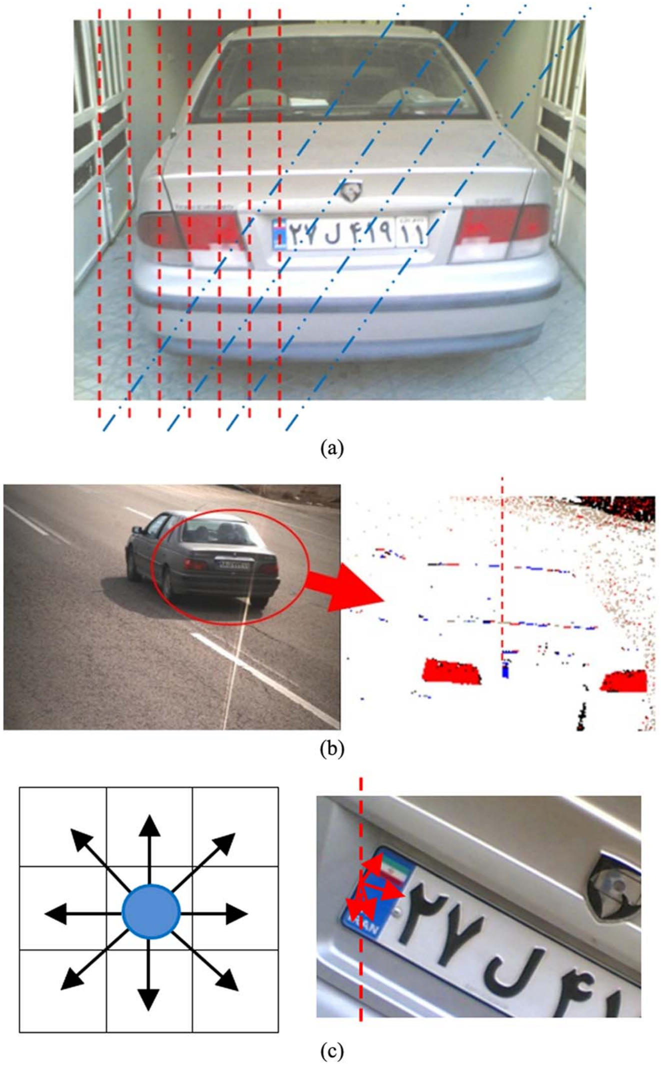


Fig. 6. Performance rate of limitation levels.


(a)

(b)

(c)

Fig. 7. (a) Vertical and slope sweeps to find color-geometric match. (b) Search index encounters a blue pixel. (c) Watershed to find the blue colony.

strip line encounters a blue pixel, it expands the blue pixels using an eight-connectivity connected component to find all the available blue adjacent pixels, as shown in Fig. 7(c). The hue of each adjacent pixel is calculated by (1) on demand and signed in target frame $g(x, y)$ as “1” for blue or “0” for do not care. As Fig. 7(b) shows, when the search index finds the first blue pixel, the proposed system expands to complete a colony of blue pixels [see Fig. 7(c)]. In simple terms, the colony of blue pixels is a connected component of blue pixels. The proposed system is biased to reject incorrect colonies. The bias is the approximate ratio of the license plate width to the image width. The bias range depends on the resolution of the image and the approximate distance of the vehicle from the camera. By using the bias range, small and big colonies (e.g., blue vehicles) sign as do not care before their shape is recognized. Fig. 8(a) shows a European license plate on a blue vehicle/background. The false colonies have been rejected using the bias [approximately 1/5 for Fig. 8(c)] and colony size ratio. The bias accepts colonies with a size of $(20 \times 10) \pm 50$, in which the image size is 400 × 600 and the bias is 0.2. Usually, license plates are separated from the background with borders where the background is pure blue. The border is obvious in Fig. 8(b), but in some cases, when the vehicle is far from the camera, the border is not clear; therefore, in such cases, the proposed method is inapplicable to license plate localization. The proposed system can provide correct results without bias, but it works faster with proper bias value. When a colony of blue pixels is found, it is checked that its dimensions match with the bias range; then, a particular moving window determines the shape of the colony by moving the window along the length of the rectangle obtained by (3). Because the shape of a standard license plate is rectangular, the length-to-width ratio should be the same as that of the blue rectangle. The length-to-width ratio of a Persian license plate, also its blue rectangle, is always constant (see Fig. 9).

Some surplus objects (e.g., flag of Iran) in the blue rectangle render shape recognition difficult. Interference by such objects is more when the camera is nearer to the license plate. For finding the correct rectangle, window matching has been applied. Fig. 10 shows how the mask and the moving window have been used to find a blue colony shape. The moving window should move in the direction of the rectangle’s length. The direction of the moving window is achieved by using the maximum distance between pixels in a colony. This is determined by

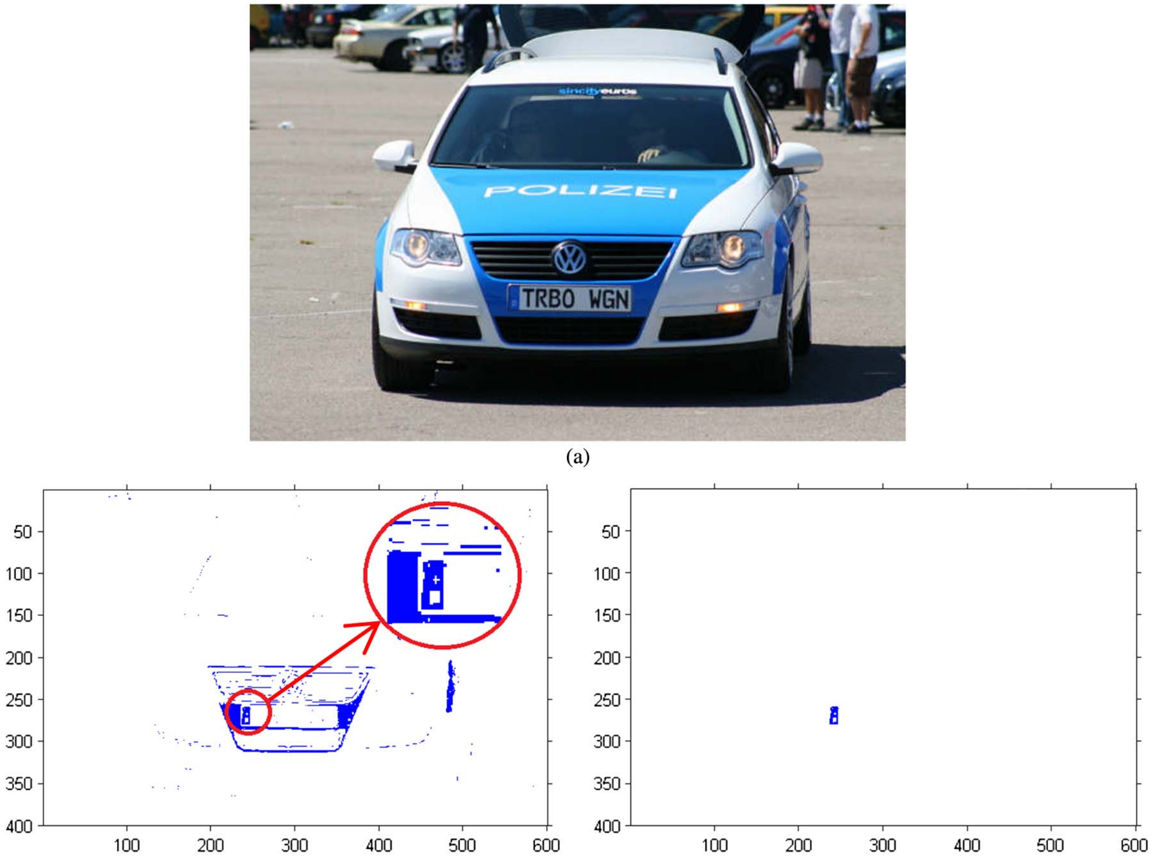
$$\beta_1 = \max\left(g(x,\sigma)\right) \qquad \beta_2 = \min\left(g(x,\sigma)\right)$$

$$\beta_3 = \max\left(g(\sigma,y)\right) \qquad \beta_4 = \min\left(g(\sigma,y)\right)$$

$$\forall\, g(x,y) \in \text{Colony}_A, \sigma \in \mathbb{N}$$

$$m_A = \begin{cases} \tan^{-1}\left(\frac{\beta_{3y}-\beta_{4y}}{\beta_{3x}-\beta_{4x}}\right), & \beta_{4y} \geq \beta_{2y} \cup \beta_{1y} \\ & \quad \geq \beta_{3y} \cup \beta_{2x} \geq \beta_{4x} \\ \tan^{-1}\left(\frac{\beta_{1y}-\beta_{3y}}{\beta_{1x}-\beta_{3x}}\right), & \beta_{3y} \geq \beta_{2y} \cup \beta_{1y} \\ & \quad \geq \beta_{4y} \cup \beta_{3x} \geq \beta_{2x}. \end{cases} \tag{3}$$

Fig. 11(a) shows the direction of the blue rectangle for the moving window. The width of the window is the width of the blue rectangle, and a window seed is defined in one third of



(a)

(b)

(c)

Fig. 8. Sample of a European license plate with a blue background. (a) Original image. (b) Finding blue colonies. (c) Rejecting not in bias range colonies.

the window width. The window seeks the blue rectangle in the direction of m and counts rectangle length when the seed is completely blue, and each row has at least one blue pixel. Equation (4) indicates the condition of window movement, i.e.,

$$\mu \equiv \int_{\delta}^{\delta+1} g(y)dy \geq 1 \quad \forall \delta \in c - \frac{w}{2} < \delta < \frac{w}{2} + c$$

$$\omega \equiv \int_{c-S}^{c+S} g(x)dx \geq S \times 2$$

$$\text{where} \quad P\left((\mu) \cup (\omega)\right) = 1 \leftarrow \begin{cases} h(c) = 1 \\ g(a,b) = 1 \end{cases}$$

$$\forall a, b \in c_x - \frac{w}{2} \leq a \leq c_x + \frac{w}{2},$$

$$c_y - \frac{w}{2} \leq b \leq c_y + \frac{w}{2} \qquad (4)$$

where c is the center of the window, w the width of the window, and S is the width of the seed. $P(\omega)$ and $P(\mu)$ are the probabilities of μ and ω, respectively. $h(x)$, a temporary matrix of the same size as that of $g(x)$, holds center information of the colony under process. For length calculation, $g(x)$ provides information on all the available colonies, including the current one, but $h(x)$ provides only seed information for the colony under process. The maximum continuous movement of the seed (in pixel) plus the window width is the length of the rectangle, as shown in

$$\text{rec}_{\text{length}} = w + 2 \times \max(\beta) \quad \text{when} \sum_{c-\beta}^{c+\beta} h(c) = 2\beta \qquad (5)$$

where $\beta \in \mathbb{Z}^+$ is half of the continuous movement of the seed in $h(x)$. Maximum available β indicates half of the maximum continuous movement of the seed. After finding Colony$_A$, if it falls outside the bias range, or if the length-to-width ratio of Colony$_A$ does not match with the license plate, then the proposed system gives up processing and continues with vertical or slope search, starting from the last search index position. All pixels in the last search that contributed to the processed colony are changed to do not care. When the search index encounters do not care pixel, it jumps until it arrives at an unknown position. Checking of the pixel type has been done on target frame $g(x, y)$, as shown in

$$g(x,y) \begin{cases} 0 & \text{do not care} \\ 1 & \text{blue pixel} \\ NaN & \text{unknown pixel.} \end{cases} \qquad (6)$$

The proposed method detects rotated license plates by finding the blue pixel colony and its direction. The direction is applied to find the shape of the colony and to crop the license plate in the correct orientation. Finding the shape of colony in the primary step of sweeping by the moving window can be

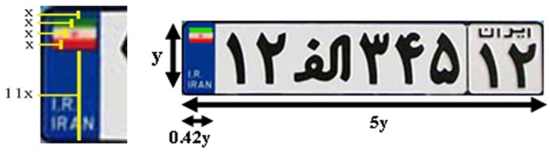


Fig. 9. Standard ratios in an Iranian license plate and its blue rectangle.

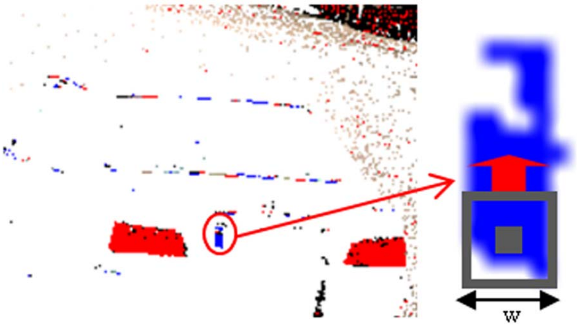
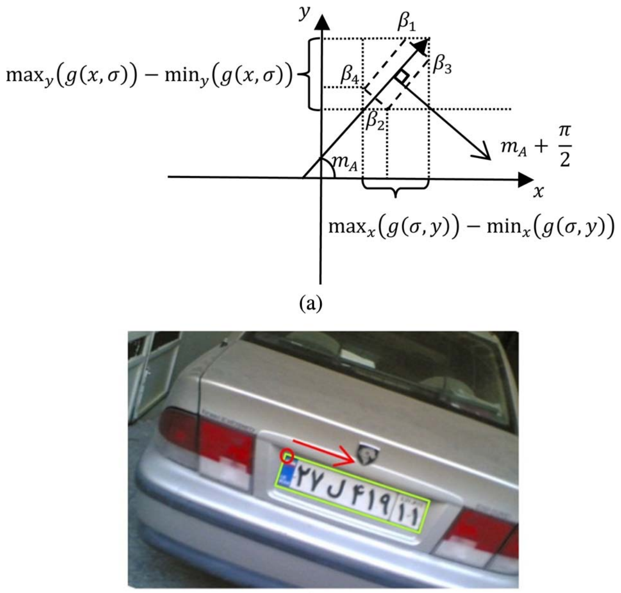


Fig. 10. Moving window in the direction of m_A to find length-to-width ratio in a blue pixel colony.


y
β_1
β_3
β_4
β_2
$\max_y(g(x,\sigma)) - \min_y(g(x,\sigma))$
m_A
$m_A + \frac{\pi}{2}$
x
$\max_x(g(\sigma,y)) - \min_x(g(\sigma,y))$

(a)

(b)

Fig. 11. (a) Calculation of the blue rectangle direction. (b) Cropping in the same direction as that of the width of the blue pixel colony.

disregarded to achieve higher speed where the direction of the license plate is available. Therefore, before using (3) and (4), some false colonies are rejected by the length-to-width ratio of the colony and the remaining colonies confirmed by the moving window. Usually, license plate photos are taken in the direction of the x-axis so that the license plate length is perpendicular to the x-axis or has an ignorable angle. Therefore, vertical sweeps can be applied to find blue candidates, as illustrated by the dashed line in Fig. 7(a). Equation (3) imposes additional processing on the system where the direction of license plate is available. The system acts faster when the given angle is used in speed control station photos; therefore, (3) is disregarded for rotated license plates; in addition, (4) applies to less colonies. In the primary step of analyzing a colony, the proposed system can reject or accept the colony by the bias range and the aspect ratio of the colony in the given direction so that many false colonies are rejected before using the moving window in (4).

In a fixed speed control station or camera, the photos may be taken at the same angle. For example, if a station takes photos with 10° skew to the x-axis, all the other photos from that station are taken with the same skew, which is approximately 10°. Assuming that hardware correction (changing the head angle of camera at speed control station) is impossible or somewhat difficult, the process of image rotation is disregarded for preserving the system's performance and speed. Instead, the image is swept by the same degree skew of camera. It means that slope sweeping is in the same direction as that of the license plate length while the orientation of the license plate is available. A sample of slope sweeping in $+45^\circ$ is shown by the dash–dot lines in Fig. 7(a). The proposed system responds faster compared with the process of searching colonies in all the available directions using (3) and (4). As in the case of vertical sweep lines, in slope sweeping also, a blue colony can be rejected or accepted directly by the bias range and the aspect ratio of the colony in the same direction as that of the sweep line. Thus, the system's speed and performance increase. The slope search can be applied at the m_A degree where the camera is fixed and always takes photos with the same degree.

C. Cropping Car Plate

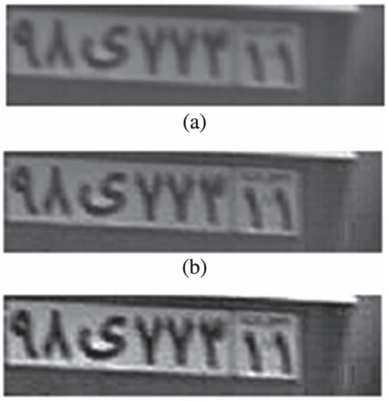
As shown in Fig. 9, the license plate's aspect ratio is standard and constant, and it can be determined by applying a vertical or a slope search. Although the length and the direction of a license plate are available from the vertical or the slope sweep, the width of the Iranian license plate can be obtained by a simple multiple using

$$\text{LicensePlate}_x = 5 \times \text{rec}_{\text{length}}$$
$$\text{LicensePlate}_y = \text{rec}_{\text{length}} \qquad (7)$$

where LicensePlate_x is the width of the license plate, and LicensePlate_y is its length. Cropping begins from the top left coordinate of Colony_A and in the same orientation as that of the blue rectangle width, as shown in Fig. 11(b). Cropping in the direction of the blue rectangle causes license plate images that are straight and without any angles. Fig. 12(a) illustrates how the plate looks after cropping. The proposed system avoids any resizing because of the use of the ratio instead of the constant size. Therefore, all images that include a license plate, regardless of the size of the image or the license plate, can be processed. The proposed method can detect more than one plate in an image (if the image includes several license plates).

D. Confirming License Plates

After finding all the candidates through vertical or slope search, the candidates are cropped from the original image. However, one question remains: "Which one is the correct license plate?" The proposed system localizes all the available



(a)

(b)

(c)

Fig. 12. Samples from cropped car plate after detection. (a) Original cropped license plate. (b) Applying histogram equalization and gamma adjustment. (c) Applying the Laplacian filter.

blue rectangles in the same aspect ratio of the license plate and crops them. The projection analysis confirms the real plates in the primary step. The projection method has to include at least eight independent peaks (because a standard Iranian license plate includes seven numbers and a letter), as shown in Fig. 1(b). Peaks in projection are achieved by two simple arrays and by finding semilocal maximum and minimum. If the peaks are recognized, the system could successfully find the real license plate; otherwise, the system rejects and deletes the candidate from the candidate list.

Finding vehicle lights or bumper area by texture, shape, and color analysis is a useful way to detect the real license plate. Considering that a few images contain fake blue rectangles in the same shape and aspect ratio of the license plate and that, statistically, the proposed system rarely detects fake blue rectangles as a license plate (1.27%), the process of finding vehicle lights or bumper is unnecessary and time consuming.

III. Recognition of License Plate Characters

The license plate, after the localization and cropping from its original image, has to go through the recognition process for classifying and recognizing its letters and numbers. The recognition process involves three steps. The first step is essential and deals with noise removal and improvement of image quality by using image processing filters, such as histogram equalization and Laplacian filters, for creating sharp edges and applying morphological functions to present the license plate in optimal position for character recognition. In the second step, the image is converted into a binary image, and all the characters are separated through a segmentation function, followed by another round of the image quality improvement. The last step is to recognize separated characters through fast and accurate classifiers, leading to the final result. License plate characters are recognized by an SVM, with a homogeneous fifth-degree polynomial kernel and one-stands-all strategy. Before applying the main SVM, a simple decision tree, with 28 features, tries to group and classify the characters. If the decision tree fails to deliver a highly reliable result, the characters in smaller groups would be passed through an SVM for reliable classification.

A. Image Quality Improvement

The main target of this step is to make the clearest and the most readable position of the image. When more details of each character become visible, the performance of the recognition system will increase. Edge sharpening, or converting smooth edges to nonsmooth ones, is the most important part of this step. By forming sharpened edges, all the important details remain in the license plate's characters and their edges. The Laplacian filter is the main filter that is used for edge sharpening, as shown in the following:

$$\check{f}(x) = f(x) - \omega \cdot f''(x)$$
$$(\nabla^2 f)(x,y) = \frac{\partial^2 f}{\partial^2 x}(x,y) + \frac{\partial^2 f}{\partial^2 y}(x,y). \qquad (8)$$

The edge is sharpened by subtracting a certain fraction $\omega \geq 0$ as a weight factor of the second derivative $f''(x)$ from the original function $f(x)$. Two-dimensional sharpening is accomplished by the second derivative in the horizontal and the vertical combined by the so-called Laplace operator. ∇^2 is a Laplace operator of function $f(x,y)$ that is defined as the sum of the second partial derivatives along the x- and y-directions [31].

For contrast and intensity/gamma adjustments, histogram equalization is used to reveal edge details to the extent possible. Fig. 12(b) shows an example of histogram equalization, and Fig. 12(c) shows the Laplacian filter on a license plate for sharpening edges.

B. Converting to Binary Images

A threshold value is required for converting the license plate into a binary image. An incorrect high or low threshold causes connectivity or discontinuity between license plate characters; hence, finding a tradeoff threshold value is important. This is generally done by using a histogram to find entropies and the probability of distribution between objects/foregrounds and background. In other words, the threshold is a value that separates two independent peaks in intensity or color histogram [31], [32]. In some cases, no two peaks (foreground and background) in a histogram are independent, and the peaks are strongly relevant; hence, finding a separator between two peaks is quite difficult. Fig. 13(a) shows a histogram in which the object/foreground and the background form two independent peaks. In reality, Fig. 13(b) shows a license plate histogram in which finding a separator between objects and background is difficult.

Equation (9) indicates a method for binarization without applying histogram. It finds the threshold value by using the local maximum, i.e., Max_L, and the local minimum, i.e., Min_L, of morphological functions [32], as follows:

$$\begin{aligned}&\text{MMT}(i,j)\\ &\quad = 1 \text{ if } 2P(i,j) > \text{Max}_L\left(P(i,j)\right) + \text{Min}_L\left(P(i,j)\right). \qquad (9)\end{aligned}$$

This method is based on an $L \times L$ window template. Local thresholding method is efficient and useful for edge detection and small object extraction. Experience shows that license plate width, divided by 17, is the proper value for window template in license plate binarization. Hence, $W = \text{LicensePlate}_x/17$ is

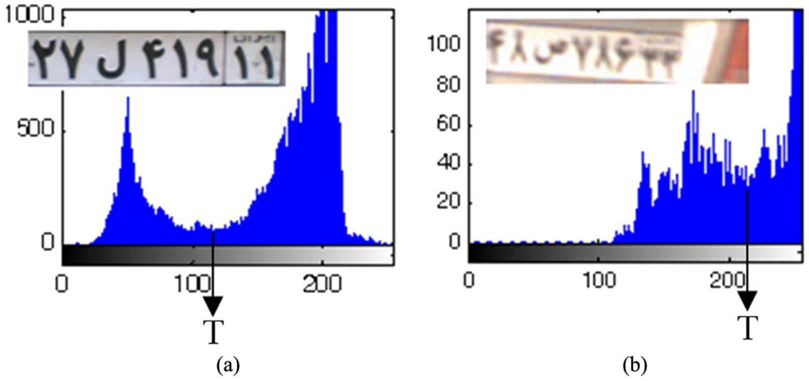


Fig. 13. Finding the threshold value via histogram. (a) Histogram with two independent peaks as the object and the background. (b) Histogram with two related peaks; setting a point as the threshold value is difficult.

used for finding the size of the window template. The template window size is dynamic and changeable by imported image size. The method described here is highly sensitive to noise, and hence, noise reduction filters must be used before applying this method.

Object-attribute-based methods, such as watershed, K means, and fuzzy, are used to extract all the available characters without using threshold value and binarization [14], [32], [33]. By these methods, the contours of the characters in the license plate are found, and the characters are extracted; then, they are passed on to the recognition step. PirahanSiah *et al.* [30] proposed a method based on multilevel thresholding that finds the most suitable threshold value using the peak signal-to-noise ratio (PSNR). PSNR is widely used as a stopping criterion in multilevel thresholding method for segmenting images. This method is time consuming and needs roughly 2 s for finding the best thresholding value.

The minimum error thresholding (MET) method is used to separate the object from the background in a license plate [30]. In MET method, the foreground and background class conditional probability density functions are assumed to be Gaussian [32], [34]. The error expression can be interpreted as a fitting error expression to be minimized, which is calculated as follows:

$$T_{\text{opt}} = \arg\min\left[P(T)\log\sigma_f(T) + (1-P(T))\log\sigma_b(T) - P(T)\log P(T))\log(1-P(T))\right] \quad (10)$$

where $\sigma_f(T)$ and $\sigma_b(T)$ are the foreground and background variances for each choice of T, respectively.

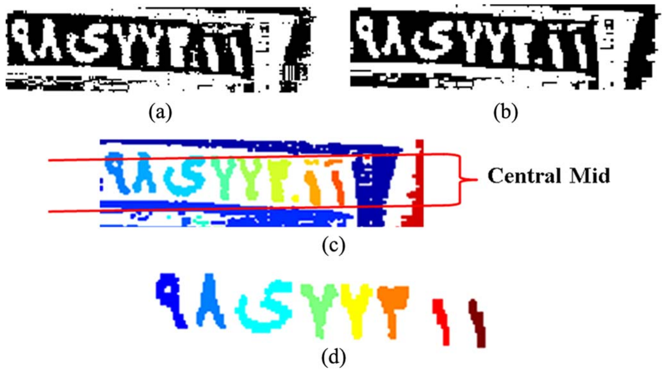
Fig. 15(b) shows connectivity and discontinuity of license plate characters that result from using an improper threshold value. The proposed system does not use just one threshold value; instead, it finds the basic threshold value by MET method from which four more threshold values, beginning with the basic threshold value, have been generated at ±0.04 interval, in the range of [0, 1]. The system converts the images into binaries by priority list, as shown in Table II. The basic/parent threshold value (priority level 1) indicates high priority, and the other values, in their order, represent progressively lower priority. The proposed system converts the license plate image into a binary one using the first value in the threshold priority list. The threshold value is correct when all the eight characters (as those in the case study of Persian license plate) are recognized with high reliance (i.e., more than 50% accuracy). If the results are otherwise by way of feedback, the next value in the threshold priority list is used. If all the threshold values are used for binary conversion and the license plate characters cannot be accurately recognized, then the system assumes the current plate as unreadable, and the final recognition result is generated by voting strategy for each character. Table II shows the rate of correct binarization using the threshold value priorities from the data set of 300 recognized vehicles. More than 94% of the license plates have been correctly binarized by the basic threshold value. The overall response time of the system decreases when other threshold values in the priority list are used.

TABLE II
CORRECT BINARIZATION BASED ON THRESHOLD VALUE PRIORITIES

Priority level	Threshold value	Correct detection
4	MET – 0.08	1.05%
2	MET – 0.04	2.14%
1	MET [34]	94.86%
3	MET + 0.04	1.27%
5	MET + 0.08	0.68%

C. Removing Surplus Components

For correct recognition, the binary license plate should be free from noisy dots and surplus objects. In the first step, the proposed system uses morphological functions (i.e., erosion and dilation) to delete small noisy dots and to correct edges, as shown in Fig. 14(b). The size of the mask filters for morphological functions is determined by the width resolution of the license plate. The mask is a rectangle, and width/40 is its size in pixel. In the next step, all the objects are labeled by a connected component labeling function, as shown in Fig. 14(c). All the labeled components are assessed in terms of size and amount of pixels so that they must be within the given range of minimum to maximum. For removing surplus components in a license plate, the given range for the size or amount of pixels is determined by the width of the license plate in pixel. In addition, the length-to-width ratio of each component is used to delete surplus components. At least two thirds of each component pixel must be in the three fifths of middle license plate, which is known as the central mid area. The central mid area shown in Fig. 14(c) must include at least two thirds of each component pixels. Using the limitations cited, the surplus and noisy components are removed. Fig. 14(d) shows a license plate that has been cleared after applying the limitations.



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

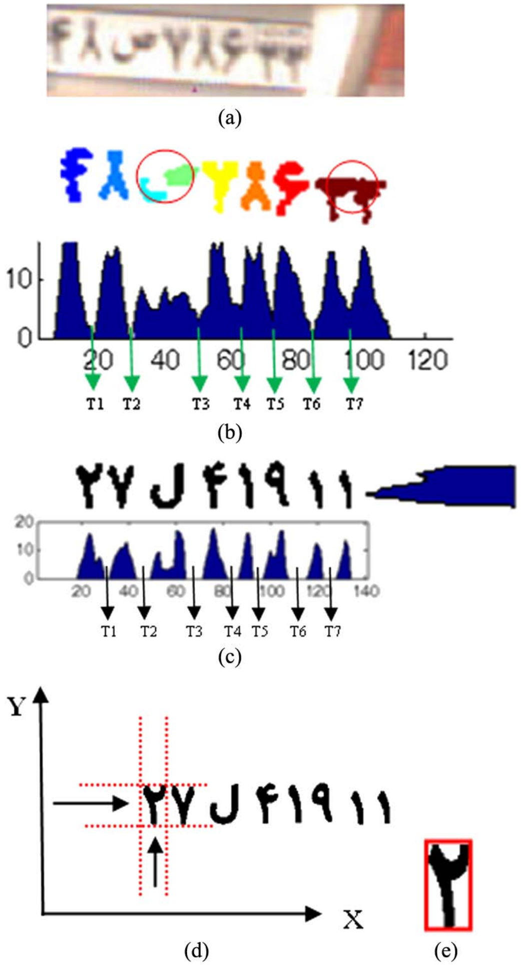
Fig. 14. Removing surplus components. (a) Binarized license plate. (b) Applying morphological functions. (c) Labeling by a connected component and central mid area. (d) Applying rules, limitations, and conditions for removing surplus components.

In some cases, character details would be wrongly removed as noisy components, for example, the dot of letter “ب” is removed by clearing methods. In reality, this is not important because other similar letters, such as “پ، ت، and ث،” are not used in Iranian license plates. Another example is the dot of letter “ج،” which is removed as a noisy component, but again, other similar letters, such as “چ، خ، and ح،” are not used in Iranian license plates. Nevertheless, the proposed system will have to be suitably modified whenever such similar letters are likely to be used in Persian license plates.

D. Character Extraction

To recognize the license plate, the system has to extract its characters; there are several ways for character extraction. In Iranian license plates, the location of each character is fixed; hence, the simplest way for extraction is to use a moving window in the constant area. Fig. 15(d) illustrates how the moving window on the constant area crops a character. The cropped character is shown in Fig. 15(e). The Hough transform is another method to extract the characters’ contour [35]. Morphological analysis is a widely applied method for character extraction [19], [36]. Connected component labeling, as a morphological function, is a fast and easy method for extracting characters in a clear license plate. The proposed system, in the first step, extracts eight objects (as those in the case study of Persian license plates) by applying connected component labeling.

After applying the connected component, some discontinuity and connectivity may show up between the license plate’s characters [see Fig. 15(b)]. In addition, a deadlock can be seen, where all the threshold values for binarization in the priority threshold list create connectivity or discontinuity in the characters. In Fig. 15(b), increasing the threshold value to solve discontinuity contributes to more connectivity in other characters. Similarly, decreasing the threshold value to solve connectivity contributes to more discontinuity in the letter. Therefore, in this case, character extraction via connected component is not possible. The other solution is to use projection for character extraction [36]. Fig. 15(b) shows a projection that has extracted eight license plate characters by separating eight peaks. Projection graph is a proper method for character extraction. It is fast, regardless of connectivity or discontinuity between license plate characters [12], [36]. The projection extraction method finds independent peaks and separates them. Vertical and horizontal projections try to extract all characters. Fig. 15(b) and (c) illustrates the extraction of characters by using projection plot. It is shown that finding the peaks in Fig. 15(b) is more challenging than in Fig. 15(c). Peaks in projection are found by two simple arrays and semilocal maximum–minimum.



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

Fig. 15. Samples of character extraction. (a) Original cropped license plate. (b) Discontinuity and connectivity in the license plate characters; a projection plot extracts eight license plate characters by separating eight peaks. (c) Simple license plate projection with eight independent peaks. (d) Moving window in constant area and trimming edges of each character. (e) Trimmed character.

After projection, eight characters have been extracted. A simple function trims all the four sides of each character. The top and bottom of each character are trimmed by horizontal sweep and the left and right by vertical sweep [see Fig. 15(d)]. Fig. 15(e) shows the extracted and trimmed number that has been passed to the recognition section.

The whole process of character extraction can be summarized as follows. First, the proposed system uses connected component labeling for character extraction. If the extraction is not successful and the system fails to find the eight objects, as required for the Iranian license plate, because of connectivity or discontinuity, the system tries to carry out extraction through projection method. If the projection method also fails, the system uses feedback for conversion into a new binary image from the license plate via another threshold value in the threshold priority list.

E. Character Recognition

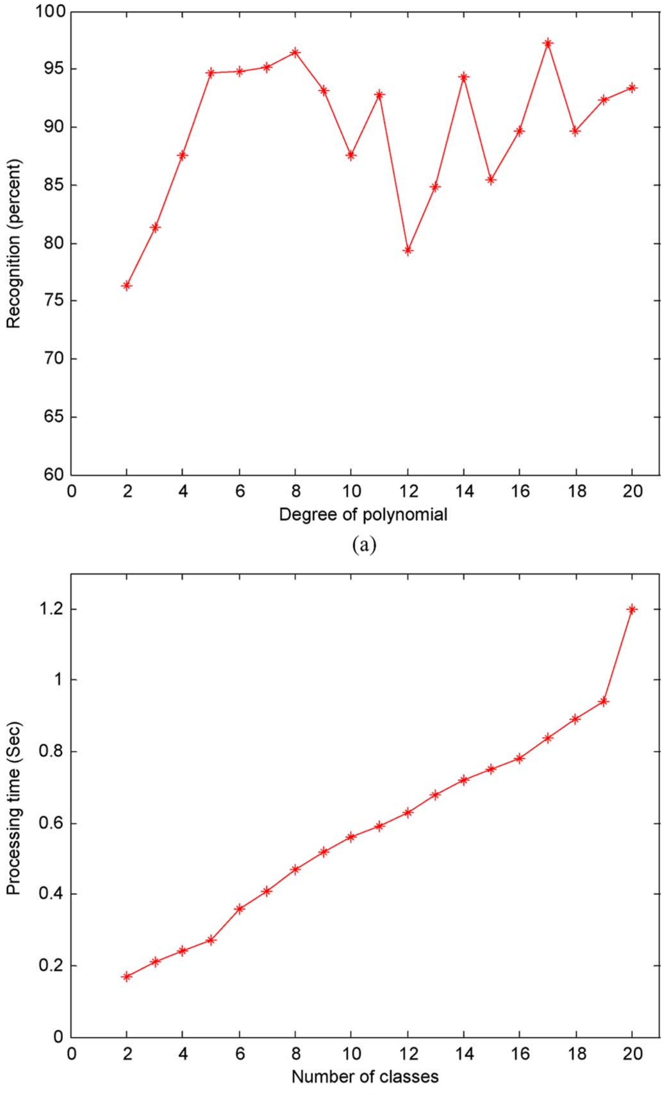
Each character extracted from the license plate will be separately recognized. Four methods are commonly used for classifying a character in the LPR system. Neural networks are well known in ALPR as a robust classifier [2], [14], [19]. They are flexible; in addition, they show high performance and recognize efficiently where a good set of features has been selected and trained by a proper data set. Template matching is another way for recognition; however, it is not flexible enough to deliver accurate results in all positions and conditions of a character [25], [37], [38]. The hidden Markov model also is used as a license plate classifier [12]. The SVM is a strong and fast classifier for real-time classification [39], and the same has been used in the proposed system as the main classifier. An SVM with a nonlinear homogeneous fifth-degree polynomial kernel and a one-stands-all strategy has been trained. Fig. 16(a) shows different degrees of polynomial function in SVM that has been trained for classifying nine classes of numbers in Iranian license plates. The least complicated function that provides acceptable result is the fifth degree. The best result provided by a polynomial function is the one with the seventeenth degree. SVM is a robust classifier, but as Fig. 16(b) shows, it decelerates when the number of classes increases. The SVM is accurate and fast when the number of classes is fewer than six for recognizing Persian numbers. Therefore, the proposed system uses a hybrid classifier, comprising a decision tree and an SVM, for providing acceptable performance. A decision tree is a simple classifier that uses some features, rules, and conditions for classification. It divides license plate characters into smaller groups, which are recognized by the SVM with high speed and reliability.

The Iranian license plate has eight characters, among which seven are numbers and one is a letter. The third character in the license plate, from left to right, is a Persian alphabet letter; therefore, the letter classifier is different from a number classifier. The last two numbers in the Persian license plate are smaller than the other numbers; hence, their classifiers are trained by a separate data set for more accurate recognition. For SVM training, the structure and features for the normal numbers and the two small numbers are the same, but the training data set and rules for the decision tree are different.

First, the decision tree groups the numbers into smaller sets and then tries to recognize, to the extent possible, those clear numbers by using some simple features and rules. The SVM recognizes the numbers in each group after grouping by the decision tree. It uses a maximum of three classes for each group instead of nine classes for all the numbers. In addition, three SVM models have to be trained, instead of one, for all numbers. Therefore, the SVM requires more memory on the microprocessor or digital signal processor boards for recognition.

The decision tree divides the Persian numbers into eight groups: the first group has “۱”; the second group has “۲, ۳, and ۴”; the third group has “۵”; the fourth group has “۴ and ۶”; the fifth group has “۷”; the sixth group has “۸”; the seventh group has “۹”; and the eighth group has “۹ and ۱.” The decision tree recognizes “۵, ۷, and ۸.” It recognizes number “۹” by its hole and head ball, but in reality, “۹” is very much similar to “۱,” particularly when vehicles are in a side lane or far from the camera. “۱,” with a bit of noise on its head, is like “۹.” If a decision tree recognizes “۹” or “۱” with high reliance, that is, all decision tree conditions and rules for it went true, these numbers (“۹” or “۱”) have been recognized; otherwise, these numbers (“۹” or “۱”) are sent to the SVM for accurate recognition as a group of “۱ and ۹.” What is previously mentioned is applicable to all numbers and groups. In addition, the decision tree recognizes that “۴” belongs to the “۲, ۳, and ۴” group or the “۶ and ۴” group by using its features and rules. SVM recognizes “۴” as belonging to the “۲, ۳, and ۴” group or the “۶ and ۴” group, depending on which group the decision tree has selected.

The decision tree requires features that are unique to each number. The features should be scale invariant, dynamic, and expandable. The first feature is the length-to-width ratio of each



(a)

(b)

Fig. 16. (a) Performance of different degrees of SVM polynomial function for classifying Persian numbers in license plate. (b) Response time of SVM with a homogeneous fifth-degree polynomial kernel and one-stands-all strategy in different classes.

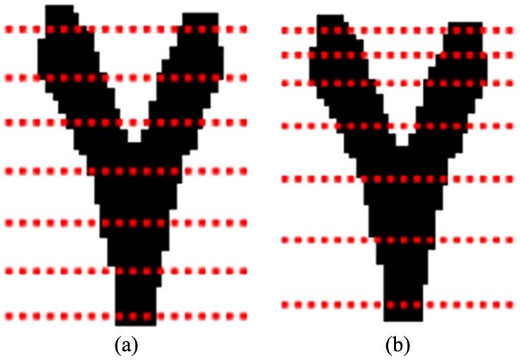


Fig. 17. Samples of features for the decision tree. (a) Horizontal search lines for edge changing. (b) Ratio of object pixels to background pixels on the strip. The density of information is greater at the top of the number's image than at the bottom of it.

number. By this feature, the decision tree recognizes “۱” easily and reliably. Periodic vertical and horizontal searches to find changes in edge are useful features. Fig. 17(a) shows horizontal line searches on number “۷,” with 14 sweep lines for each character. Each sweep line in Fig. 17(b) indicates the ratio of object pixels to background pixels on the line strip. Statistical data and experience show that gathering the lines at the top of each Persian number provides more useful information in determining the ratio of object pixels to background pixels for recognition [see Fig. 17(b)]. Fig. 17(a) shows the number of edges in line strips. For example, in number “۷,” the three top lines show four changes in the edge, and the four bottom lines show two changes.

The next feature is the Euler number, which is the same as the number of holes in a binary image. It is calculated by morphological function. The ratio of holes area to the entire area is another feature, which can be determined when the system calculates the Euler number. Other simple features, such as pressing ratio (ratio of object pixels to all pixels), are useful for accurate recognition.

After grouping by the decision tree, one of the two following situations may arise: either the number has been correctly recognized or it has to pass through the SVM for recognition. In the first situation, the result is available, and the proposed system shifts to recognize the next character, whereas in the second situation, the character in a given group passes to the SVM classifier for recognition. Three models are trained in the SVM for recognizing license plate numbers. The first model classifies “۲, ۳, and ۴”; the second one classifies “۴ and ۶”; and the third one classifies “۹ and ۱.” The situation is most complicated if the SVM had to classify all numbers together. However, classifying the three models based on grouping by decision tree is excellent for a fast and accurate classification.

Like the decision tree, the SVM requires feature extraction from each character. Dividing a character into some tiles and then using the ratio of object pixels to background pixels of each tile is a proper feature for recognition. A number's image is divided into 24 segments by a 6 × 4 mask (see Fig. 18).

Using other features such as gravity point, horizontal search lines (same as the decision tree features), vertical search lines, and pressing ratio enhances the performance and accuracy of the recognition system.

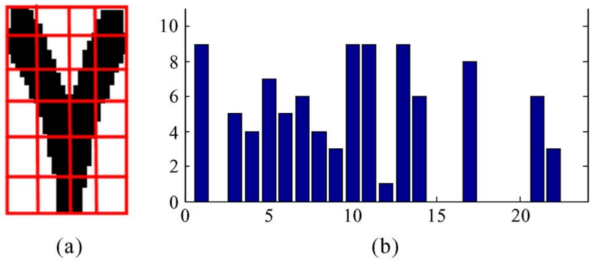


Fig. 18. Segmentation by a 6 × 4 mask and its bar plot of 24 features.

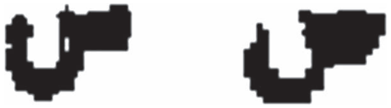
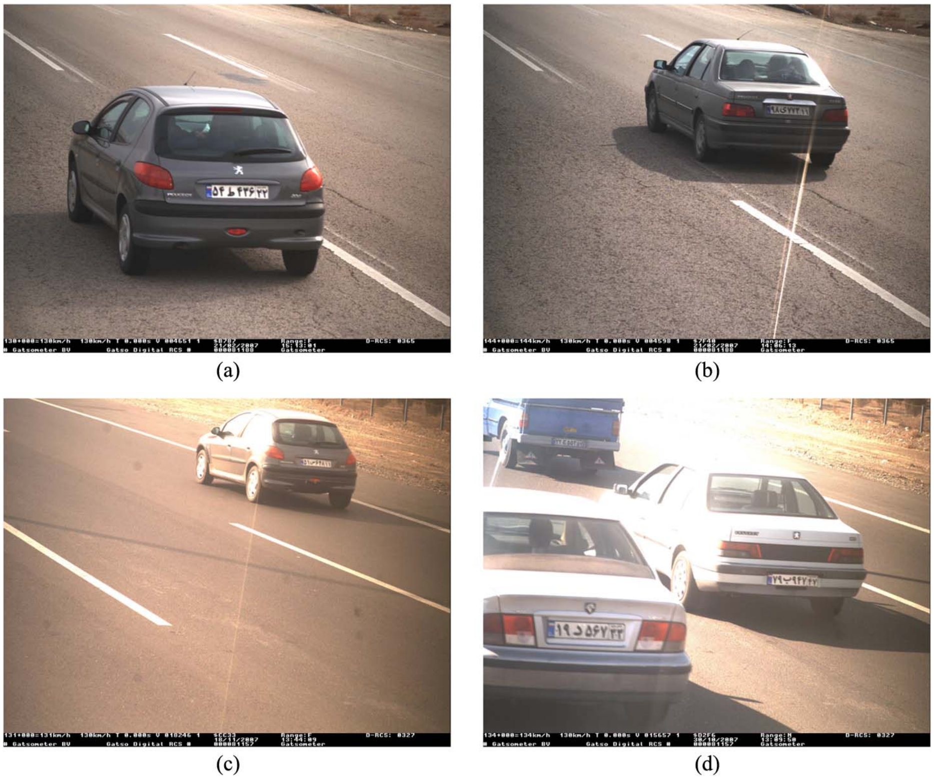


Fig. 19. Similarity between “ص and “س.”

The phases for alphabet letter recognition are similar to those for number recognition. Thirteen alphabet letters, including “ب, ل, و, د, ص, س, ن, ه, م, ق, ط, ج,, and ی,” are used in private Iranian license plates. Two alphabet letters, namely, “ت and ع,” are applied for public license plates, and the letter “الف” is applied for government license plates. The decision tree uses some features to classify letters, including length-to-width ratio, Euler number, vertical–horizontal search lines, and projection. The decision tree recognizes “ب, ل, ه, ج, ط, and د” directly. It generates three groups of letters, including “ص and س”; “ن, ی, and ق”; and “م and و.” The SVM recognizes letters in each group and extracts the features by tiling. The ratio of object pixels to background pixels of each tile is the main SVM input feature. The structure is the same as that of Persian number feature extraction. However, number tiling is somewhat different from letter tiling. In the group of “ص and س,” the segmenting tile is 6 × 4 instead of 4 × 6; moreover, in groups “ی, ن,, and ق” and “م and و,” the segmentation tile is 5 × 5. Other features, such as the gravity point and the vertical–horizontal search lines, are effective in terms of system accuracy. Fig. 16(a) and (b) shows the compromised result, and the speed for training an SVM is provided by a homogeneous fifth-degree polynomial kernel. Thus, it follows that higher degrees increase the response time of the system, but with no notable increase in system accuracy.

Public, private, or government license plates are recognized by a color histogram in the detection section. In Iran, the government license plate has a red background, and the public license plate has a yellow background beside the blue rectangle (see Fig. 2). A hue histogram, after license plate localization, reveals the type of license plate. This feature is very useful because Iranian government vehicle license plates are the only license plates that use the letter “الف”; therefore, the recognition of this letter is not necessary. Public plates use two letters, i.e., “ت and ع,” which are easily and quickly recognized by the decision tree, even by using only the first feature, namely, the length-to-width ratio. The proposed system will have to be suitably modified whenever such letters are likely to be used or added in Persian license plates.

Taking photos of speeding vehicles on highways creates motion noise, notwithstanding professional photographic equipment and high-speed shutters. This noise blurs the difference



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

Fig. 20. Samples from speed control cameras on highways. (a) Vehicle on the speed lane. (b) Vehicle on the middle lane. (c) Vehicle on the side lane. (d) Several vehicles in the image.

TABLE III
SYSTEM PERFORMANCE IN IMAGES FROM SPEED CONTROL CAMERAS ON HIGHWAYS

Conditions (Gatso speed control camera	Data set	Correct detection	Correct recognition	Detection performance	System performance
On the speed lane	500	483	463	96.6%	92.6%
On the middle lane	350	326	305	93.14%	87.14%
On the side lane	250	197	162	78.8%	64.8%
More than one vehicle	50	48	47	96%	94%

between “ص” and “س” and makes them look so similar (see Fig. 19) that their distinction by human vision becomes quite difficult. In some cases, the SVM generates results that are more accurate than even those generated by human recognition and vision.

IV. RESULTS OF EXPERIMENTAL SETUP

The performance of the proposed method has been evaluated under four different conditions: 1) the vehicle is in the speed lane [see Fig. 20(a)]; 2) the vehicle is in the middle lane [Fig. 20(b)]; 3) the vehicle is in the side lane [see Fig. 20(c)]; and 4) the image includes license plates of several vehicles [see Fig. 20(d)]. Table III shows the evaluation results obtained from Gatso control speed cameras on highways under each of four mentioned conditions. The proposed detection method is possible to detect several license plates in an image. The proposed system is considered successful in that it at least detects and recognizes a license plate correctly.

TABLE IV
SYSTEM PERFORMANCE IN LOW-RESOLUTION IMAGES

Conditions	Data set	Correct detection	Correct recognition	Detection rate	System performance
Day light, straight photography	250	242	236	96.8%	94.4%
Night light, straight photography	140	128	101	91.42%	72.14%
Daylight photography at an angle of up to 20°	150	112	93	74.66%	62%

A standard speed control station should cover and focus on a lane, and the speed lane is the target lane for Iranian speed control cameras. Therefore, photos of vehicles in other lanes, taken by this type of speed control camera, are not standard. Our evaluation, based on the speed lane, ensures 96.6% performance in detection rate and 95.85% performance in recognition rate. In addition, the proposed system ensures 92.6% accuracy in the overall system performance.

Detection and recognition rates under three different conditions of photography—daylight, straight; night light, straight; daylight, at an angle of up to 20°—have been assessed, and the results are shown in Table IV. Parked vehicles, which are stable, have been photographed by a normal cell phone camera with

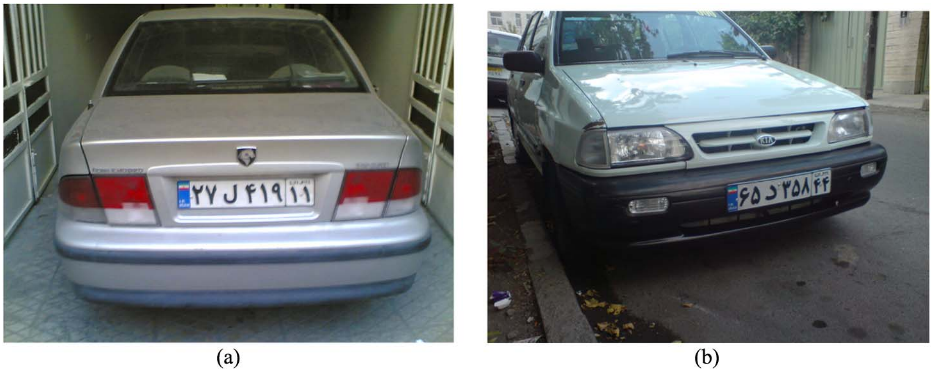

(a) (b)

Fig. 21. Photographic positions. (a) Straight photography. (b) Photography at an angle.

TABLE V
COMPARISON OF DIFFERENT SYSTEM PERFORMANCES

Detection method	Recognition method	Detection rate	System performance
Fuzzy logic [27]	Neural networks	85%	82.62%
Morphological/Adaboost [40]	SAMME	96.93%	90.45%
Morphological [39]	SVM	97.5%	94%
Moving window/projection [28]	Neural networks	—	95%
Morphological [20]	Template matching	97.3%	84.14%
Edge detection and moving window [29]	Neural networks	99.67%	95.77%
Edge statistic and morphological [41]	—	96.5%	—
Edge detection/edge density [6]	—	96.67%	—
Geometric model matching by connected component analysis [26]	Neural networks	91%	86.9%
The proposed method	Decision tree and SVM	96.8%	94.4%

TABLE VI
SYSTEM RESPONSE/COMPUTATION TIME

Conditions (Gatso speed control camera)	Detection time (s)	Recognition time (s)	Total system response time (s)
On the speed lane with given direction	0.52	0.23	0.75
On the middle lane with given direction	0.58	0.24	0.82
On the side lane with given direction	0.65	0.23	0.88
More than one vehicle with given direction	1.12	0.47	1.59
On the speed lane with unknown direction	0.61	0.23	0.84
On the middle lane with unknown direction	0.67	0.26	0.93
On the side lane with unknown direction	0.72	0.25	0.97

a 1.3-megapixel resolution. The distance of the camera from the vehicle ranges from 1.5 to 3 m. Photos have been taken in direct position/without angle [see Fig. 21(a)] or with skew angle [see Fig. 21(b)]. Photography at night is by cell phone flash and under good streetlight conditions.

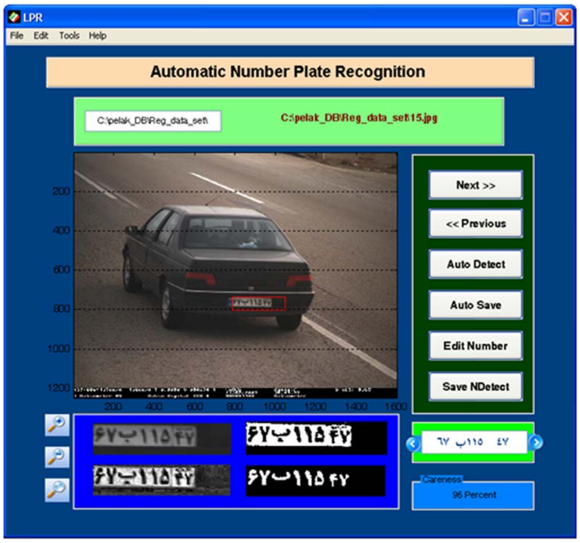
Performance comparison between the proposed system and the other Iranian LPR systems is quite difficult because they used different Iranian license plate images and data sets. Table V presents the comparison between the performance of the proposed ALPR system and that of the other systems in use for recognizing Iranian license plates. The system performance is the ability of an LPR system to generate correct results, including detection rate, segmentation rate, and recognition rate. System performance is calculated as follows:

$$\text{System Performance} = \text{Detection rate} \times \text{Segmentation rate} \times \text{Recognition rate}. \quad (11)$$

The computation time of the proposed system has been assessed, and the results are presented in Table VI. The assessment has been done under two different conditions: the direction of the license plate is given and primary slope sweeping is applied to each "under processing" colony without using (3) and (4); and the orientation of the license plate is unknown and the moving window method is applied to each "under processing" colony. What is shown in Table VI is the average elapsed time for 100 runs in a system with a dual-core 1.7-GHz CPU and 4-GB RAM.

V. CONCLUSION

A new approach for license plate detection, based on the color features in Iranian license plates using periodic vertical or slope sweep, has been presented. The proposed method detects the location of license plates by recognizing its hue and shape. The localization is scale and rotation invariant. The proposed localization system detects all the license plate candidates available in a scene. After detecting the license plate candidates, it extracts them and passes on the image to the recognition section for recognizing the eight characters in each Iranian license plate. The proposed detection system localizes minimum candidates and detects the real candidate (i.e., real license plate) directly in 98.73% of the images.



LPR

Automatic Number Plate Recognition

C:\pelak_DB\Reg_data_set\

C:\pelak_DB\Reg_data_set\15.jpg

٦٧ ب١١٥ ٤٧

Careness

96 Percent

Fig. 22. User interface of the implemented software.

After successful detection, the quality of the license plate image is improved, particularly in the characters' edges, by applying image processing filters, such as histogram equalization and the Laplacian filter. All surplus components are removed by applying some rules and morphological functions. Characters in the license plate are extracted by connected component or projection analysis and passed on to the hybrid classifier for recognition. The hybrid classifier comprises a decision tree and an SVM.

Color images for the proposed system can be generated by normal surveillance photographic equipment; hence, professional photographic equipment, such as infrared camera, is not necessary. The types of Iranian license plates, such as public license plates with yellow background or government license plates with red background, were identified by a hue histogram in the detection section.

The proposed method has been extensively analyzed; in addition, it has been completely implemented and has proven to be practical. Fig. 22 shows the graphical user interface of Iranian ALPR software based on the proposed detection and recognition methods. The main application that has been programmed via Java saves the plate number, the vehicle's speed, the coordinates of the license plate location, the percentage accuracy for each recognized character, the date, the time, and the address and location of the resized original image in a database.

Efforts to provide faster and more accurate ALPR systems will have to continue. A system that recognizes license plates in all the weather conditions—foggy or rainy—and even when the license plates are broken or smeared, as efficiently and reliably as does human vision, is a distinct possibility in the near future.

ACKNOWLEDGMENT

The authors would like to thank Dr. A. Amiri for his enthusiastic guidance throughout this research work. They would also like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] A. H. Ashtari, "License plate localization and recognition in color images based on new feature of Iranian license plate and support vector machine," M.S. thesis, Dept. Eng., Islamic Azad Univ., Dezful, Iran, 2009.

[2] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008.

[3] W. Jia, H. Zhang, X. He, and M. Piccardi, "Mean shift for accurate license plate localization," in *Proc. IEEE Intell. Transp. Syst.*, 2005, pp. 566–571.

[4] S. L. Chang, L. S. Chen, Y. C. Chung, and S. W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

[5] X. Shi, W. Zhao, and Y. Shen, "Automatic license plate recognition system based on color image processing," in *Proc. ICCSA*, 2005, pp. 307–314.

[6] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," *Image Vis. Comput.*, vol. 27, no. 8, pp. 1134–1142, Jul. 2009.

[7] R. W. Rodieck, *The First Steps in Seeing*. Sunderland, MA, USA: Sinauer Associates, 1998.

[8] E. R. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing," in *Proc. IEEE ICIP*, 1994, pp. 301–305.

[9] C.-C. Lin and W.-H. Huang, "Locating license plate based on edge features of intensity and saturation subimages," in *Proc. 2nd ICICIC*, 2007, p. 227.

[10] F. Wang, L. Man, B. Wang, Y. Xiao, W. Pan, and X. Lu, "Fuzzy-based algorithm for color recognition of license plates," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 1007–1020, May 2008.

[11] Y. Yanamura, M. Goto, D. Nishiyama, M. Soga, H. Nakatani, and H. Saji, "Extraction and tracking of the license plate using Hough transform and voted block matching," in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 243–246.

[12] T. D. Duan, T. L. H. Du, T. V. Phuoc, and N. V. Hoang, "Building an automatic vehicle license plate recognition system," in *Proc. Int. Conf. Comput. Sci. RIVF*, 2005, pp. 59–63.

[13] N. Zimic, J. Ficzko, M. Mraz, and J. Virant, "The fuzzy logic approach to the car number plate locating problem," in *Proc. IIS*, 1997, pp. 227–230.

[14] S. M. Youssef and S. B. AbdelRahman, "RETRACTED: A smart access control using an efficient license plate location and recognition approach," *Exp. Syst. Appl.*, vol. 34, no. 1, pp. 256–265, Jan. 2008.

[15] H. Kwanicka and B. Wawrzyniak, "License plate localization and recognition in camera pictures," in *Proc. 3rd Symp. Methods Artif. Intell.*, 2002, pp. 243–246.

[16] H. Bai, J. Zhu, and C. Liu, "A fast license plate extraction method on complex background," in *Proc. IEEE Intell. Transp. Syst.*, 2003, pp. 985–987.

[17] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognit. Lett.*, vol. 26, no. 15, pp. 2431–2438, Nov. 2005.

[18] H. Zhao, C. Song, and S. Zhang, "License plate recognition system based on morphology and LS-SVM," in *Proc. IEEE GrC*, 2008, pp. 826–829.

[19] J. Jiao, Q. Ye, and Q. Huang, "A configurable method for multi-style license plate recognition," *Pattern Recognit.*, vol. 42, no. 3, pp. 358–369, Mar. 2009.

[20] S. H. Kasaei, S. M. Kasaei, and S. A. Kasaei, "New morphology based method for robust Iranian car plate detection and recognition," *Int. J. Comput. Theory Eng.*, vol. 2, no. 2, pp. 264–268, Apr. 2010.

[21] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M. R. Akbarzadeh-T, "Vehicle recognition based on Fourier, wavelet and curvelet transforms—A comparative study," in *Proc. ITNG*, 2007, pp. 939–940.

[22] F. M. Kazemi, S. Samadi, H. R. Poorreza, and M. R. Akbarzadeh-T, "Vehicle recognition using curvelet transform and SVM," in *Proc. ITNG*, 2007, pp. 516–521.

[23] V. Beëanoviéa, M. Kermitb, and A. J. Eidec, "Feature extraction from photographical images using a hybrid neural network," in *Proc. SPIE 9th Workshop Virtual Intell./Dyn. Neural Netw.*, 1999, vol. 3728, pp. 351–361.

[24] K. Jung, "Neural network-based text location in color images," *Pattern Recognit. Lett.*, vol. 22, no. 14, pp. 1503–1515, Dec. 2001.

[25] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2009.

[26] M. H. Dashtban, Z. Dashtban, and H. Bevrani, "A novel approach for vehicle license plate localization and recognition," *Int. J. Comput. Appl.*, vol. 26, no. 11, pp. 22–30, Jul. 2011.

[27] M. J. Ghasemi, H. R. Tajozzakerin, and A. R. Omidian, "An Iranian national number plate localization and recognition system for private vehicles," *Int. J. Acad. Res.*, vol. 2, no. 6, pp. 13–19, Nov. 2010.

[28] A. Broumandnia and M. Fathy, "Application of pattern recognition for Farsi license plate recognition," *ICGST Int. J. Graph., Vis. Image Process.*, vol. 5, no. 2, pp. 25–31, Jan. 2005.

[29] M. Rasooli, T. Branch, S. Ghofrani, and E. Fatemizadeh, "Farsi license plate detection based on element analysis and characters recognition," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 4, no. 1, pp. 65–80, Mar. 2011.

[30] F. PirahanSiah, S. N. H. S. Abdullah, and S. Sahran, "Adaptive image segmentation based on peak signal-to-noise ratio for a license plate recognition system," in *Proc. ICCAIE*, 2010, pp. 468–472.

[31] W. Burger and M. J. Burge, *Digital Image Processing: An Algorithmic Introduction Using Java*. New York, NY, USA: Springer-Verlag, 2007.

[32] M. Sezgin, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imaging*, vol. 13, no. 1, pp. 146–168, Jan. 2004.

[33] K. Jung, K. In Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognit.*, vol. 37, no. 5, pp. 977–997, May 2004.

[34] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognit.*, vol. 19, no. 1, pp. 41–47, 1986.

[35] Y. Zhang and C. Zhang, "A new algorithm for character segmentation of license plate," in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 106–109.

[36] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognit.*, vol. 38, no. 11, pp. 1961–1975, Nov. 2005.

[37] C. A. Rahman, W. Badawy, and A. Radmanesh, "A real time vehicle's license plate recognition system," in *Proc. IEEE Conf. Adv. Video Signal Based Surveillance*, 2003, pp. 163–166.

[38] Y. P. Huang, S. Y. Lai, and W. P. Chuang, "A template-based model for license plate recognition," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, 2004, pp. 737–742.

[39] A. Delforouzi and M. Pooyan, "Efficient Farsi license plate recognition," in *Proc. 7th ICICS*, 2009, pp. 1–5.

[40] M. M. Dehshibi and R. Allahverdi, "Persian vehicle license plate recognition using multiclass Adaboost," *Int. J. Comput. Elect. Eng.*, vol. 4, no. 3, pp. 355–358, Jun. 2012.

[41] H. Mahini, S. Kasaei, F. Dorri, and F. Dorri, "An efficient features-based license plate localization method," in *Proc. 18th ICPR*, 2006, pp. 841–844.

Amir Hossein Ashtari (S'12) received the B.S. and M.S. degrees in computer engineering, hardware and architecture of computer systems from Islamic Azad University, Tehran, Iran, in 2006 and 2009, respectively. He is currently working toward the Ph.D. degree in the Centre for Artificial Intelligence Technology, Pattern Recognition Research Group, Department of Computer Science, National University of Malaysia (UKM), Bangi, Selangor, Malaysia, under the supervision of Associate Professor Dr. Md. Jan Nordin.

His research interests include traffic controlling, license plate recognition/detection, fragment objects analysis, shape descriptors, and transforms.

Md. Jan Nordin received the B.S. and M.S. degrees in computer science from Ohio University, Athens, OH, USA, in 1982 and 1985, respectively, and the Ph.D. degree in engineering information technology from Sheffield Hallam University, South Yorkshire, U.K., in 1995.

He is currently an Associate Professor with the Centre for Artificial Intelligence Technology, National University of Malaysia (UKM), Bangi, Selangor, Malaysia. His current research interests include pattern recognition, computer vision, intelligent system, and image reconstruction.

Mahmood Fathy received the Ph.D. degree in image processing and processor design from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1991.

He is currently an Associate Professor with the Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran, where he has been also the Head of the Security Information Technology Group since 2009. He has coauthored five books, over 70 scientific papers for different journals, and 300 conference papers. His research interests include computer networks, network on chip, computer architecture, distributed image and video processing, and intelligent transportation systems.