An Algorithm for License Plate Recognition Applied to Intelligent Transportation System

Ying Wen, Yue Lu, *Member, IEEE*, Jingqi Yan, Zhenyu Zhou,
Karen M. von Deneen, and Pengfei Shi, *Senior Member, IEEE*

***Abstract*—An algorithm for license plate recognition (LPR) applied to the intelligent transportation system is proposed on the basis of a novel shadow removal technique and character recognition algorithms. This paper has two major contributions. One contribution is a new binary method, i.e., the shadow removal method, which is based on the improved Bernsen algorithm combined with the Gaussian filter. Our second contribution is a character recognition algorithm known as support vector machine (SVM) integration. In SVM integration, character features are extracted from the elastic mesh, and the entire address character string is taken as the object of study, as opposed to a single character. This paper also presents improved techniques for image tilt correction and image gray enhancement. Our algorithm is robust to the variance of illumination, view angle, position, size, and color of the license plates when working in a complex environment. The algorithm was tested with 9026 images, such as natural-scene vehicle images using different backgrounds and ambient illumination particularly for low-resolution images. The license plates were properly located and segmented as 97.16% and 98.34%, respectively. The optical character recognition system is the SVM integration with different character features, whose performance for numerals, Kana, and address recognition reached 99.5%, 98.6%, and 97.8%, respectively. Combining the preceding tests, the overall performance of success for the license plate achieves 93.54% when the system is used for LPR in various complex conditions.**

***Index Terms*—Bernsen algorithm, character recognition, feature extraction, license plate recognition (LPR), support vector machine (SVM).**

I. Introduction

LICENSE PLATE RECOGNITION (LPR) plays an important role in numerous applications such as unattended parking lots [1], [3], [4], security control of restricted areas [5], [6], and traffic safety enforcement [8], [9]. This task is quite challenging due to the diversity of plate formats and the nonuniform outdoor illumination conditions during image acquisition, such as backgrounds [2], [10], illumination [7], [15], [22], vehicle speeds [11], and distance ranges between the camera and the vehicle [19]. Therefore, most approaches work only under restricted conditions such as fixed illumination, limited vehicle speed, designated routes, and stationary backgrounds.

A typical system for LPR consists of four parts, i.e., obtaining an image of the vehicle, license plate localization and segmentation, character segmentation and standardization, and character recognition. The performance of the locating operation is crucial for the entire system, because it directly influences the accuracy and efficiency of the subsequent steps. However, it is also a difficult obstacle to overcome because of different illumination conditions and various complex backgrounds. Researchers have proposed many methods of locating the license plates, such as the edge detection method [12], [13], [28], [30], line sensitive filters to extract the plate areas [31], the window method [1], [2], [25], [32], and the mathematics morphology method [33]. Although these algorithms can process the license plate's location, they possess formidable disadvantages such as sensitivity to brightness, longer processing time, and lack of versatility in adapting to the varying environment.

Character segmentation has, in the past, been accomplished by such techniques as projection [10], [35], morphology [41]–[43], relaxation labeling, and connected components [14], [44]. There have been a large number of character recognition techniques reported such as model match [29], Bayes' classifier [45], [46], artificial neural networks [47], [48], fuzzy c-means [44], support vector machine [49], Markov processes [50], and K-nearest neighbor classification [51]–[53]. Although these algorithms can process the license plate segmentation and recognition, most of them only process a single-line character segmentation and only two types of character recognition: English and numeric characters. More complex plate construction methods and more types of character recognition were not discussed.

For LPR research, people may be dedicated to one aspect of LPR, such as research on license plate localization and segmentation [2], [16], [18], [21], [25] and LPR [4], [7], [19], [20], [29]. However, for a complete and successful license plate system, a meticulous and comprehensive dissection of the individual parts and how they work together is necessary. Table I highlights studies concerning various LPR systems presented in recent literature. This literature reported plate location rate, character recognition rate, and overall performance.

Manuscript received January 19, 2010; revised July 24, 2010 and December 20, 2010; accepted January 30, 2011. Date of publication March 3, 2011; date of current version September 6, 2011. This work was supported in part by the 973 Program under Grant 2011CB3o22oo and in part by the National Natural Science Foundation of China under Grant 60775009. The Associate Editor for this paper was Q. Ji.

Y. Wen and Z. Zhou are with Columbia University, New York, NY 10032 USA (e-mail: yw2365@columbia.edu; zz2177@columbia.edu).

Y. Lu is with the Department of Computer Science and Technology, East China Normal University, Shanghai 200062, China (e-mail: ylu@cs.ecnu.edu.cn).

J. Yan and P. Shi are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200030, China (e-mail: jqyan@sjtu.edu.cn; pfshi@sjtu.edu.cn).

K. M. von Deneen is with the School of Life Sciences and Technology, Xidian University, Xi'an 710126, China (e-mail: vondenk@ufl.edu).

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.2011.2114346

TABLE I
OVERALL PERFORMANCE OF THE LICENSE PLATE RECOGNITION SYSTEM IN THE LITERATURE AND OUR SYSTEM

Ref.	Rate (%) pd:plate detection rate Cr:character recognition rate Ov:overall rate	Character types	Platform /Processor	Speed (s)	Scientific background P:license plate detection method R:character recognition method
Ref. [1]	96.7 (Pd) 97.1 (Cr) 93.9 (Ov)	English numeral	CPU 3.0G	0.293	P:search window location R:feed forward ANN
Ref. [12]	95.9 (Pd) 92.3 (Cr) 90 (Ov)	Chinese numeral English	CPU 3.0G	0.125	P:edge analysis R:feed forward NN
Ref. [14]	91.7 (Pd) 90.93 (Ov)	English numeral	Xilinx Virtue IV FPGA	no report	P:Gabor filter connected component
Ref. [17]	91 (Ov)	English numeral	IBM RS	1.1	P:filtering, enhancement R:template matching
Ref. [22]	96.5 (Pd) 89.1 (Cr) 86 (Ov)	English numeral	VC++ 6.0	0.276	P:sliding concentric windows R:probabilistic NN
Ref. [23]	97.9 (Pd) 95.6 (Cr) 93.7 (Ov)	English numeral	no report	no report	P:fuzzy disciplines R:neural network
Ref. [35]	87 (Ov)	English numeral	no report	no report	P:projection R:Enclidean distance
our system	**97.16 (Pd)** **97.88 (Cr)** **93.54 (Ov)**	**English, Chinese numeral, Kana**	CPU 1.8G VC++ 6.0	0.284	P:improved Bernsen algorithm connected component R:support vector machine

In the last column in Table I, the scientific background for plate identification and LPR is indicated. Character style is also considered due to the different characters on the license plate from different countries. The systems presented in [1], [12], [14], [17], [22], [23], and [25] report their overall performances as (90.1%, 93.9%, 90.93%, 91%, 86%, 93.7%, and 87%), respectively. However, the systems were only capable of processing a single line of large characters, even if there were two lines of characters on the license plate [12], [14], [17], [19], [22]–[24], [29], [31], [35]. The highest performing LPR system currently available reported a success rate of 93.7%, yet the system could only process numeric and English characters [35].

In this paper, our work focuses on a solution for image disturbance resulting from uneven illumination and various outdoor conditions such as shadow and exposure, which are generally difficult for obtaining successful processed results using traditional binary methods. Additionally, we discussed the feature extraction methods for Chinese, Kana, English, and numeric characters and adopted a support vector machine (SVM) for classification. Moreover, we reported the system's whole performance and each individual model performance. Table I shows that the final overall performance of our system achieved 93.54%, and the processed characters consisted of numerals, English, Chinese, and Kana. The novel contributions of this paper are given as follows: 1) a novel binary method, i.e., the shadow removal method, which is based on an improved Bernsen algorithm combined with the Gaussian filter; and 2) a character recognition algorithm, which is the SVM integration, where the character features are extracted from the elastic mesh and the entire address character string is taken as the object of study, as opposed to a single character. In addition to the preceding contributions, feature extraction methods for Chinese characters are also examined to choose some effective features to meet the practical application demands. An image slant correction trick and image enhancement trick are also presented. The rest of this paper is organized as follows: In Section II, license plate preprocessing is presented, where an improved Bersen algorithm and image tilt correction trick are proposed. Character recognition algorithms are discussed in detail in Section III. Experimental results are presented in Section IV, and finally, a discussion and future work are presented in Section V.

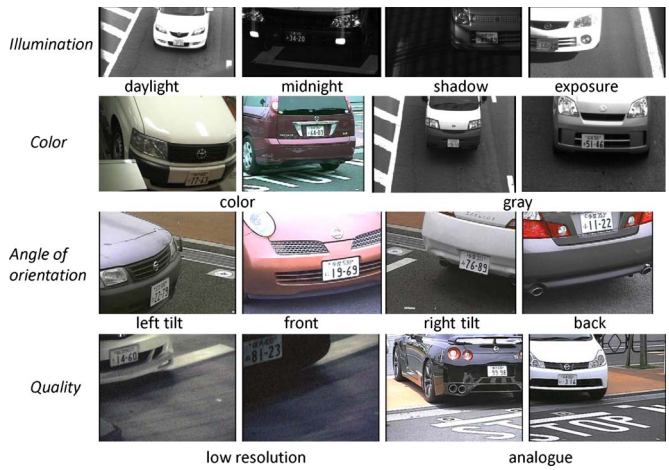


Fig. 1. License plate images.

II. LICENSE PLATE PREPROCESSING

License plate preprocessing is a necessary step in LPR, which includes plate detection, correction, and segmentation. The goal of detection is to locate regions of interest that are similar to the license plate. Due to the angle of orientation, the image may have a slant and distortion; thus, transformation or

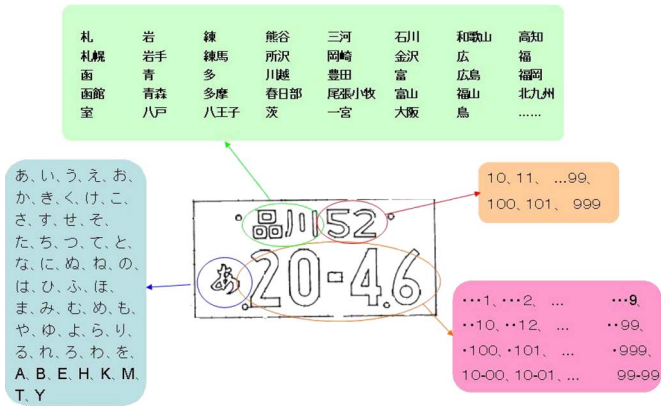


Fig. 2. Japanese license plate sample.

Fig. 3. License plate images from different countries.

Fig. 4. Shadow image and traditional binary results. (a) Image with a shadow. (b) Result by the local Otsu. (c) Result by the global Otsu. (d) Result by the differential binary.

correction of image is an important step before the character segmentation.

A. Japanese License Plate Description

In this paper, a Japanese license plate is the object of study, and some samples are presented in Fig. 1, followed by a brief description of the proposed Japanese LPR process. The images in the first row are captured by different illuminations, which include daylight, night, shadow, and exposure conditions. The second-row images indicate that the license plate images may be colored or gray. The third-row images show the images captured from various angles of orientation. The image resolution may be low in the last row images due to the filming equipment. Two types of the license plates should be considered, i.e., white characters/black background and black characters/white background.

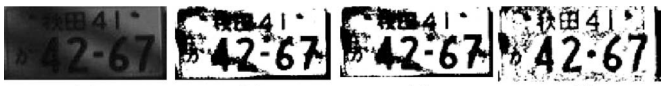
Fig. 2 shows a sampling of the typical Japanese license plate. The Japanese license plate consists of Kana, Chinese, English, and numeric characters. The distribution of the positions of the characters is presented. In the first row, a Chinese address character string, which consists of one to four Chinese characters out of 158 possible characters is circled in green. Next to the Chinese characters, circled in orange, is a small numeric string, which typically consists of two or three numerals. In the second row, circled in blue, is either a Kana or an English character. There are 43 possible Kana characters and eight possible English characters. Next to the Kana or English characters, circled in red, are large numbers (typically one to four numbers). The Japanese license plate has two features: 1) The license plate consists of four types of characters (Chinese, English, Kana, and numeric characters) to increase the difficulty of character recognition. 2) There are two lines of characters in the frame, and each line has two types of characters to increase the difficulty of character segmentation.

The main differences between our studied plate with the license plates of other countries (shown in Fig. 3) are the following: 1) Most license plates of other countries may include two or three types of characters, such as English and numeric characters on the American license plate and Chinese, English, and numeral characters on the Chinese license plate. 2) Most license plates have only a single line of characters such as on the plates from Germany, U.K., and other European nations. Although there may be two lines of characters on the license plates from different nations, most existing LPR systems only process the second line of characters, which is larger than the first line of characters. Since the first line of characters is not processed, the information obtained by the existing LPR may not satisfy the practical application demand [1], [14], [17]–[19], [22], [23], [31], [35]. Hence, the practical application value of our LPR is superior to that of other existing LPR methods. In the succeeding sections, we present license plate detection and license character recognition.

B. License Plate Location

The brightness distribution of various positions on a license plate image may vary due to the condition of the plate and the effect of the lighting environment. Since a binary method with a global threshold cannot always generate satisfactory results in such cases, the adaptive local binary method is often used. The local binary method means that an image is divided into $m \times n$ blocks, and then, each block is processed with the binary method. In our research, two local binary methods were adopted, i.e., the local Otsu and an improved Bernsen algorithm. Otsu [36] is a traditional binary method, which we used on each subblock. However, the performance of Otsu is contingent on the illumination conditions, which greatly vary. To resolve the uneven illumination obstacle, particularly for shadow images, we proposed a novel binary method, i.e., the improved Bernsen algorithm.

1) Improved Bernsen Algorithm: Since the processed license plates were obtained under various illumination scenarios and complex backgrounds, shadows or uneven illumination were unavoidable on the license plate. Hence, shadow or uneven illumination removal becomes a necessary step in our system. In this paper, an improved Bernsen algorithm is proposed for uneven illumination, particularly for shadow removal. Fig. 4(a) shows a license plate image with a shadow. Fig. 4(b)–(d) shows the results of the local Otsu, global Otsu, and differential local threshold binary methods, respectively. From these binary results, we can see that these traditional binary approaches cannot effectively remove the shadow, and the



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

Fig. 5. Results of a shadow image with different window sizes of the Bernsen algorithm. (a) Plate. (b) $k = 3, l = 3$. (c) $k = 10, l = 10$. (d) $k = 0, l = 10$

license plate cannot be successfully detected and segmented. To solve this problem, a new binary approach was proposed, i.e., the improved Bernsen algorithm.

a) Bernsen algorithm: In order for the target and background to be separated with a histogram showing a bimodal pattern of images, the global threshold binary methods could obtain good results such as with Otsu and the mean gray value. However, in the practical environment, due to image noise and other reasons, the image histogram does not present a bimodal pattern. In this case, traditional binary methods cannot obtain satisfactory results. Local threshold methods are commonly used in the identification of more serious interference or uneven illumination of the image, such as the Bernsen algorithm [37] and Niblack algorithm [38]. In general, with the excellent performance of the local binary algorithm, the Bernsen algorithm is an appropriate solution to the problem of uneven illumination.

Suppose that $f(x, y)$ denotes a gray value of point (x, y). Consider a block whose center is a point (x, y) and size is $(2w + 1) \times (2w + 1)$. The threshold $T(x, y)$ of $f(x, y)$ is computed by

$$T_1(x,y) = \frac{\max\limits_{-w\le k,l\le w} f(x+l,y+k) + \min\limits_{-w\le k,l\le w} f(x+l,y+k)}{2}. \tag{1}$$

Then, the binary image is obtained by

$$b(x,y) = \begin{cases} 0, & \text{if } f(x,y) < T_1(x,y) \\ 255, & \text{else.} \end{cases} \tag{2}$$

Fig. 5(a) shows a shadow image, and Fig. 5(b)–(d) shows the results of the Bernsen algorithm using different window sizes. From this figure, we can see that the Bernsen algorithm is sensitive to noise, which disturbs the characters' extraction. Removal of noise and preservation of the characters are of importance in this study.

b) Improved Bernsen algorithm: Suppose that $\overset{\Lambda}{f}(x, y)$ denotes the gray value obtained with the Gaussian filter, σ is the scale of the Gaussian filter, and k and l are the parameters of the window. An improved Bernsen algorithm is depicted here.

Step 1) Compute the threshold $T_1(x, y)$ of $f(x, y)$ based on (1).

Step 2) Create the Gaussian filter for the window $s = (2w + 1) \times (2w + 1)$ of $f(x, y)$, i.e.,

$$\overset{\Lambda}{f}(x,y) = \frac{1}{(2w+1)^2} \sum_{x,y\in S} f(x,y) \times \exp\left\{-\frac{1}{2}\left[\left(\frac{x}{\sigma}\right)^2 + \left(\frac{y}{\sigma}\right)^2\right]\right\}. \tag{3}$$

Step 3) Compute the threshold $T_2(x, y)$ of $\overset{\Lambda}{f}(x, y)$ as

$$T_2(x,y) = \frac{\max\limits_{-w\le k,l\le w} \overset{\Lambda}{f}(x+l,y+k) + \min\limits_{-w\le k,l\le w} \overset{\Lambda}{f}(x+l,y+k)}{2}. \tag{4}$$

Step 4) Obtain a binary image by

$$b(x,y) = \begin{cases} 0, & \text{if } f(x,y) < \beta\left((1-\alpha)T_1(x,y) + \alpha T_2(x,y)\right); \beta \in (0,1) \\ 255, & \text{else} \end{cases} \tag{5}$$

where α is a parameter to adjust the balance between the Bernsen algorithm with the Gaussian filter and the traditional Bernsen algorithm ($\alpha \in [0, 1]$). When α is equal to 0, the proposed algorithm is the Bernsen algorithm. When α is equal to 1, the proposed algorithm is the Bernsen algorithm with a Gaussian filter. By adopting an appropriate α, the shadow can be effectively removed, and the characters can be successfully identified.

Step 5) Apply median filter to remove the noise.

In this algorithm, α and w have significant impacts on the processing results. The parameters are analyzed in detail as given here.

1) w: The traditional Bernsen algorithm adopts the min-max value of a local window as the inspection threshold. k and l are the parameters of w in (1).
   a) k: k and l are the main parameters for the Bernsen algorithm time consumption. Suppose that k is the y-axis direction size of w and that l is the x-axis direction size of w. Some processed images are presented in Fig. 5 with different values for k and l. Fig. 5(a) shows a shadow image. Fig. 5(b)–(d) shows the results with different k's and l's. When the values of k and l are large, the computation of the Bernsen algorithm is very time consuming, and the effectiveness of shadow removal is not clear [see Fig. 5(c)]. When $k = 0$ or $l = 0$, shadow removal is evident (see Fig. 5(d), particularly the shadow on the characters), and the computation cost is less. Selecting $k = 0$ or $l = 0$, this results in a different scan direction, and there is no influence on the binary result. In our study, we adopted the horizontal scan, i.e., $k = 0$ to implement the binary algorithm.
   b) l: When $k = 0$, the window is only determined by l. We found that the size of l influences the processed result of the characters on the license plate through experimentation. If l is small, the character may be removed, but noise may be produced. In our experiment, we found the value of l is often defined by a value within the width of the characters to obtain good results. For example, in Fig. 6, the maximum width of a character "5" is 14 pixels, and the minimum width is 4 pixels; thus, the mean width (nine pixels) is chosen as the value of l. Fig. 6 shows the processed results of the shadow image with different l values. We can

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

Fig. 6. Processed results of a shadow image with different l values of the improved Bernsen algorithm. (a) Plate. (b) $l = 4$. (c) $l = 9$. (d) $l = 20$.

Value	10	205	207	205	225	215	205	10	...
Bernsen result	0	0	0	255	225	255	...		

Fig. 7. Example of the Bernsen algorithm.

(a) (b) (c)

Fig. 8. Effect of β for the Bernsen algorithm. (a) Plate. (b) $\beta = 1$. (c) $\beta = 0.9$.

see that choosing the mean width of a character as the inspection l can obtain good results.

2) β: The Bernsen algorithm is sensitive to the difference of the current pixel value and the threshold. Fig. 7 shows an example of the sensitivity of the Bernsen algorithm (suppose $l = 5$). The red color "0" is the error result. To resolve this problem, the threshold can be multiplied by a constant β to reduce the Bernsen algorithm's sensitivity. Fig. 8 presents the effect of β of the Bernsen algorithm. Based on the experimental experience, the value of β was defined as 0.9 to obtain optimal results.

3) α: When T1 (1) is smaller than the current black pixel, the value of the pixel is set to 255 according to (2). When an image is processed with the Gaussian filter, T2 [in (4)] is bigger than T1 in the black pixel area, particularly in the characters' area [see Fig. 9(b)]. Hence, $T = \beta(\alpha T2 + (1-\alpha)T1) \leq T1$; therefore, the horizontal part of the character and plate frame are preserved by this fine adjustment. The Gaussian filter does not blur the edges of the characters, because it only processes the original image to gain T2. The final binary result is obtained by (4). Hence, the Gaussian filter improves the performance of the Bernsen algorithm in the proposed algorithm. Fig. 9(c) and (d) shows a comparison of the traditional Bernsen algorithm and the proposed algorithm. We can see that some characters and the plate frame achieve good binary results. The figure shows that the proposed algorithm is superior to the traditional Bernsen algorithm. In our experiments, the scale σ of the Gaussian filter is set to 10 to achieve optimal results, and α is set to 0.5.

Fig. 10 shows some examples of uneven illumination images processed by the improved Bernsen algorithm. This method can obtain a good binary result, and it is noted that the strokes of the Kana characters are also retained, despite fewer pixels.

2) License Plate Detection: After the license plate is processed with the binary method, the system steps into license plate detection. The result of the detection is the key to the following work and directly affects the performance of the entire system.

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

Fig. 9. Comparison of the traditional Bernsen algorithm and the proposed algorithm. (a) License plate with a shadow. (b) Gaussian filter result on the plate. (c) Bernsen algorithm result. (d) Proposed algorithm result.

Fig. 10. Some examples processed by the improved Bernsen algorithm under uneven illumination.

Connected component analysis (CCA) [40] is a well-known technique in image processing that scans an image and labels its pixels into components based on pixel connectivity (i.e., all pixels in a connected component share similar pixel intensity values) and are, in some way, connected with each other (either four-connected or eight-connected). Once all groups have been determined, each pixel is labeled with a value according to the component to which it was assigned. CCA works on binary or gray-level images, and different measures of connectivity are possible. For the present application, we applied CCA to binary images searching in an eight-connectivity situation.

Based on the Japanese license plate information, two types of Japanese license plates are presented, i.e., black characters on a white background and white characters on a black background (shown in Fig. 12). Two detection methods were employed: one was the detection of a white frame with CCA, and another was the detection of black characters with CCA. If the type of the license plate is unknown, the license plate detection method follows two procedures.

Method 1: License plate location with a frame. Candidate frames are detected based on prior knowledge of the license plate. The yellow frames are the detected results shown in Fig. 12(a). Using method 1, some candidate regions can be detected. The frame is detected by the connected component technique, which is sensitive to the edge of the frame. If the frame is broken, the license plate cannot be correctly detected. Method 1 can be easily calculated, and one must first judge whether the detected area coincides with the plate size.

Method 2: License plate location without a frame. When the frame of the license plate cannot be detected, method 2 is

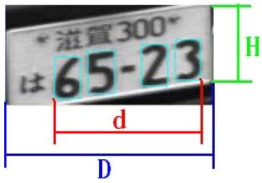


Fig. 11. Computation method of the license plate size using the connected components.

Fig. 12. License plate location methods.

adopted. This method detects the plate by large numeral extraction. If the license plate with white characters/black background is not detected, its inverse image should be detected.

Fig. 12(b) shows that there is no license plate detected with method 1. First, to attain the connected components of similar size, some parameters are adopted based on the characters' prior information such as the pixel amount of a connected component, width that is greater than 10, height that is greater than 20, ratio of height to width that is less than 2.5 or greater than 1.5, and so on. In this way, all retained connected components are similar in size. Second, to remove some noncharacter connected components, other constraints are also provided by character position information on the license plate such as the distance between two adjacent characters or average character tilt angle. In this way, discrete connected components are removed, and the connected components may be the characters that are preserved. In Fig. 11, four blue frames corresponding to the character-connected components are detected based on prior knowledge. For a standard license plate (see Fig. 2), the ratio θ of the distance between the first large numeral and the last large numeral and the license plate width is known. The ratio is constant, despite an alteration in the size or scale of the license plate. First, we computed the distance d of the first numeral and the last numeral. Then, we obtained the plate width $D = d/\theta$. According to the same scheme, we also obtained the plate height. Considering the image tilt problem, we increased the values of the width and height to define the plate position. If three large numerals were detected, we also adopted the aforementioned scheme to acquire the license plate position using a different ratio of the three characters' width and the plate width. A blue frame in Fig. 12(b) is a detected result using method 2.

In general, method 1 consumes less time than method 2. Method 2 is effective only if there are no results from using method 1. Compared with method 1, the location rate of method 2 is low.

Whether adopting method 1 or method 2, candidate frames may present more than one frame at a time [there are two yellow frames in Fig. 12(a)]. For the candidate frames obtained by method 1, we employed CCA to extract large numerals and compute the penetration times of the midline of the plate to filter some candidate frames. The number of penetration times denotes the number change from black pixels to white pixels along the midline. Fortunately, although the method is simple, nonlicense plate frames are efficiently filtered. However, this method does not fit the candidate frames obtained by method 2. For the remaining frames, this step cannot provide the final judgment. We sent these candidate frames through the following steps and discriminated the true license plate based on the recognition results.

C. Character Segmentation

Real license plate images are prone to slant and distortion due to different angles of orientation. Therefore, horizontal and vertical correction and image enhancement are required prior to character segmentation.

1) Horizontal Correction: In this paper, the size of the image varies. For example, the size of a small image is about 40×80 pixels, whereas a large one is about 200×400 pixels. First, all detected license plates needed to be resized to 100×200 pixels. We then used the connected component technique to detect large numerals and find the center position of each large numeral. Next, we computed the tilt angle of every two central points, and the average tilt angle was obtained. Finally, we adopted a 2-D rotation method to correct the image according to the average tilt angle. Fig. 13(a) shows the horizontal correction method. The left column images are the detected license plates. The middle column images are the resized images. The blue frames are the large numerals detected by CCA, and a red line across the centers of the numerals is the horizontal tilt angle of the image. According to the tilt angle, the license plates can be corrected [shown in the right column in Fig. 13(a)].

2) Vertical Correction: Although an image is processed by horizontal adjustment, the image may have a vertical slant [see the first image of Fig. 13(b)]. Due to the vertical slant, small numerals present an adhesion status, and it is difficult to divide the small numerals with CCA or other projection techniques. In this case, vertical correction is important for small numeral segmentation.

To find the vertical slant angle, we proposed a method of finding the projection minimum. After the horizontal correction, the area of the large numerals shown in the second image of Fig. 13(b) is extracted to locate the projection minimum. First, the width of the horizontal projection is recorded. The black projective length shown in the second image of Fig. 13(b) is the width of the horizontal projection. When the image is rotated n degrees ($n \leq 5$ in our study), the projection value and rotation angle are recorded. The procedure is repeated until the image is rotated from -45° to 45°. (The maximum slant angle is 45° in our license plates.) Next, we found the projection minimum in all of the projection data, which corresponded to the rotation angle or the vertical slant angle. The second and third images of Fig. 13(b) show the process of finding the projection minimum. Finally, the image could be corrected based on the vertical tilt angle. We could not adopt the 2-D rotation technique to correct

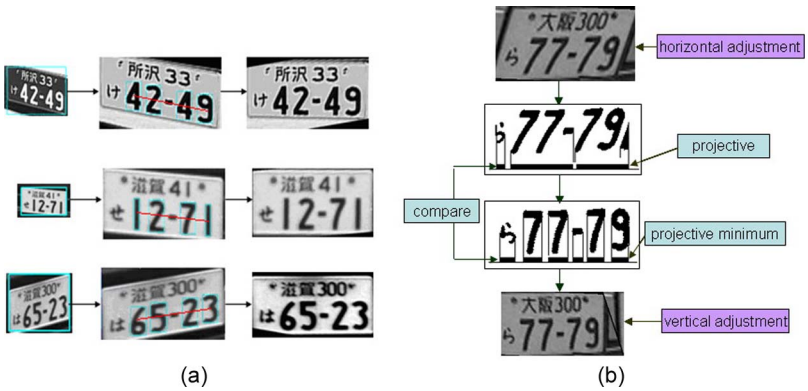


Fig. 13. Image tilt correction. (a) Horizontal correction. (b) Vertical correction.

the image in this step, because the image may have a horizontal slant after rotation. In this case, we adopted the pixel horizontal movement method to correct the image. Since the projection computation determined the center of the image as the origin of the 2-D rotation, each pixel movement distance could be computed by

$$d_x = \left(y - \frac{1}{2}H\right) / \tan(\zeta) \tag{6}$$

where d_x is the horizontal distance of each pixel movement, x and y are the coordinates of the current pixel, H is the height of the image, and ζ is the vertical slant angle. The last image in Fig. 13(b) is the result of vertical correction; hence, small numerals can be successfully extracted.

3) Image Gray Equilibrium: For seeing a difference in image quality, the image gray equilibrium is a necessary step in the segmentation stage. However, for ambient illumination, it is difficult to obtain a satisfactory result using traditional methods such as the histogram equilibrium and gray stretch. In this paper, a gray statistical approach (GSA) was used for image enhancement. In the statistical approach, the expectation σ_s and mean E_s of an image are defined as

$$\sigma_s = \frac{1}{n}\sum (f - E_s)^2 \tag{7}$$

$$E_s = \frac{1}{n}\sum f. \tag{8}$$

The image is revised by the statistical approach [39]

$$g = \frac{\sigma_d}{\sigma_s}(f - E_s) + E_d \tag{9}$$

where σ_d and E_d are the targets of expectation and mean, respectively.

However, there are some shortcomings of GSA: 1) heavy computation and 2) difficulty to define σ_d if σ_s is unknown. We found that the predefined ratio κ of σ_d and σ_s could solve the problem. Hence, the computation equation (9) could be revised as

$$g = \kappa(f - E_s) + E_d. \tag{10}$$

The ratio κ and the target mean E_d were obtained via numerous experimental tests, and in our study, they were defined as 1.92 and 100, respectively. Fig. 14 shows the image gray equilibrium results processed with different methods. The first row of images is the original license plates. The second, third, and fourth rows of images are the images processed by the histogram equilibrium, gray stretch, and the proposed method, respectively. From these figures, we can see that traditional methods cannot fit all of the images, and the results obtained by GSA are superior to the results of others.

Fig. 14. Image gray equilibrium results of the different methods.

Fig. 15 shows a flowchart indicating the character segmentation of the license plate. When a plate was located, we already knew that the plate either had white characters/black background or black characters/white background. Before the character segmentation, the plate was transformed to black characters/white background. Then, all of the plates were resized to 100 × 200. Afterward, the tilt correction and image enhancement were implemented. Next, we used the projective technique to segment the license plate into two blocks, and some characters were extracted from each block. Finally, the characters were resized to a uniform size. Fig. 16 presents the examples of license plate segmentation.

III. Character Recognition Algorithms

In the system, all characters on the Japanese license plate were recognized. There are four character types on each plate, i.e., Chinese characters, Kana, English characters, and numerals.

In this section, we present some feature extraction methods for character recognition. In general, a feature fits this type of character recognition, whereas it does not necessarily fit other

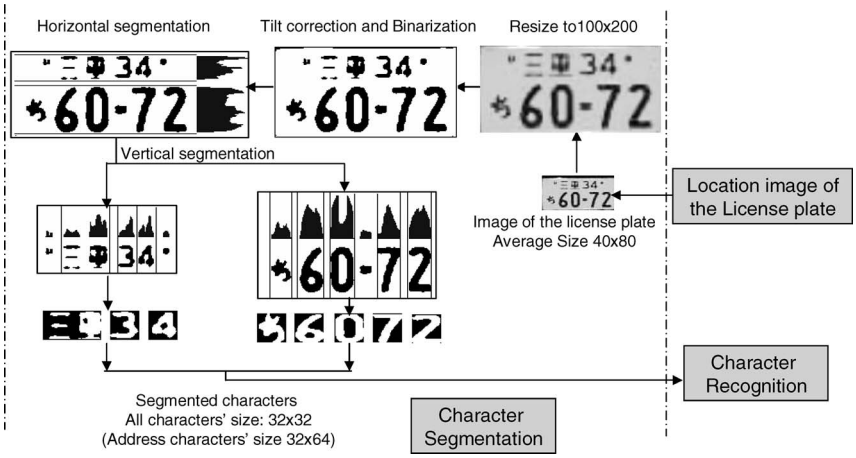


Fig. 15. Flowchart indicating the character segmentation of the license plate.

Fig. 16. Successful examples of the license plate segmentation.

characters. To test the effects of the feature extraction method, certain experiments were implemented. The recognition results of some suitable feature extraction methods were integrated to get a satisfactory recognition rate.

A. Feature Extraction for Character Recognition

1) Feature Extraction for Chinese Characters: On the Japanese license plate, there is a group of Chinese characters consisting of one, two, three, or four Chinese characters, which denote the address (see Fig. 2). In our study, this group of Chinese characters is named the address character string. We took the address character string as a recognized object rather than a single Chinese character. The goal was to reduce the difficulty of character segmentation and postprocessing. Before the address character string recognition, the entire address character image was normalized to 32×64 pixels.

For Chinese characters, Akiyama and Nagita [27] proposed a statistical method based on the stroke direction features, i.e., the global direction contributivity density (G-DCD), the local direction contributivity density (L-DCD), and the peripheral direction contributivity (PDC).

Here, we described the feature extraction approaches of G-DCD and L-DCD. First, we defined the eight stroke directions L_i $(i = 1, 2, \ldots, 8$, i.e., 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°), which indicates eight distances between the pixel on a stroke and eight directional edges of the stroke (see Fig. 17). Normalization of d_i was obtained by

$$d_i = \frac{(L_i + L_{i+4})}{\sqrt{\sum_{i=1}^{4}(L_i + L_{i+4})}}, \quad i = 1, 2, 3, 4. \tag{11}$$

Based on the definition of stroke direction, the direction feature of a point on a stroke is a quaternion $D = (d_1, d_2, d_3, d_4)$.

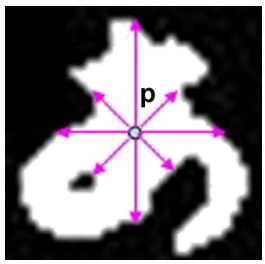


Fig. 17. Direction feature of the stroke.

The stroke direction feature reflects the stroke direction and the connected relationship of the strokes. Based on the idea of the stroke direction feature, two types of direction features were proposed by Akiyama [27].

1) G-DCD. This is a 1-D feature vector reflecting the complexity, direction, and connected relationship of character strokes. Suppose that the size of a character image is $m \times n$. First, scanning the image from left to right, all crosspoints (a, b, c, d) of the scanning line and stroke contour are obtained. The direction features of d_3 of all of the crosspoints are then calculated and added. At the horizontal direction $(T = 0)$, we get m direction features, and m is divided into r parts. The feature values of each part are added, so that the r horizontal direction feature values are also obtained. According to a similar operation, other direction stroke features (45°, 90°, and 135°) can be obtained. Fig. 18(a) shows the sketch of the G-DCD feature extraction. $T = 0$, 1, 2, 3 denotes a 0°, 90°, 45°, and 135° direction feature extraction, respectively.
2) L-DCD. Similar to G-DCD, L-DCD is computed from all subimages and is then divided into the original image [shown in Fig. 18(b)]. The character image is segmented into $p \times q$ blocks, and the direction features for each block are added to construct L-DCD. In addition to d_3, d_1 is also computed. Hence, the dimension of L-DCD is $2 \times p \times q$. This feature reflects the local structure of the character.
3) Contour feature. When the first character pixel is detected with the horizontal scan, the x-axis distance of the pixel is recorded. The character pixel here means the pixel from

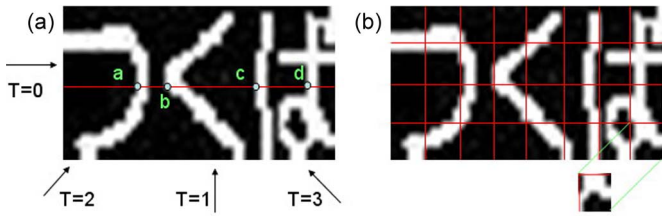


Fig. 18. Sketches of G-DCD and L-DCD feature extraction. (a) G-DCD. (b) L-DCD.

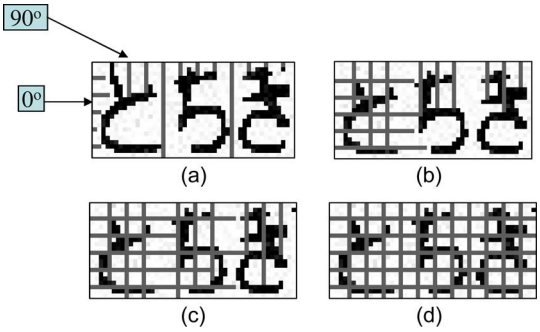


Fig. 19. Contour features and penetrated feature. (a) One order contour. (b) Two order contour. (c). Three order contour. (d). Penetrated density feature.

the background to the character. The distance is known as the one-order contour feature. Similarly, the two-order contour feature and three-order contour feature denote the x-axis distance of the second pixel and the third pixel, respectively. Contour features also include four directions or angles, i.e., 0°, 45°, 90°, and 135°. Fig. 19(a)–(c) shows the sketches of one-, two-, and three-order contour features, respectively.

4) Penetrated feature. When scanning an image in the horizontal direction, the number of times that a white pixel turns to a black pixel is recorded. We named this feature the penetrated feature. There are also four direction features, i.e., 0°, 45°, 90°, and 135°. Fig. 19(d) shows the sketch of the penetrated feature.

2) Feature Extraction for Kana and English Characters: The presented positions of Kana and English characters are the same on a Japanese license plate. In our study, Kana and English characters were normalized to a 32×32 size. Similar to Chinese character recognition, we also applied the same feature extraction methods to recognize Kana and English characters.

3) Feature Extraction for Numerals: For numeral feature extraction, the image density feature was employed. First, all numeric characters were normalized to a 32×32 size and then divided into 4×4 blocks. The density feature was calculated in each block, and the dimension of the feature was 16. We also recorded the number of the numeric pixels along the divided line. (There were eight divided lines.) We took this value as the added feature, and therefore, the feature dimension turned out to be $16 + 4 + 4 = 22$.

B. Elastic Mesh

The mesh is the unreal grid for area segmentation of a character image. If an image is segmented by horizontal and vertical N lines, then the image is divided into $N \times N$ blocks. Here, each block is called a mesh. If the mesh lines are evenly distributed in the horizontal and vertical directions, this mesh is called a uniform mesh. The opposite pattern is then called an elastic mesh. One difference between the elastic mesh and the uniform mesh is the difference in segmentation. The former adopts even segmentation of the image density, whereas the latter adopts even segmentation of the width or height of an image.

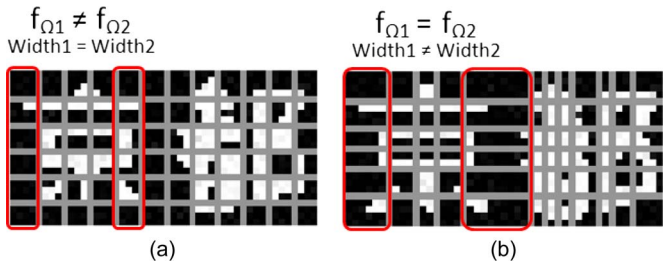


Fig. 20. Uniform and elastic meshes. (a) Uniform mesh. (b) Elastic mesh.

Suppose that $f(x, y)$ is the density of an image and that the size of an image is $M \times N$. The elastic mesh horizontal line I_i and vertical line J_j can be computed by

$$\int_1^M \int_{I_i}^{I_{i+1}} f(x,y)dxdy = \int_1^M \int_{I_k}^{I_{k+1}} f(x,y)dxdy \quad \forall i,k = 1,2,\ldots,N_1 - 1 \tag{12}$$

$$\int_1^N \int_{J_j}^{J_{j+1}} f(x,y)dxdy = \int_1^N \int_{J_k}^{J_{k+1}} f(x,y)dxdy \quad \forall j,k = 1,2,\ldots,N_2 - 1 \tag{13}$$

where N_1 and N_2 are the predefined number of the horizontal and vertical mesh lines, respectively. The preceding equations denote that the density of a character in each horizontal or vertical zone is the same.

The elastic mesh is obtained via the histogram process. Based on the distribution of the character stroke, the elastic mesh is obtained by even segmentation of the projection histogram. The histogram is evenly segmented so that the character image is unevenly segmented. Fig. 20 shows how to obtain a uniform and elastic mesh. f_Ω denotes the character density in the Ω area (area framed in red). From this figure, we can see that the density is the same in each elastic mesh, but the width of the area is different. This is unlike that seen in the uniform mesh. Since the elastic mesh is finely divided for a character's density, the elastic mesh is more appropriate for local deformation than the uniform mesh.

For a real image, the extracted characters unavoidably present problems such as distortion, deformation, and adhesion of the strokes. Feature extraction in the elastic mesh can compensate for these shortcomings. In our recognition system, the elastic mesh combined with the feature extraction approaches is applied as the input features.

The appeal of SVM lies in its strong connection to the underlying statistical learning theory [54]. According to the structural risk minimization principle, a function that can

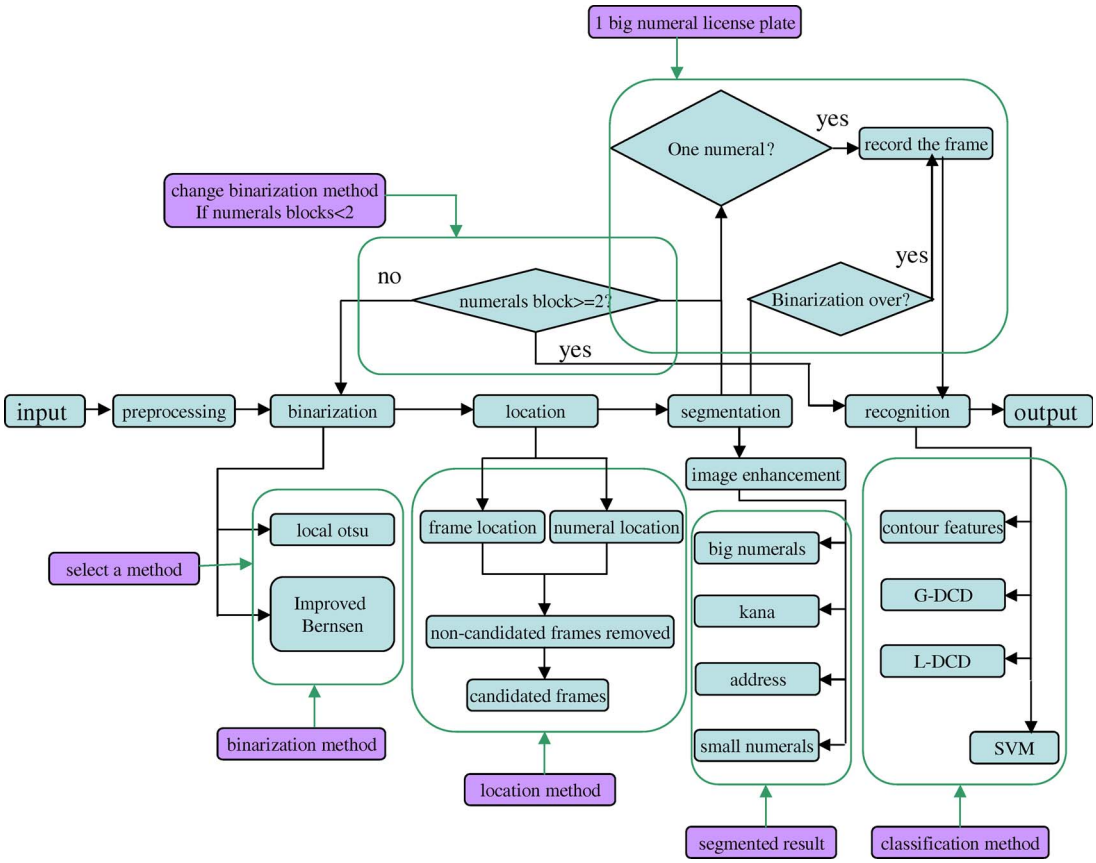


Fig. 21. Structure of the LPR system.

accurately classify training data and belongs to a set of functions with the lowest capacity (particularly in the VC dimension) will, at best, generalize, regardless of the dimensionality of the input space. In the case of a canonical hyperplane, minimizing the VC dimension corresponds to maximizing the margin. As a result, for many applications, SVM has been shown to provide a better generalization performance than traditional techniques. In this paper, we employed SVM as a classifier to recognize characters.

Based on the preceding statement, Fig. 21 shows the structure of the LPR system. When the system detects an input image, preprocessing is the first step including binarization, noise removal, etc. Some candidate license plates are then extracted from the image using frame location or character location. If no license plate is found, a refused recognition result is given. However, sometimes, a license plate is detected more than once. Next, the system steps into the segmentation and recognition stages. If the license plate is correctly segmented, the recognition stage will be processed. Otherwise, the system analyzes another license plate candidate. In the segmentation stage, there are two judgements, i.e., one large numeral plate and a binary alteration procedure. The former is a judgement of only one large numeral on a white frame plate. The latter runs the procedure once again via another binary method when the plate using the current binary method cannot be effectively segmented. During the character recognition stage, four types of characters are recognized. The final recognition result is integrated with different recognition methods by election. The

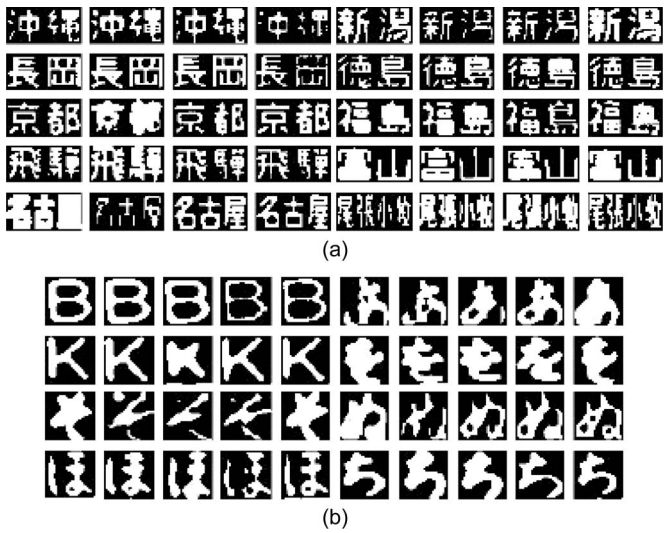


Fig. 22. Samples of address character strings and Kana characters.

recognition results of all of the characters will be considered to form a valid license number if they agree with the compositional semantics of the license numbers mentioned earlier.

IV. Performance of the System and Problem Analysis

In this paper, our experiments were tested on a Microsoft Windows XP operating system, Intel Core 1.8G central

TABLE II
DIFFERENT FEATURE PERFORMANCE COMPARISON

Feature	Dimension	Recognition rate (%)	Processing time (s/character)
G-DCD	64	97.03	0.018
L-DCD	128	96.95	0.010
Contour	48	97.28	0.005
Penetrated	12	96.03	0.001
Fast Zernike	50	98.05	0.138
PDC	1536	97.12	0.156

processing unit and 1.5G random access memory. The software implementation was realized using the Microsoft Visual C++ environment.

A. Character Recognition Approaches and Results

In our research, the character database contains four types of characters: Chinese character strings, Kana or English characters, and numerals. All samples were extracted from real license plates; therefore, fuzzy samples and stroke adhesion are noticeable. Some normalized samples are shown in Fig. 22. In general, on a license plate, the gray value of characters is similar, and the gray value difference between characters and the background is also similar. Furthermore, information on binary characters may be well preserved, even if noise is removed. Hence, feature extraction is simple and effective on binary characters, compared with gray-value characters. In a practical application using the Japanese license plate, the classes of address character strings, Kana, and numerals are 139, 51, and 10, respectively. In our character database, the sample number of the address character strings, Kana, and numerals are 8248, 7877, and 4145, respectively.

First, we completed some experiments comparing different feature performances. We randomly selected 30(samples) $\times$ 50(classes) in an address character string database as the training set, and we selected 1000 samples for testing purposes. To compare the feature performances, G-DCD, L-DCD, PDC, contour, penetrated, and fast Zernike wavelet moments [26] were selected. SVM was adopted as the classifier (polynomial kernel $d = 3$). The recognition rates, processing time, and feature dimension of these experiments are listed in Table II. The top recognition rate was 98.05% using the Fast Zernike feature; however, it is time consuming due to computation complexity. In a practical LPR, the overall processing time normally does not exceed 0.5 s including the plate location, character segmentation, and recognition. Thus, the simple, effective, and short time features would be an added advantage. Although the optimal recognition rate was achieved by Fast Zernike and PDC, they were not appropriate for our system demand. Hence, we chose G-DCD, L-DCD, and the contour features, which achieved more than 97% recognition for Chinese and Kana characters. Although the penetrated feature is the fastest and takes only 1 ms to complete, its recognition rate is lower than that of others. The Japanese license plate contains seven numerals at most and only one address character string and one Kana or English character so that the main time consumption lies in numeral recognition; therefore, we took the penetrated feature for printed numeral recognition.

TABLE III
RECOGNITION RESULTS OF CHINESE CHARACTER STRINGS

Feature	Kernel function	Kernel parameter	Error rate (%) Uniform mesh	Error rate (%) Elastic mesh
G-DCD	Polynomial	d=3	2.37	2.23
	RBF	σ=0.001	2.18	2.21
L-DCD	Polynomial	d=3	3.14	2.99
	RBF	σ=0.001	3.08	2.78
Contour	Polynomial	d=3	2.64	2.58
	RBF	σ=0.00001	1.47	1.21
Penetrated	Polynomial	d=3	4.19	3.86
	RBF	σ=0.001	3.12	2.93

TABLE IV
RECOGNITION RESULTS OF KANA AND ENGLISH CHARACTERS

Feature	Kernel function	Kernel parameter	Error rate (%) Uniform mesh	Error rate (%) Elastic mesh
G-DCD	Polynomial	d=3	1.95	1.93
	RBF	σ=0.001	1.91	1.91
L-DCD	Polynomial	d=3	2.72	2.53
	RBF	σ=0.001	2.57	2.30
Contour	Polynomial	d=3	2.23	1.95
	RBF	σ=0.00001	1.67	1.59
Penetrated	Polynomial	d=3	2.57	2.45
	RBF	σ=0.001	2.47	2.45

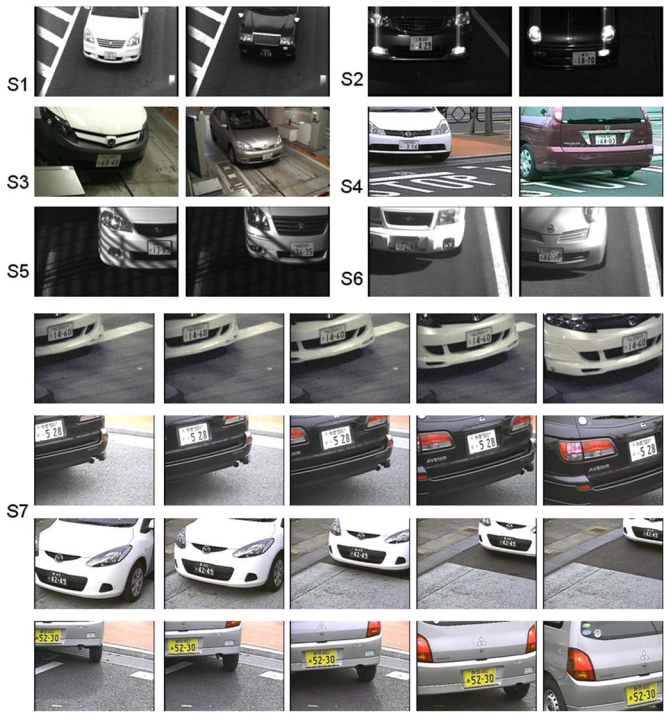


Fig. 23. Samples of the license plate sets.

If there is only one incorrectly recognized character, all of the LPR results will be wrong. We thus analyzed the address string and Kana recognition in detail, because the address string and Kana recognition are more difficult than printed numeral recognition. We randomly selected 30 and 50 samples of each class of address character strings and Kana characters as the training samples, and the rest were testing samples. We repeated the experiments five times. SVM was adopted as the classifier, in which the polynomial and RBF kernel functions were applied. Since the parameter selection of the kernel function is

TABLE V
IMAGE DATABASE COMPOSITION

Samples	camera	Description	Average plate size	Tilt	Images
Sample set 1	digital	road entrance, daylight, moving vehicles	40x 80 pixels	15°	710
Sample set 2	digital	road entrance, night, moving vehicles	40x 80 pixels	15°	86
Sample set 3	analog	indoor, nonmoving vehicles	40x 80 pixels	30°	73
Sample set 4	analog	outdoor, daylight, moving vehicles	various	15°	58
Sample set 5	digital	shadow packing, nonmoving vehicles	40x 80 pixels	15°	124
Sample set 6	digital	road, exposure, moving vehicles	various	25°	357
Sample set 7	digital or analog	parking or road, daylight, moving vehicles	various	various	7618
Total number of images					9026

TABLE VI
ALGORITHM PERFORMANCES OF OUR SYSTEM

Images	Performance in plate location	Performance in plate segmentation	Performance in plate recognition	Overall performance for each sample set
Sample set 1	702/710(98.87%)	684/702(97.43%)	674/684(98.54%)	674/710(94.93%)
Sample set 2	85/86(98.84%)	85/85(100%)	84/85(98.82%)	84/86(97.67%)
Sample set 3	70/73(95.89%)	68/70(97.14%)	66/68(97.06%)	66/73(90.41%)
Sample set 4	56/58(96.55%)	55/56(98.21%)	53/55(96.36%)	53/58(91.38%)
Sample set 5	121/124(97.58%)	120/121(99.17%)	118/120(98.33%)	118/124(95.16%)
Sample set 6	347/357(97.20%)	340/347(97.98%)	335/340(98.53%)	335/357(93.84%)
Sample set 7	7253/7618(95.21%)	7141/7253(98.46%)	6965/7141(97.54%)	6965/7618(91.43%)
Average performance	97.16%	98.34%	97.88%	93.54%

important for the optimal recognition rate, some experiments were implemented to obtain an optimal parameter. Meanwhile, the compared results of the uniform and elastic meshes were also presented. The recognition algorithm, feature extraction method, classifier selection, and parameter setting altogether affected the results. The experimental results are shown in Tables III and IV. From these tables, the performance of the contour feature is better than that of others, and the results of the elastic mesh are better than that of the uniform mesh. The value of the polynomial kernel parameter is easier to define than that of the RBF kernel, but the performance of the polynomial kernel is not better than that of the RBF kernel.

By the preceding experimental comparison, we employed the elastic mesh to extract the contour feature, G-DCD, and L-DCD and used SVM as a classifier to recognize the address character string and Kana characters; thus, the average recognition rate achieved 97.8% and 98.6%, respectively. In our paper, the recognition rates of G-DCD, L-DCD, and contour features achieved over 97% for the Chinese character string or Kana characters, and the processing time did not exceed 0.01 s per character. For numeral recognition, we selected the image density feature and used model matching, perception, and SVM as the three classifiers, and the average numeral recognition rate achieved 99.5%.

B. Performance of LPR

The complete testing image database consists of 9026 digital images from seven sets. The majority of the images represent Japanese license plates from natural scenes obtained under various illumination conditions. In Fig. 23, sample sets 1 and 2 contain the actual images acquired on the road and were obtained by a digital camera adjusted to acquire 640 × 480 pixel images. The images were not processed in real time but stored for later processing, and the size of the license plate was about 40 × 80 pixels (the average size of a Kana character was about 3 × 7 pixels). Sample sets 3 and 4 represent natural scenes of nonmoving vehicles manually obtained from various parking areas using an analog camera adjusted to acquire 1024 × 768 pixel images. The size of the license plate varied from 40 × 80 pixels to 100 × 200 pixels. Sample sets 5 and 6 represent shadow and exposure images from the road obtained by a digital camera adjusted to acquire 780 × 640 pixel images. Sample set 7 represents the dynamic acquisition images containing vehicles from various application setups, and the image size varied from 640 × 480 pixels to 1024 × 768 pixels. The size of the license plate and angle of orientation fluctuated, because the distance between the digital camera and the vehicle varied from the vehicle appearing in front of the camera. Table V indicates the composition of the complete image database.

TABLE VII
PROCESSING TIME OF OUR SYSTEM

Processing model	Average time (milliseconds)
License plate location	158
License plate segmentation	35
License plate recognition	91
TOTAL TIME	284

As far as the overall performance calculation is concerned, this can be achieved by calculating the percentage of license plates that have been correctly identified by the machine and verified by a person supervising the test sample. This percentage is calculated by

$$A = (L \times S \times R)\% \tag{14}$$

where A is the total system accuracy, and L, S, and R are the percentage of successful plate locations, segmentations, and recognitions (successful recognition of all characters on the plate), respectively. Table VI depicts the performance of the proposed system for every sample set. The location, segmentation, recognition, and overall performances for each sample set are presented. For sample sets 1, 2, and 5, because the factors

Fig. 24. Samples that our system unable to process. (a) Dirty plate. (b) low resolution plate. (c) One big numeral with no frame.

influencing image quality such as angle of orientation, license plate size, and digital camera were stable, the rates of 94.93%, 97.67%, and 95.16% were achieved, respectively. Since sample sets 3 and 4 were captured by analog camera from various angles of orientation or surroundings, the low performances achieved were 90.41% and 91.38%. The performances of sample sets 5 and 6 were better than that of sample sets 3 and 4 but not more than that of sample sets 1, 2, and 5. Sample 7 contained 7618 images, which were the dynamic acquisition images under various illumination scenes and surroundings, and it also attained 91.43%. Finally, the average location, segmentation, recognition, and overall performances of LPR achieved 97.16%, 98.34%, 97.88%, and 93.54%, respectively. Overall, the LPR still achieved good recognition results.

Table VII reports the required processing time for every step of the proposed algorithm. The average time of locating the plate, segmenting the character, LPR, and total processing time were 158, 35, 91, and 284 ms, respectively. Our system satisfied the demands of the practical application in both recognition performance and processing speed.

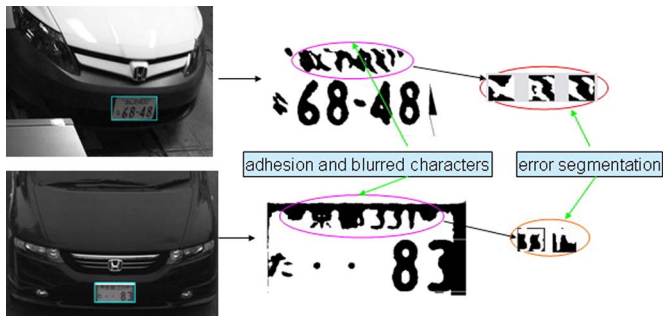


Fig. 25. Samples of the segmentation problem.

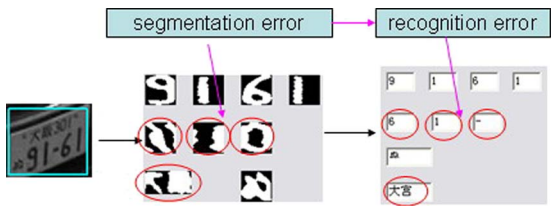


Fig. 26. Samples of the recognition problem.

C. Problems Identified: Restrictions of the System

In our LPR system, some restrictions were analyzed.

1) Physical appearance of the plates. Most of the images failed to identify representative plates without easily distinguishable characters, either due to plate damage or physical appearance (i.e., extremely dirty plates or ones with stickers and unofficial stamps affixed on their surface). The remaining failed cases were due to a sudden change in the environmental illumination condition or incorrect settings during the fine tuning of the experimental setup. In such an outdoor environment, illumination not only changes slowly as daytime progresses, but it may rapidly change due to changing weather conditions or passing objects (i.e., clouds) [see Fig. 24(a)].
2) Image resolution. For character segmentation and a complete success in character recognition, an average plate resolution of 40 $\times$ 80 pixels was necessary. (The resolution of a Kana character was about 3 $\times$ 7 pixels.) Below that limit, the characters appear blurred and distorted when subjected to plate processing and character extraction. Consequently, given that the license plate had a resolution of at least 40 $\times$ 80 pixels or more, successful identification of the plate and its characters was expected. As the resolution became lower, misclassifications occurred [see Fig. 24(b)].

In our system, some restrictions of the system are presented: 1) Location problem. The algorithm is not fit for the license plate with only one large numeral and no frame [see Fig. 24(c)]. In this case, CCA cannot precisely detect the frame. Fortunately, there are only a few such cases (three cases in 9026 plates). 2) Segmentation problem. Most segmentation errors resulted from low-resolution images. Fig. 25 shows the segmentation error examples as a result of character adhesion due to low-resolution images. 3) Recognition problem. In general, segmentation error causes unavoidable recognition error. Only a small portion of the characters fails to be recognized due to blurred or low-resolution images (see Fig. 26). 4) Restriction of the improved Bernsen algorithm. The binary method is optimal for shadow removal for black characters on a white plate. However, it is not fit for white characters on a black plate, as shown in Fig. 27. Resolving these problems is our future research.

V. Discussion and Future Work

Compared with most previous studies that, in some way, restricted their working conditions, the techniques presented in this paper are much less restrictive. The proposed LPR algorithm consists of three modules: 1) locating the license plates; 2) segmenting the characters; and 3) identifying the license characters. There are several commercial LPR systems whose evaluation is beyond the review capabilities of this paper due to the fact that their operation is strictly confidential, and moreover, their performance rates are often overestimated for promotional purposes.

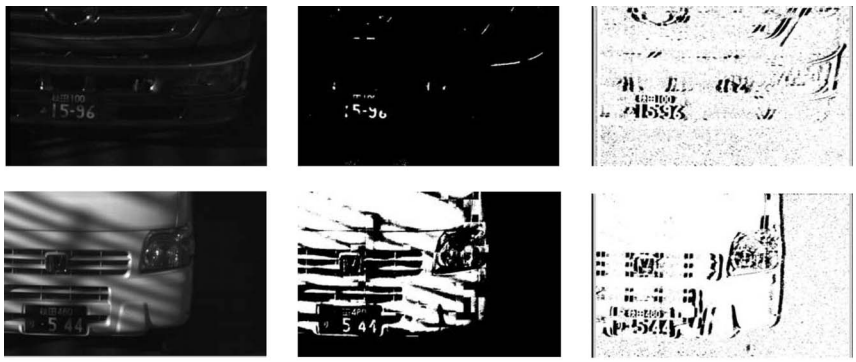


Fig. 27. Failed images processed by the binary methods. (a) White character plate with a shadow. (b) Otsu results. (c) Improved Bernsen algorithm results.

As already discussed, Table I lists the current LPR systems. Compared with the structures presented in Table I, both systems described in [12] and [23] report a better overall performance (93.9% and 93.7%, respectively), whereas our overall performance achieved 93.54%. However, in [12] and [23], as already discussed in the introduction, their systems only process English and numeric characters, which are on one line. Independent of character segmentation and character recognition, the difficulty of our system is greater than that of other systems. The rates of locating the plate, segmenting the characters, and recognizing the characters achieve 97.16%, 98.34%, 97.88%, and 93.54%, respectively. For a single rate or overall performance, our system achieved satisfactory results.

In summary, although the proposed algorithm is concerned with license plates from one specific country, many parts of the algorithm can be readily used with license plates from other countries. Specifically, since uneven illumination commonly influences plate detection, the improved binary algorithm is readily adapted to other surrounding schemes. Since numerals and Roman letters are commonly used to form license numbers, the proposed OCR technique is applicable to any similarly constituted license plates. In addition, OCR for Chinese characters also provides a technique for other applications.

However, our system has restrictions, which will be studied in the future. To make our techniques applicable in less restrictive working conditions, further efforts will be concentrated on improving the license plate processing speed.

ACKNOWLEDGMENT

The authors would like to thank Japanese Omron Company for providing information and images.

REFERENCES

[1] Y. P. Huang, C. H. Chen, Y. T. Chang, and F. E. Sandnes, "An intelligent strategy for checking the annual inspection status of motorcycles based on license plate recognition," *Expert Syst. Appl.*, vol. 36, no. 5, pp. 9260–9267, Jul. 2009.

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

[3] B. Chen and H. H. Cheng, "A review of the applications of agent technology in traffic and transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 2, pp. 485–497, Jun. 2010.

[4] N. H. C. Yung, K. H. Au, and A. H. S. Lai, "Recognition of vehicle registration mark on moving vehicles in an outdoor environment," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 1999, pp. 418–422.

[5] S. Cherng, C. Y. Fang, C. P. Chen, and S. W. Chen, "Critical motion detection of nearby moving vehicles in a vision-based driver-assistance system," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 70–82, Mar. 2009.

[6] O. A. Omitaomu, A. R. Ganguly, B. W. Patton, and V. A. Protopopescu, "Anomaly detection in radiation sensor data with application to transportation security," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 324–334, Jun. 2009.

[7] P. Davies, N. Emmott, and N. Ayland, "License plate recognition technology for toll violation enforcement," in *Proc. IEE Colloq. Image Anal. Transp. Appl.*, 1990, pp. 7/1–7/5.

[8] Y. S. Huang, Y. S. Weng, and M. C. Zhou, "Critical scenarios and their identification in parallel railroad level crossing traffic control systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 968–977, Dec. 2010.

[9] K. Yamaguchi, Y. Nagaya, K. Ueda, H. Nemoto, and M. Nakagawa, "A method for identifying specific vehicles using template matching," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 1999, pp. 8–13.

[10] L. Salgado, J. M. Menendez, E. Rendon, and N. Garcia, "Automatic car plate detection and recognition through intelligent vision engineering," in *Proc. IEEE Int. Carnahan Conf. Security Technol.*, 1999, pp. 71–76.

[11] G. Adorni, F. Bergenti, and S. Cagnoni, "Vehicle license plate recognition by means of cellular automata," in *Proc. IEEE Int. Conf. Intell. Veh.*, 1998, pp. 689–693.

[12] J. B. Jiao, Q. X. Ye, and Q. M. Huang, "A configurable method for multistyle license plate recognition," *Pattern Recognit.*, vol. 42, no. 3, pp. 358–369, Mar. 2009.

[13] 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.

[14] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neural-network-based license plate recognition system," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2675–2683, Sep. 2008.

[15] F. Wang, L. C. 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.

[16] J. M. Guo and Y. F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1417–1424, May 2008.

[17] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. Veh. Technol.*, vol. 44, no. 4, pp. 790–799, Nov. 1995.

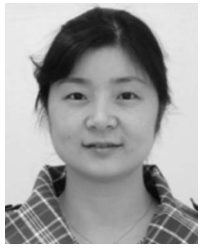
[18] W. J. Jia, H. F. Zhang, and X. J. He, "Region-based license plate detection," *J. Netw. Comput. Appl.*, vol. 30, no. 4, pp. 1324–1333, Nov. 2007.

[19] T. Naito, T. Tsukada, K. Yamada, K. Kozuka, and S. Yamamoto, "Robust license-plate recognition method for passing vehicles under outside environment," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2309–2319, Nov. 2000.

[20] V. Shapiro, G. Gluhchev, and D. Dimov, "Towards a multinational car license plate recognition system," *Mach. Vis. Appl.*, vol. 17, no. 3, pp. 173–183, Aug. 2006.

[21] M. S. Pan, J. B. Yan, and Z. H. Xiao, "Vehicle license plate character segmentation," *Int. J. Autom. Comput.*, vol. 5, no. 4, pp. 425–432, Oct. 2008.

[22] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, Sep. 2006.
[23] 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. 42–52, Mar. 2004.
[24] 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.
[25] Z. G. Xu and H. L. Zhu, "An efficient method of locating vehicle license plate," in *Proc. 3rd ICNC*, 2007, pp. 180–183.
[26] A. Broumandnia and J. Shanbehzadeh, "Fast Zernike wavelet moments for Farsi character recognition," *Image Vis. Comput.*, vol. 25, no. 5, pp. 717–726, May 2007.
[27] T. Akiyama and N. Hagita, "Automatic entry system for printed documents," *Pattern Recognit.*, vol. 23, no. 11, pp. 1141–1154, 1990.
[28] H. F. Zhang, W. J. Jia, X. J. He, and Q. Wu, "A fast algorithm for license plate detection in various conditions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 8–11, 2006, vol. 3, pp. 2420–2425.
[29] 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*, Taipei, Taiwan, Mar. 21–23, 2004, pp. 737–742.
[30] G. H. Ming, A. L. Harvey, and P. Danelutti, "Car number plate detection with edge image improvement," in *Proc. 4th Int. Symp. Signal Process Appl.*, 1996, vol. 2, pp. 597–600.
[31] S. Luis, M. Jose, R. Enrique, and G. Narucuso, "Automatic car plate detection and recognition through intelligent vision engineering," in *Proc. IEEE 33rd Int. Carnahan Conf. Security Technol.*, 1999, pp. 71–76.
[32] Y. Zhang and Y. Pan, "A new approach for vehicle license plate locating from color image," *J. Image Graph.*, vol. 6, no. 4, pp. 374–377, 2001.
[33] Q. Zuo and Z. Shi, "A real-time algorithm for license plate extraction based on mathematical morphology," *J. Image Graph.*, vol. 8, no. 3, pp. 281–285, 2003.
[34] X. Shi, Y. He, and M. Cai, "Application of genetic algorithm in license plate location," *J. Highway Transp. Res. Develop.*, vol. 17, no. 2, pp. 33–36, 2000.
[35] H. A. Hegt, R. J. Dela Haye, and N. A. Khan, "A high performance license plate recognition system," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 1998, vol. 5, pp. 4357–4362.
[36] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.
[37] J. Bernsen, "Dynamic thresholding of gray-level images," in *Proc. 8th Int. Conf. Pattern Recog.*, Paris, France, 1986, pp. 1251–1255.
[38] W. Niblack, *An Indroduction to Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, 1986, pp. 115–116.
[39] B. He, T. Y. Ma, Y. J. Wang, and H. L. Zhu, *Visual C++ Digital Image Processing*. Beijing, China: Posts Telecom, 2002.
[40] A. Rosenfeld and J. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, no. 4, pp. 471–494, Oct. 1996.
[41] M. H. T. Brugge, J. H. Stevens, J. A. G. Nijhuis, and L. Spaanenburg, "License plate recognition using DTCNNs," in *Proc. 5th IEEE Int. Workshop Cellular Neural Netw. Appl.*, 1998, pp. 212–217.
[42] D. S. Gao and J. Zhou, "Car license plates detection from complex scene," in *Proc. 5th Int. Conf. Signal Process.*, 2000, vol. 2, pp. 1409–1414.
[43] J. C. H. Poon, M. Ghadiali, G. M. T. Mao, and L. M. Sheung, "A robust vision system for vehicle license plate recognition using grey-scale morphology," in *Proc. IEEE Int. Symp. Ind. Electron.*, 1995, vol. 1, pp. 394–399.
[44] J. A. G. Nijhuis, M. H. T. Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema, and M. A. Westenberg, "Car license plate recognition with neural networks and fuzzy logic," in *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 5, pp. 2232–2236.
[45] K. Fukunaga and D. L. Kessell, "Estimation of classification error," *IEEE Trans. Comput.*, vol. C-20, no. 12, pp. 1521–1527, Dec. 1971.
[46] H. C. Palm, "A new method for generating statistical classifiers assuming linear mixtures of Gaussian densities," *Pattern Recognit.*, vol. 2, pp. 483–486, Oct. 1994.
[47] S. B. Cho, "Neural-network classifiers for recognizing totally unconstrained handwritten numerals," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 43–53, Jan. 1997.
[48] M. J. Er, S. Wu, and H. L. Toh, "Face recognition with RBF neural networks," *IEEE Trans. Neural Netw.*, vol. 13, no. 3, pp. 697–710, May 2002.
[49] Y. Wen, Y. Lu, and P. F. Shi, "Handwritten Bangla numeral recognition system and its applicaiton to postal automation," *Pattern Recognit.*, vol. 40, no. 1, pp. 99–107, Jan. 2007.
[50] Y. Cui and Q. Huang, "Character extraction of license plates from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1997, pp. 502–507.
[51] B. Zhang and S. N. Srihari, "Fast k-nearest neighbor classification using cluster-based trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 525–528, Apr. 2004.
[52] Y. Wen and P. F. Shi, "A novel classifier for handwritten numeral recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pp. 1321–1324.
[53] Y. Hamamoto, S. Uchimura, and S. Tomita, "A bootstrap technique for nearest neighbor classifier design," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 1, pp. 73–79, Jan. 1997.
[54] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

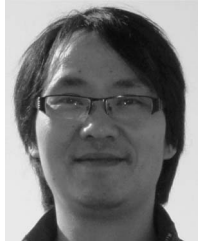


Ying Wen received the B.Sc. degree in industry automation from Technology of Hefei University, Hefei, China, in 1997, the M.Sc. degree in imaging processing and pattern recognition from Shanghai University, Shanghai, China, in 2002, and the Ph.D. degree in imaging processing and pattern recognition from Shanghai Jiao Tong University in 2009.

From 1997 to 1999, she was an Engineer with Jiaoling Motor Company Limited. From 2002 to 2009, she was a Senior Engineer with the Ministry of Posts and Telecommunications. She currently does research as a Postdoctoral Research Fellow with the Pediatric Brain Imaging Laboratory, Columbia University, New York, NY. Her research interests include documentary processing, pattern recognition, machine learning, and medical image processing.

Yue Lu (M'05) received the B.S. degree in wireless technology and the M.S. degree in telecommunications and electronic systems from Zhejiang University, Hangzhou, China, in 1990 and 1993, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from Shanghai Jiao Tong University, Shanghai, China, in 2000.

He is currently a Professor and Chairman with the Department of Computer Science and Technology, East China Normal University, Shanghai. He was a Research Fellow with the Department of Computer Science, National University of Singapore, Singapore. He is also currently the Deputy General Engineer with Shanghai Research Institute of China Post and Founding Director of the East Normal University-Shanghai Research Institute of China Post Joint Laboratory for Pattern Analysis and Intelligent Systems. He has contributed to more than 80 reviewed publications in journals and conference proceedings. His research interests include document image recognition and retrieval, natural language processing, biometrics, and intelligent system development.



Jingqi Yan received the B.S. degree in automatic control and the M.S. and Ph.D. degrees in pattern recognition and intelligent systems from Shanghai Jiao Tong University, Shanghai, China, in 1996, 1999, and 2002, respectively.

Since 2002, he has been with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University. His current research interests include computer graphics and visualization, biometrics, intelligent transport systems, and image processing.

Zhenyu Zhou received the B.Sc., M.Sc., and Ph.D. degrees in biomedical engineering from Southeast University, Nanjing, China, in 2004, 2006, and 2009, respectively.

From 2007 to 2008, he was a Visiting Scholar with McKnight Brain Institute, University of Florida, Gainesville. He is currently a Postdoctoral Research Fellow with Columbia University, New York, NY. He is a contributing author of more than 30 journal and conference proceeding papers. His research interests include biological science and medical engineering, particularly image and signal processing, functional magnetic resonance imaging, and diffusion tensor imaging.

Dr. Zhou is a Student Member of the International Society for Magnetic Resonance in Medicine. He is a Council Member of the Research Center of Learning Science, Nanjing.

Karen M. von Deneen received the AAS and BUS degrees from Morehead State University, Morehead, KY, in 1997 and 1998, respectively, the M.S. degree in animal science from Oregon State University, Corvallis, in 2002, and the Ph.D. degree in pathobiology and large animal clinical sciences from the University of Florida, Gainesville, in 2009.

From 2007 to 2008, she was a Visiting Scholar with Xidian University, Xi'an, China, and the Institute of Automation, Graduate School of the Chinese Academy of Sciences, Beijing, China. She is currently an Associate Professor with the School of Life Sciences and Technology, Xidian University. She is a contributing author to at least 16 journal articles. She has been an invited editor for two textbooks and numerous journal articles. Her research interests are functional magnetic resonance imaging and neuroimaging applications.

Pengfei Shi (SM'09) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 1962 and 1965, respectively.

In 1980, he was with the Institute of Image Processing and Pattern Recognition (IPPR), SJTU. During the past 30 years, he has worked in the area of image analysis, pattern recognition, and visualization. He has published more than 100 papers. He is currently the Director of the Institute of IPPR, SJTU, and a Professor of pattern recognition and intelligent systems with the Faculty of Electronic and Information Engineering.