IJAMML 16:1 (2022) 51-76 September 2022
ISSN: 2394-2258
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/ijamml_7100122251

# DESIGN OF ROAD LICENSE PLATE RECOGNITION SYSTEM BASED ON MATLAB

**Ziyi Wang, Fei Li, Jiahua Peng, Kexin Wang and Jianqiang Gao**

School of Medical Information Engineering, Jining Medical University, Rizhao Shandong 276826, China

## Abstract

With the growth of regional economy, more and more ordinary families own private cars. The increase in the number of cars brings convenience to people and puts pressure on vehicle management. In the urban traffic system, many places that need to register and manage vehicles are still judging vehicle information through human eyes. The method of observing only with the human eye is not only inefficient, labor-intensive, and prone to errors. Based on the MATLAB software environment, this paper introduces a road license plate recognition system, which can detect the vehicle information, automatically capture the license plate position and output the license plate content, which can improve the accuracy and speed of obtaining the license plate number.

*Corresponding author.
*E-mail address*: jianqianggaohh@126.com (Jianqiang Gao).


2020 Mathematics Subject Classification: 68T10, 68U10.
Received June 10, 2022

*Keywords*: license plate number recognition, image processing, GUI design, MATLAB simulation technology.

## 1. Introduction

The license plate of a car is the identification of each car information. The main way to accurately judge the vehicle information is to identify the license plate information. License plate recognition refers to the process of recognizing the license plate information from the images by performing various processing on the captured car images.

Compared with traditional methods, the license plate recognition system makes up for its shortcomings and has the following advantages:

(1) License plate recognition is efficient. Manually input a car image, the system can automatically determine the position of the license plate, segment the license plate and display the recognition result on the screen. From software startup to the output of recognition results, it only takes a few seconds.

(2) License plate recognition is accurate. The traditional way of making judgments by the naked eyes is not only inaccurate, but also prone to fatigue. The license plate recognition system places the numbers and characters in different template libraries, normalizes the characters and uses the template matching method for character recognition, which can achieve high accuracy.

(3) License plate recognition is malleable. License plate recognition technology will be widely used in many occasions, such as community, shopping mall access control, vehicle violation detection, crime detection, etc., so it has strong ductility.

## 2. The Composition of License Plate Recognition System

The license plate recognition system is a comprehensive application of various technical methods in the field of image processing. According to functional requirements, the system will mainly include five parts: Image acquisition, Image preprocessing, License plate positioning and segmentation, Character segmentation and Character recognition [1]. The contents of these five parts are briefly described below:

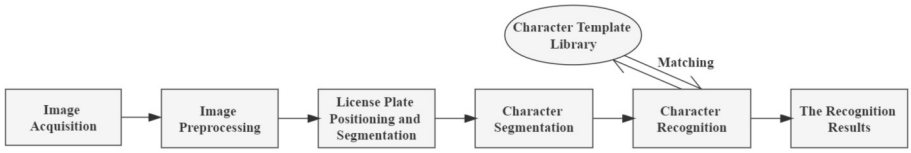



**Figure 1.** License plate recognition process.

**(1) Image acquisition:** Images are captured by external devices such as video cameras or digital cameras, and stored in a folder. When using the system, open the folder and manually input these car images.

**(2) Image preprocessing:** Image preprocessing is a prior operation performed on the image before other work in order to optimize the image quality of the region of interest, including a series of operations such as Image denoising, Image enhancement, and Grayscale histogram adjustment. In this system, Image preprocessing includes Image grayscale, Histogram equalization, Edge detection and other steps [2].

**(3) License plate positioning and segmentation:** This step is divided into two sections: License plate positioning and License plate segmentation. After the Image preprocessing is completed, the main part of the license plate needs to be segmented from the whole image. The license plate positioning adopts the vertical projection method [3]. After the positioning is completed, the small noise in the image is further removed, the area distribution that meets the conditions is counted, the total number of areas is displayed, and the license plate is segmented.

**(4) Character segmentation:** The Character segmentation of the license plate is the main factor affecting the success of the recognition. Since most of the small cars use the license plate with blue background and white characters, the colour segmentation method is adopted. A reasonable threshold is set according to the background colour of the license plate, and the number of pixels in this colour range is counted in the row and column directions to determine a reasonable area for segmentation [4].

**(5) Character recognition:** There are two commonly used methods for Character recognition: Template matching method and Character recognition method based on neural network. The template matching method is adopted in this paper. This method is to cover the original stored character template on the entire image, make the difference between the segmented character and the template, and judge the similarity between the character and the standard template according to the difference. It is simple in principle, easy to operate. After the recognition is completed, the system will output the recognition result.

## 3. Technical Selection of License Plate Recognition System

This system takes MATLAB GUI (Graphical User Interface) as the main technical framework. MATLAB GUI refers to the use of graphical methods to display the computer operation user interface. It is a visual and interactive window for MATLAB users. It is generally composed of graphical controls such as windows, drop-down menus or dialog boxes [5].

Using a GUI interface for license plate recognition has the following advantages:

(1) Good human-computer interaction interface. Designers can modify the functions of each control by viewing the callback function ("*callback*"), and do not need to read the cumbersome code to operate the interface.

(2) It has complete graphics processing functions to realize the visualization of calculation results and programming. MATLAB has powerful image processing functions and provides a variety of image processing functions, which are in line with the theme of license plate recognition. Many functions such as "*rgb2gray*" (image grayscale) and "*graythresh*" (binarization threshold function) can be called directly.

(3) Friendly user interface and naturalized language are easy for learners to understand and master. The design of the system will be carried out on the entire operation interface, and the operation of each button and dialog box will be simple and easy to understand. Even if the user lacks professional knowledge, it is easy to understand the operational methods when they use it.

## 4. Analysis of License Plate Recognition System

The license plate recognition system mainly includes five parts: Image acquisition, Image preprocessing, License plate positioning and segmentation, Character segmentation and Character recognition [1].

As shown in Figure 2, on the right side of the interface are various operations on the image, including Image grayscale, Edge detection, and License plate segmentation. Generally, the operations are performed in order, from top to bottom, and the running results will be displayed in real time on each coordinate axis on the left. The "*CLEAR*" *Button* is used to clear the identified results for the convenience of the next step.

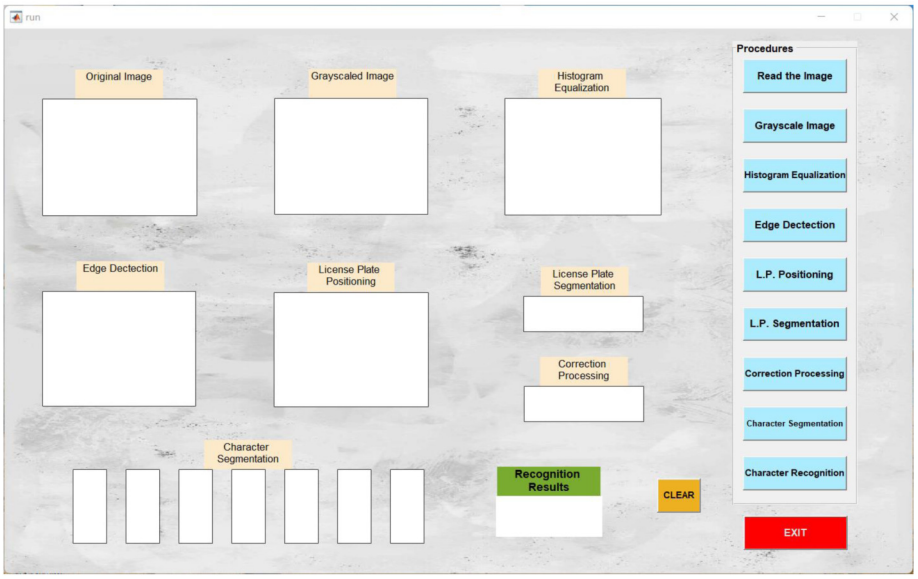



**Figure 2.** The interface of the system.

Clicking the *"EXIT" button* will shut down the system.

The principle of each module will be described in detail below [6]:

### 4.1. Image acquisition

Use the "*uigetfile*" function to open a dialog that lists the files in the current folder. The name of the file can be selected or entered manually. In order to enhance the applicability of the system, many different types of pictures such as *jpg.*, *jpeg.*, and *bmp.* can be added for selection.

### 4.2. Image preprocessing

Image preprocessing includes three steps: Image grayscale, Histogram equalization, and Edge detection. Good preprocessing facilitates subsequent operations on images.

(1) Grayscale image: Most of the car images captured by the machine are color images. Direct processing of colour images will waste a lot of system resources, aggravate computer operation, and cause slow operation [7]. Therefore, grayscale processing is required. In this system, the *"rgb2gray*" function is used to convert the colour image to grayscale. The following is the source code used.

```
image = handles.I;
gray = rgb2gray(image);        % Grayscale Image
axes(handles.axes2);
imshow(gray);
```

(2) Histogram equalization: The second step of image preprocessing is Histogram equalization. Histogram equalization is to make the grayscale histogram of the image as evenly distributed as possible, which can effectively adjust the image brightness distribution and enhance the contrast. As shown below, we use the "*histeq*" function to achieve histogram equalization, enhance the image, and name the processed new image "new_gray".

```
image = handles.I;
gray = rgb2gray(image);
new_gray = histeq(gray);    % Histogram Equalization
axes(handles.axes3);        % Contrast Enhance
imshow(new_gray);
```

(3) Edge detection: This step is to detect the edge of the license plate and prepare for Character segmentation. The contamination of the license plate and the influence of the connection between the license plate and the rivet on the edge detection should be considered. Usually, the gray value of the two sides of the edge in the image varies by a large amount. The first-order derivative of the gray value to the coordinate is obtained. When the derivative is greater than a certain threshold, this

point can be regarded as an edge point, and points whose intensity is less than the threshold are ignored [8]. Since the image is enhanced first and then eroded, it is used to eliminate the small interference domain, separate the object at the slender point and smooth the edge of the larger object without changing its area. It is decided to use the *Canny* operator for edge detection, which can meet the requirements of the system [9].

The following is the source code of Edge detection, the *"edge"* function is used to call the *Canny* operator for edge detection. When "both" is omitted, it is performed in both horizontal and vertical directions by default [9].

```
if size(new_gray,1)>1000
    new_gray_1 = imresize(new_gray,0.1);
else
    new_gray_1 =new_gray;
end
bw = edge(new_gray_1,'canny'); % Using the edge function
axes(handles.axes4);          % call the Canny operator
imshow(bw);
```

### 4.3. License plate positioning and segmentation

(1) License plate positioning: The license plate positioning adopts the vertical projection method, That is, set a row vector and a column vector, the row vector is 1 row $y$ column, and the column vector is 1 column $x$ row [10]. The image is projected in the horizontal and vertical directions, traversing each pixel, and the length and width of the license plate can be determined according to the shadow, and then the location of the license plate can be obtained.

(2) License plate segmentation: However, there is still a lot of noise interference in the grayscale image at this time. Therefore, for morphological image processing, first use the *"imerode*" function to perform erosion operation on the image, reduce and refine the highlighted noise part in the image, and remove the noise [11]. Then, the small objects are removed by "*bwareaopen*" function to further remove noise. After these two steps of "pre-processing", most of the noise in the grayscale image can be removed, and only the license plate is retained as the region of interest.

Finally, use "*regionprops*" to count the area distribution of the marked area, display the total number of areas, and perform license plate segmentation. The source codes of this step are as follows:

```
se = ones(40);
d = imdilate(d,se);          % Image expansion
d = imerode(d,se);           % Image corrosion, reduce and refine the
                             % highlighted area or white part of the image
d = bwareaopen(d,100);       % Remove small objects
STATS = regionprops(d);      % Count the area distribution of the marked area
area = [];
```

### 4.4. Correction processing

License plate correction is a unique function designed by this system, which can avoid the influence of the tilt of the captured picture. It can automatically determine the tilt direction of the picture and adjust the picture to the level, which is beneficial to the subsequent steps of Character segmentation. Correction processing is performed in both horizontal and vertical directions. Treat the license plate image as a two-

dimensional function $f(x, y)$. The linear integral of $f(x, y)$ in the vertical direction is its projection on the $x$ axis, the linear integral of $f(x, y)$ in the horizontal direction is its projection on the $y$ axis. As the following contents of this paper, first use the "*edge*" function, call the *sobel* operator to detect the edge of the license plate in the horizontal direction, use the *Radon* transform to calculate the tilt angle, and finally use "*imrotate*" to adjust the image to the horizontal. *Radon* transform is a kind of integral transform, which is a projection transformation of image matrix in the ray direction of a specified angle. It is often used for license plate tilt correction and line dectection (Figure 3) [12].

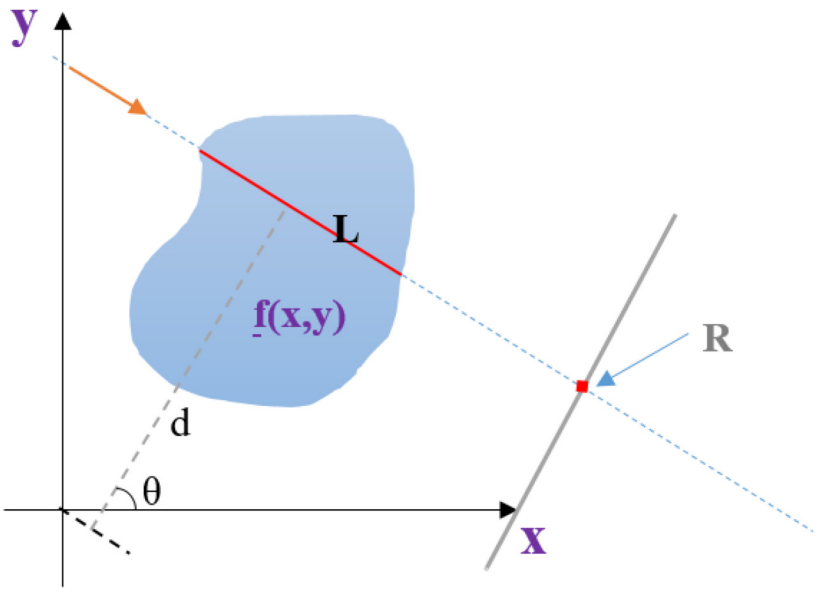



**Figure 3.** A ray determined by θ, $d$ passes through the image $f(x, y)$ to obtain a projection with a brightness value of $R$.

After the correction processing, the borders in the horizontal and vertical directions are removed respectively, and we will obtain the license plate image with only characters.

```
I_gray_bw=edge(I_gray,   'sobel','  % Horizontal correction
horizontal');
theta1=1:180;
[R1,xp1]=radon(I_gray_bw,theta1);  % Radon transform for
                                   tilt correction
[I1,J1]=find(R1>=max(max(R1)));
qxj_hor=90-J1;
I_HOR_JZ=imrotate(I,qxj_hor,
'bilinear','crop');
```

### 4.5. Character segmentation

After the above steps, we segmented the license plate image from the original image, and removed most of the interference, Character segmentation will be performed below. Firstly, we segment the first character, and the first pixel and non-zero column are found first as the left boundary of the first character. Search to the right on this basis, find the first pixel and the column with zero, as the right boundary. Next, judge whether the cropping is abnormal by the ratio of the width of the character found to the overall width of the license plate. If it is abnormal, select the pixels in the left and right boundaries and the smallest column as the right boundary, set it to zero, and perform multiple abnormality detection and correction.

In the same way, the next six characters can be obtained. Since there are still some differences in the divided characters, it is necessary to unify the size of the characters, that is, to perform character normalization. Character normalization can improve the recognition accuracy. In this system, the size of the cropped characters is normalized with the "*imresize*" function, and the image is drawn. License plate character segmentation is the main factor affecting the success of recognition. Some codes for character segmentation are shown below.

```
T=2;
myb = find(sumK>=T);
myf = zeros(1,length(myb)); % Number of columns greater than the threshold
for ii = 2:length(myb)
    if myb(ii)-myb(ii-1)==1
     myb(ii-1) = 0; % Front edge of points with gray value other than 0
    else
     myf(ii) = myb(ii); % Back edge of points with gray value other than 0
    end
end
```

### 4.6. Character recognition

There are two commonly used methods for character recognition: Template matching method and Character recognition method based on neural network. This paper adopts the Template matching method, which is simple in principle and easy to operate. After the recognition is completed, the recognition result is output.

Before recognition, put the characters to be recognized into a string, the 1st-10th elements of the string store numbers 0-9, the 11th-34th elements store letters "*A*"-"*Z*" (except letters "*I*" and "*O*"), the 35, 36$^{th}$ elements store the Chinese characters "*Hu*" and "*Su*". In this way, when the first character is recognized, it is carried out in the 35th and 36th elements, the second character is recognized in the 11-34th element, and the 3-7th character is recognized in the 1-34th element. The templates of Chinese characters and characters are stored separately, which can shorten the running time greatly. In order to improve the recognition accuracy, only some certain Chinese characters are selected.

When identifying, subtract the segmented character from each template. The closer the subtracted value is to 0, the more similar the character is to the template. In each difference, the value closest to 0 is selected as the final recognition result. As shown below:

```
for i = 1:size(images_test_all,2)
    images_test = double(images_test_all(:,i));
    pred(i) = predict(Theta1, Theta2, images_test');
end
chepai = [];
for i = 1:size(pred,2)
   if pred(i)>0   % If conditional statement for judgment
     chepai = [chepai,Name{pred(i)}];
   end
end
```

### 4.7. Clearing of results

In addition, the system also designed a *"CLEAR"* button, which is able to reset and clear the contents of the current run for subsequent operations after the license plate recognition is completed.

On the whole, the final accuracy of license plate recognition depends to a large extent on various preprocessing of the image before recognition [13]. For areas of interest, preprocessing must be performed before license plate segmentation, which can greatly improve the accuracy of license plate recognition. At the same time, in the image denoising stage, it is also possible to consider using a low-pass filter to denoise the image, as well as image enhancement and smoothing of the car image [14], [15]. Besides, consider the influence of uneven illumination or brightness, extreme weather and surrounding scenery on recognition, and improve the anti-interference of the system as much as possible [16].

## 5. Realization of License Plate Recognition System

This paper actually selects 6 car images named *"*A1*"*-*"*A6*"* as a data set to test the function of the system. To test the system's "tilt correction" feature, two images of them were significantly tilted (A2, A3).

### 5.1. Read the image

First select the license plate images to be recognized, click the "*Read the Image*" button, select the images to be recognized in the dialog box, and successfully display the original images, which are shown below named "A1"-"A6".

### 5.2. Image preprocessing

(1) Grayscale image: Click the "*Image Grayscale*" button to grayscale the images, and the grayscaled images will be displayed in axes2. As shown in the figure below named "B1"-"B6".

(2) Histogram equalization: Click the *"Histogram Equalization*" button to perform histogram equalization on the obtained grayscale images. It can be seen from the display results ("C1"-"C6") that the histogram equalization adjusts the images brightness and enhances their contrast.

(3) Edge detection: Click the "*Edge Detection*" button, the system will display the image after edge detection in axes4, and the points where the grayscale change exceeds a certain threshold can be regarded as edge points. It can be seen from the figure ("D1"-"D6") that the connections between the license plate and the body are judged as edge points, and the system can identify them well.

### 5.3. License plate positioning and segmentation

(1) License plate positioning: Click the "*License Plate Positioning*" button, and the system will automatically display the license plate areas, remove most of the noise interferences, as shown below named "E1"-"E6".

(2) License plate segmentation: Click the "*License Plate Segmentation*" button, the system will extract the license plate images from the grayscale images and displays them in axes6. As shown in the figure ("F1"-"F6"), the license plate images have certain inclinations at this time, and still contain unnecessary parts such as the frame.

### 5.4. Correction processing

Then perform correction processing, as shown in the figure below, after clicking the button, we can find that the license plate images are adjusted to the level, and the border and other edge points are removed, and only the main parts of the characters are included ("G1"-"G6").

### 5.5. Character segmentation and recognition

(1) Character segmentation: Click the *"Character Segmentation*" button to cut out the 7 characters contained in the obtained license plate images. As shown ("H1"-"H6"), each character is well positioned and not too much detail is lost.

(2) Character recognition: Then, recognize each character segmented, click the *"Character Recognition*" button, the recognition results will be displayed in the static text box text2, the recognition process takes about 15 seconds, the running screenshots are displayed in pictures "I1"-"I6".

### 5.6. Reset and result clear

To facilitate the next identification, the user can click the *"CLEAR*" button to clear the running result. As shown in the Figure 5.

### 5.7. Exit the system

After the identification, click the *"EXIT*" button to exit the system and clear the running data in the "command window" (Figure 5).

### 5.8. The results of recognition

The recognition results of the 6 pictures are as follows: Figure 4.1 to Figure 4.5 are the results of each step. Judging from the quality of image processing in each step and the final recognition results, this system has removed most of the noise interferences, accurately recognized the license plate information and achieved its original design intents.


**A1**


**B1**


**C1**

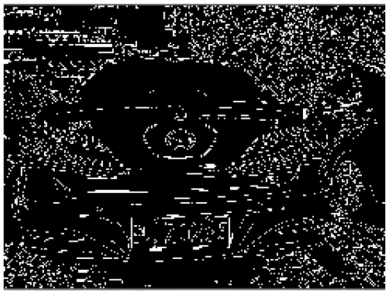
**D1**


**E1**

**A2**

**B2**

**C2**



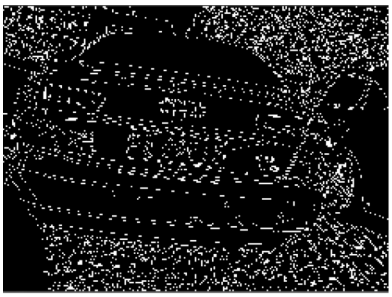
**D2**

**E2**

**A3**

**B3**

**C3**



**D3**

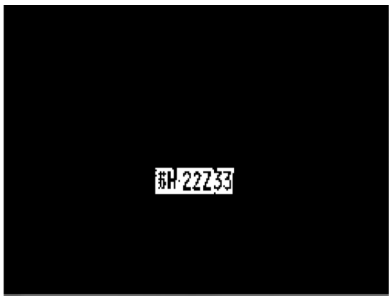
**E3**

**A4**

**B4**

**C4**

**D4**

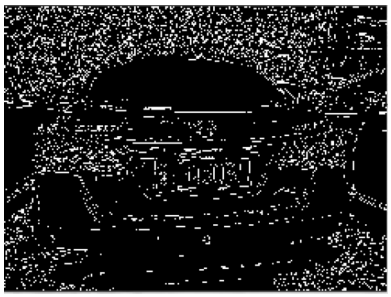


**E4**

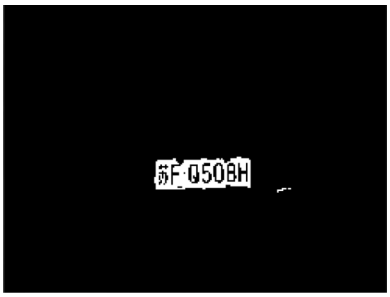

**A5**


**B5**


**C5**



**D5**

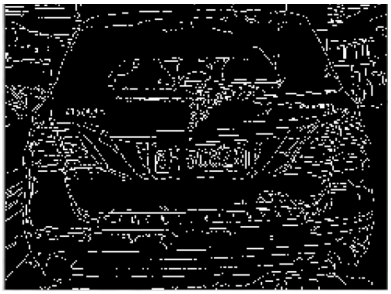



**E5**

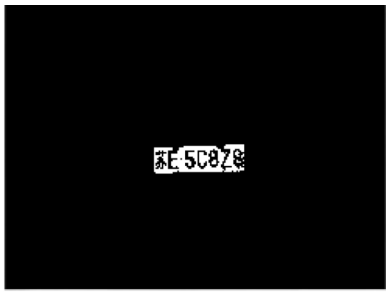
**A6**

**B6**

**C6**



**D6**



**E6**

**Figure 4.1.** The results of license plate recognition.


**Figure 4.2.** The results of license plate segmentation.


**Figure 4.3.** The results of correction processing.

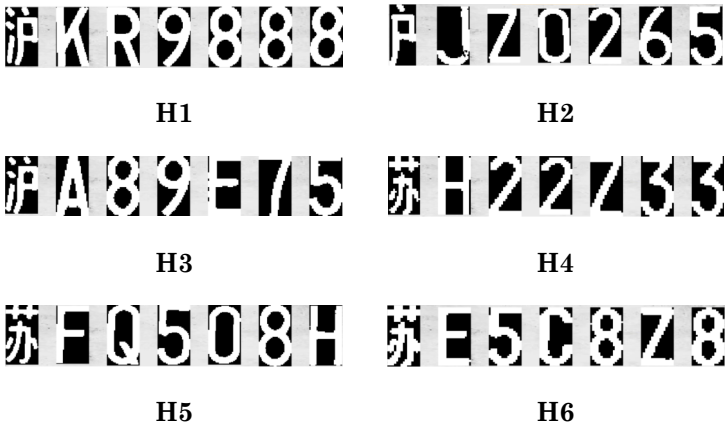



**Figure 4.4.** The results of character segmentation.

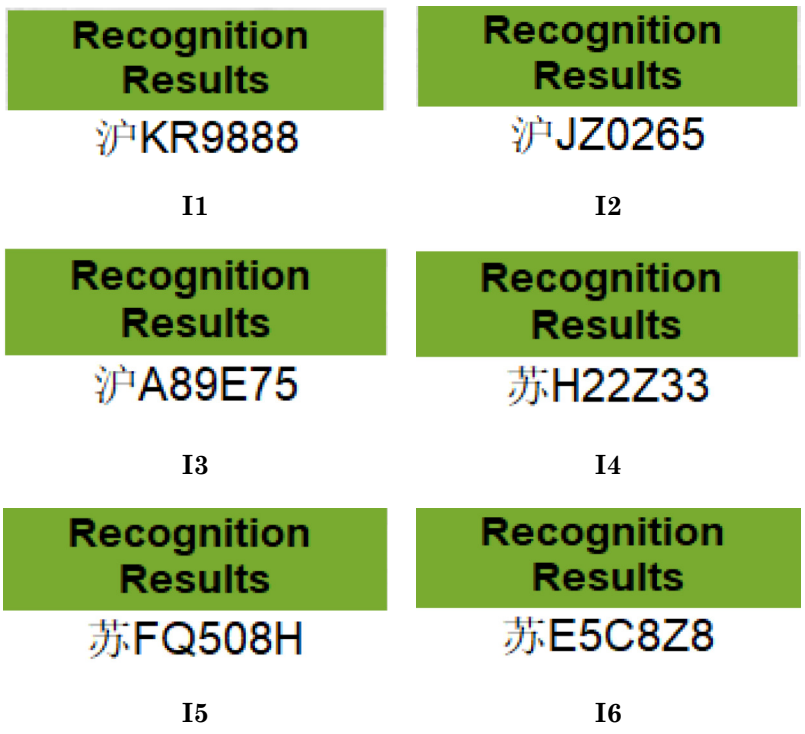



**Figure 4.5.** The results of recognition.

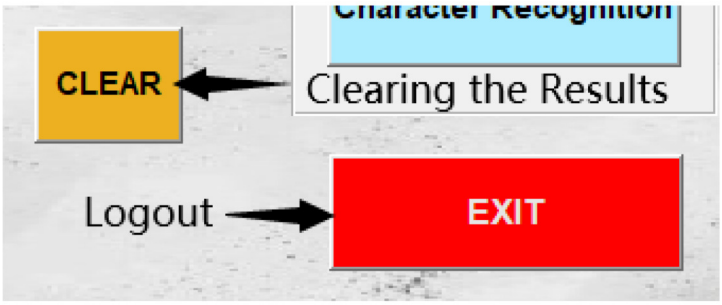



**Figure 5.** "CLEAR" button and "EXIT" button.

## 6. Conclusions

This paper uses MATLAB software as a platform to design a license plate number recognition system. The system has a GUI interface, which can locate the license plate area from the license plate image and identify the license plate content. Through the introduction of the subject of license plate recognition, this paper deeply studies the image processing methods involved in license plate recognition and the advantages of using MATLAB for license plate recognition. License plate recognition mainly includes key steps such as Image preprocessing, Character segmentation, and Character recognition. Image preprocessing includes grayscale processing, denoising processing, and image brightness adjustment, which is the first operation on the image before other work; the key to character segmentation is to select a reasonable threshold, and determine the left and right boundaries of the characters through this threshold; use the template matching method for character recognition, and place letters and Arabic numerals in different templates, which can improve the accuracy of recognition. When designing, a variety of processing methods can often be selected, and different processing sequences can be selected to obtain different effects. The function of the system is tested by using the data set. The edge of the image extracted by edge detection, the denoising effect of the denoising stage and the final recognition result all show that the system can extract the license plate information from the image and achieve the final purpose of license plate recognition.

## Acknowledgements

This work is partly supported by the Supporting Fund for Teachers' research of Jining Medical University under Grant No. JYFC2019KJ014, and the Doctoral Research Foundation of Jining Medical University under Grant No. 2018JYQD03, and the 2022 undergraduate innovation training program of Jining Medical University under Grant No. cx2022087z.

## References


[1] Hu Lina, License plate number recognition system based on MATLAB, Shanxi Electronic Technology 4 (2017), 55-58.

[2] Xiang Tuowen and Liu Liu, An image recognition method, system and process [OL], [2022-5-29].

http://www.xjishu.com/zhuanli/55/201710756031.html

[3] Wang Lijing and Li Feiguo, Research on license plate recognition algorithm using MATLAB, Electronic Testing (21) (2018), 50-41.

[4] Lian Ning and Xu Yanlei, License plate location method based on mathematical morphology and color features, Chinese Journal of Graphics 35(05) (2014), 774-779.

[5] wendy_ya, Introduction and construction of GUI interface of MATLAB [OL], (2020-4-7) [2022-6-14].

https://blog.csdn.net/didi_ya/article/details/105357279

[6] AliMsh [OL], [2022-5-11], Realization of license plate recognition with MATLAB.

https://blog.csdn.net/AliMsh/article/details/82634388

[7] Yin Hengjie and Cao Yan, Research on license plate recognition system based on MATLAB, Electronic Products World 28(12) (2021), 42-44.

[8] Siman Wang, Xiukui Chen, Kai Zhu, Duanduan Feng, Rui Zhang and Tao Zhang, Design of a simple system for recognition of vehicle license plate numbers in university campus, Computer Programming Skills and Maintenance 11 (2021), 108-111.

[9] Yang Cao, Digital image edge detection based on MATLAB, Information Technology and Informatization 2 (2020), 140-142.

[10] Zhang Jinfeng, Design of license plate recognition system based on MATLAB, Information and Computer (Theoretical Edition) 33(02) (2021), 153-155.

[11] Li Mengkun, Ke Zhengxuan and Yu Dingrong, License plate number recognition algorithm for Internet of vehicles based on moving edge calculation, Computer Engineering and Design 42(11) (2021), 3151-3157.

[12] Deepsprings [OL], [2022-6.17], Principle and application of Radon transform.

https://blog.csdn.net/deepsprings/article/details/119321544

[13] Xu Lichen, Ye Chao and Liu Qi, Design of license plate number recognition system based on MATLAB, Value Engineering 34(18) (2015), 181-183.

[14] Sun Chun, License plate recognition based on GUI graphics processing system, Software Guide 20(02) (2021), 210-214.

[15] Zhang Xiaofeng, Xu Fengchang and Su Yan, Research on the license plate recognition based on MATLAB, Procedia Engineering 15 (2011), 1330-1334.

[16] Zhang Yi, Design and implementation of license plate image recognition based on MATLAB, Information and Communication 11 (2019), 79-81.

[17] Wang Keping et al., Digital Image Processing: MATLAB Edition, Beijing: Machinery Industry Press, Volume 7 (2015), 170-172. ISBN 978-7-111-51032-1.

[18] Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Electronic Industry Press, 2020.

[19] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, Digital Image Processing using MATLAB, Second Edition, Electronic Industry Press, 2014.

■