



中国地质大学（武汉）

基于 MATLAB 的车牌识别系统

学 院： 自动化学院

课 程： 数字图像处理

指导老师： 上官星辰

学 号： 20201000128

班 级： 231202

姓 名： 刘瑾瑾

2023 年 12 月 11 日

目录

一、引言	1
二、概要设计	1
2.1 图像采集	1
2.2 图像预处理	2
2.3 车牌定位原理	2
2.4 字符分割原理	2
2.5 字符识别原理	3
三、详细设计	3
3.1 图像预处理	3
3.1.1 灰度变换	3
3.1.2 平滑处理	3
3.1.3 边缘提取	4
3.1.4 腐蚀图像	4
3.1.5 形态学滤波	5
3.2 车牌定位	5
3.3 字符分割	6
3.4 字符识别	7
3.5 界面设计	8
四、总结与讨论	9
附录	10

一、引言

车牌识别系统是指能够检测到受监控路面的车辆并自动提取车辆牌照信息如含汉字字符、英文字母、阿拉伯数字及号牌颜色进行处理的技术。车牌识别是现代智能交通系统中的重要组成部分之一，应用十分广泛。它以数字图像处理、模式识别、计算机视觉等技术为基础，对摄像机所拍摄的车辆图像或者视频序列进行分析，得到每一辆汽车唯一的车牌号码，从而完成识别过程。通过一些后续处理手段可以实现停车场收费管理，交通流量控制指标测量，车辆定位，汽车防盗，高速公路超速自动化监管、闯红灯电子警察、公路收费站等等功能。

数字图像处理技术是通过计算机对于数字化的图像进行优化处理的一项技术，在如今科技快速发展的背景下数字图像处理技术能够充分应用于车辆的车牌识别中，从而实现车牌图像数据采集的精准性。通过对数字图像技术在车牌识别中图像采集、图像预处理、字符分割以及字符识别的应用，来全面增强车牌识别的准确度。

二、概要设计

一个完整的车牌识别系统应包括车辆检测、图像采集、图像预处理、车牌定位、字符分割、字符识别及结果输出等单元。当车辆到达触发图像采集单元时，系统采集当前的视频图像。车辆识别单元对图像进行处理，定位出牌照位置，再将车牌中的字符分割出来进行识别，然后组成车牌号码输出。

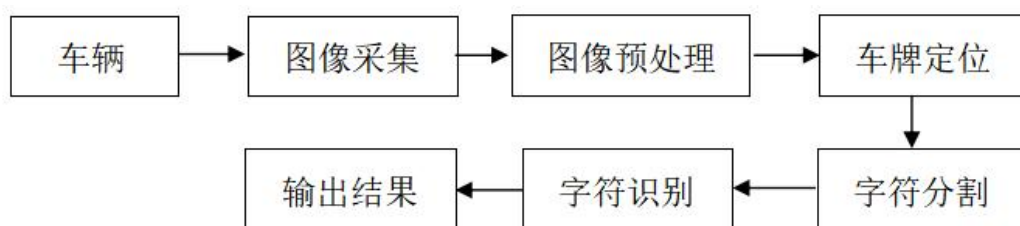


图 1 车牌识别原理图

2.1 图像采集

在车牌识别系统中，图像采集是车牌识别的基础性工作。由于受到汽车自身运动的影响，图像采集设备在采集过程中容易产生图像模糊、图像失真的情况。

同时，图像采集过程中的光照变化、拍摄曝光程度、车牌污染都会对于图像的最终采集情况产生影响。所以我们需要对图像的预处理，提高识别的准确度。

2.2 图像预处理

在车牌图像数据采集之后，便需要对于车牌图像进行预处理，减少外界环境以及拍摄设备对于图像采集的质量影响。首先，需要对于车牌图像进行图像灰度变换处理，将图像中车牌信息的突出特征进行捕捉，并且对于突出信息进行图像特征的增强性技术处理。其次，则是需要进行图像滤波处理，通过降噪来对图像进行平滑式处理，从而更好地实现车牌图像的精准定位。最后，需要对于图像进行灰度图像的二值化处理，这是因为图像拍摄过程中容易存在字符边缘模糊、重影的问题，需要通过二值化处理来将图像中的信息进行分离式处理。

2.3 车牌定位原理

通过对车牌信息图像的采集与处理，能够对车牌的突出信息进行强化，从而更好地实现关键信息的清晰化呈现。在车牌的识别系统中，不仅需要对于车牌的基本信息进行全面的捕捉，还需要根据车牌信息对车牌所在区域进行划分，从而在车牌字符的分割中对车牌图像信息进行精准化定位。首先对采集到的图像进行大范围相关搜索，找到符合汽车车牌特征的若干区域作为候选区，然后对这些候选区域做进一步分析、评判，最后选定一个最佳的区域作为车牌区域，并将其从图像中分割出来。

2.4 字符分割原理

完成车牌区域的定位后，再将车牌区域分割成单个字符，然后进行识别。字符分割一般采用垂直投影法。垂直投影法是基于一种数学形态原理来进行车牌信息的字符分割与定位。在进行水平投影以及垂直投影的分析之后，能够根据图像上字符的上升点与下降点来测算字符的高度与宽度，进而在分析的过程中得出一个合理的阈值，根据相应的阈值参数来进行车牌图像的字符分割与图像定位。

2.5 字符识别原理

车牌图像的字符识别功能是在图像采集、图像预处理、字符分割与特征提取基础上所形成的，其主要将本次采集到的车牌信息与信息库中的信息进行匹配，从而对本次车牌信息进行精准化定位，形成最终的识别结果，本次采用模板匹配的方法。基于模板匹配算法首先将分割后的字符二值化,并将其尺寸大小缩放为字符数据库中模板的大小，然后与所有的模板进行匹配，最后选最佳匹配作为结果。

三、详细设计

3.1 图像预处理

3.1.1 灰度变换

灰度变换是一种图像处理技术，它主要涉及将一幅彩色图像转换为灰度图像。灰度图像是一种只包含亮度信息而不包含颜色信息的图像。图像灰度化的算法包括最大值法、平均值法和加权平均值法三种。在 MATLAB 中，`rgb2gray` 函数采用加权平均法，具体为：

$$G = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B$$



图 2 灰度变换效果图

3.1.2 平滑处理

在 MATLAB 中，通过使用形态学操作中的开操作 `imopen`，可以去除图像

中的小细节，使图像平滑，突出较大的结构。这有助于去除噪声、小斑点或其他不重要的细节，从而增强图像的主要特征。通过使用 `imsubtract` 函数，可以计算原始图像与开操作后的图像之间的差异，以便更好地突出差异性的特征。这种操作通常有助于改善图像的质量，使得后续的处理步骤更加有效。

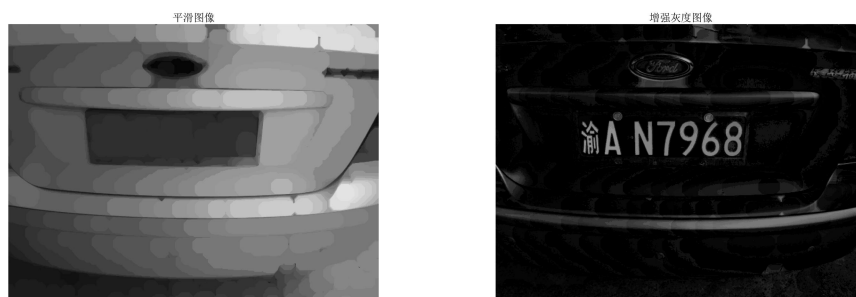


图 3 平滑处理效果图

3.1.3 边缘提取

边缘是以图像的局部特征不连续的形式出现的，也就是指图像局部亮度变化最显著的部分，如灰度值的突变、颜色的突变、纹理结构的突变等。图像边缘有方向和幅度两个特性，通常沿边缘的走向灰度变化平缓，垂直于边缘走向灰度变化剧烈。由于边缘是图像上灰度变化最剧烈的地方，传统的边缘检测就利用这个特点，通过计算图像中像素的梯度值来确定边缘点，常用的图像边缘提取算子有 Roberts 算子、sobel 算子、Prewitt 算子、Laplacian 算子等。

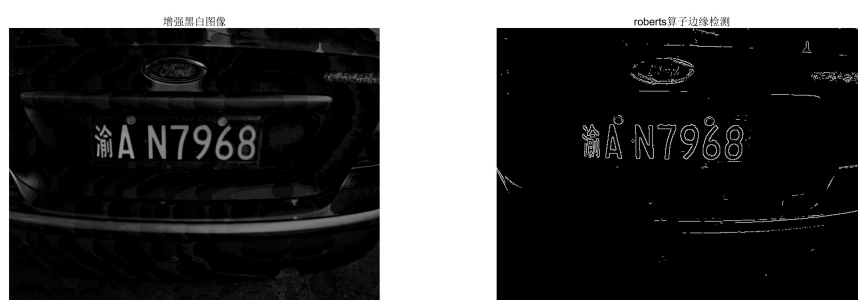


图 4 边缘提取效果图

3.1.4 腐蚀图像

腐蚀是数学形态学中的一种基本操作，它主要用于缩小图像中的亮区域，或

者将物体的边缘进行腐蚀。通过将结构元素在图像上滑动，检查结构元素的覆盖区域，并将该区域的最小值赋给输出图像。这导致图像中的亮区域缩小，边缘变得更加清晰。



图 5 腐蚀图像效果图

3.1.5 形态学滤波

经过上述的处理，牌照区域已经非常明显，其边缘得到了加强。此时可进一步确定牌照在整幅图象中的准确位置。使用形态学滤波的方法，以将开、闭运算结合起来构成形态学噪声滤波器。在本程序中使用开闭两个基本运算进行形态学滤波后，通过 `bwareaopen` 函数来去除对象中不相干的小对象。

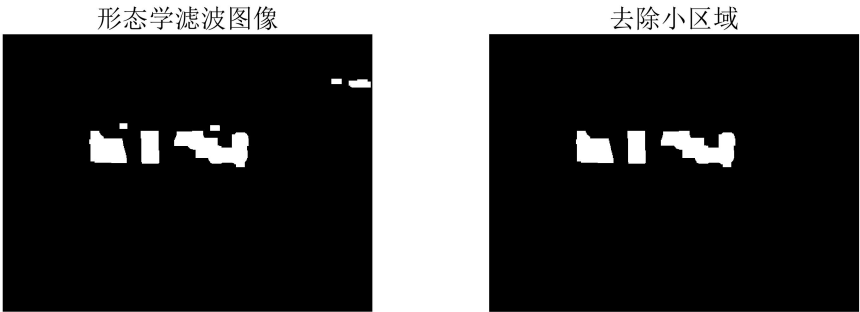


图 6 形态学滤波效果图

3.2 车牌定位

自然环境下，汽车图像背景复杂，光照不均匀，在自然背景中准确地确定牌照区域是整个图像识别过程中的关键。算法流程如下：

- (1)对二值图像进行区域提取。计算并比较区域特征参数，提取车牌区域。进行闭运算，可使得水平相邻的边缘连接成为连通区域；进行开运算可使得车牌

区域与其它背景区域分开，成为独立的连通域。

(2) 计算包含所标记区域的最小宽和高。并根据先前知识，提取并显示更接近的车牌二值图。

(3) 通过计算车牌旋转角度解决车牌倾斜问题。由于车牌倾斜导致投影效果峰谷不明显，需车牌矫正处理，采取线性拟合方法，计算出车牌上边或下边图像值为 1 的点拟合直线与水平 X 轴的夹角。



图 7 车牌定位图像

3.3 字符分割

完成牌照区域的定位后，再将牌照区域分割为单个字符，可采用垂直投影法。由于字符在垂直方向上的投影必然在字符间或字符内的间隙处取得局部最小值，并且该位置应满足牌照的字符书写格式、字符、尺寸限制等条件。利用垂直投影法实现复杂环境下汽车图像中的字符分割效果较好。

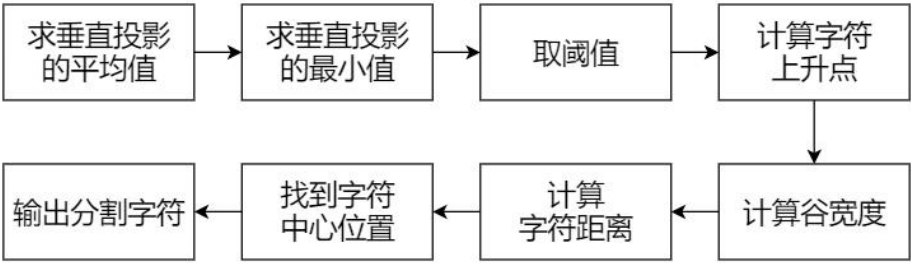


图 8 车牌分割的算法流程

通过分析计算字符的水平投影和垂直投影，可获得车牌字符高度、字符顶行与尾行、字符宽度、每个字符的中心位置，以方便提取分割字符。然后计算车牌垂直投影，去掉车牌垂直边框，获取车牌及字符平均宽度。最后计算车牌每个字符的中心位置和最大字符宽度，提取分割字符。



图 9 字符分割结果

3.4 字符识别

模板匹配算法是一种用于在图像中寻找特定模式或对象的计算机视觉技术。该算法的核心思想是将一个小的参考图像（称为模板）在待匹配图像中进行滑动，通过比较模板与图像局部区域的相似度来寻找匹配位置。通常，相似性度量采用的是像素值的比较或相关性系数。

算法的步骤包括选择合适的模板和设定相似性度量阈值。通过在图像上移动模板，计算每个位置处的相似性度量，并与设定的阈值进行比较，找到相似度高于阈值的位置即为匹配位置。

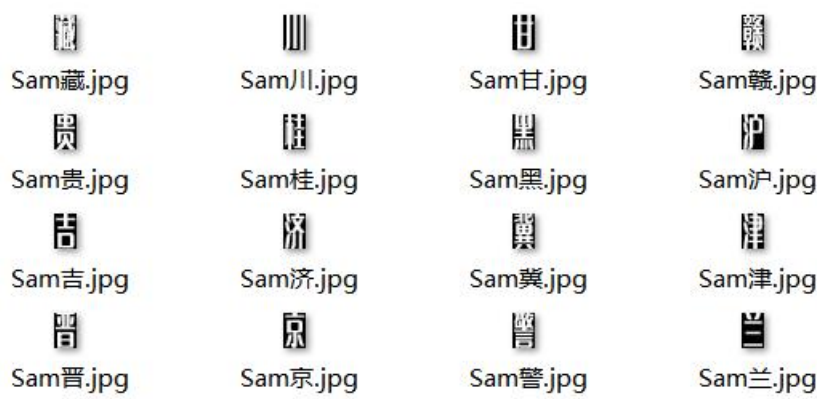


图 10 字符模板库

此处采用相减的方法来求得字符与模板中哪一个字符最相似，然后找到相似度最大的输出。汽车牌照的字符一般有七个，大部分车牌第一位是汉字，通常代表车辆所属省份，紧接其后的为字母与数字。车牌字符识别与一般文字识别在于它的字符数有限，汉字共约 30 多个，大写英文字母 26 个，数字 10 个。

首先取字符模板，接着依次取待识别字符与模板进行匹配，将其与模板字符相减，得到的非 0 值越少那么就越匹配。把每一幅相减后的图的非 0 值个数保存，即为识别出来的结果，准确率相对较高。如图 11 所示，字符识别结果与车牌相同，准确无误。



图 11 识别结果图

3.5 界面设计

MATLAB 的 GUI 界面设计如图 12 所示，按下“打开”按钮即可输入图片，按下“确认”按钮即可对图片中车辆的车牌进行识别。

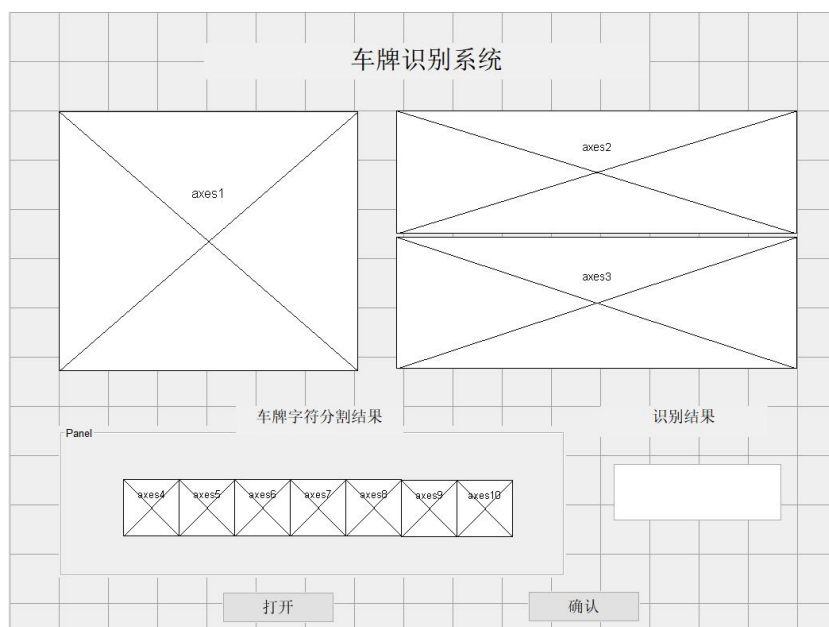


图 12 MATLAB 中 GUI 设计界面

输入示例图片，GUI 界面可以显示车牌定位图像、车牌区域二值化图像、车牌字符分割结果和识别结果：



图 3 车牌识别图像

四、总结与讨论

通过此次设计基于 MATLAB 的车牌识别系统，我对数字图像处理的基本知识：图像边缘检测、图像形态学滤波等都有了进一步的理解，学习到了适用 MATLAB 的 GUI 工具进行界面设计。在数字图像处理课程中，我们学习到了图像滤波、图像增强、图像几何变换、图像形态学处、图像分割等知识。而在实际的系统设计中，不同的方法适用于不同的应用场景，我们需要分析各种算法的适用场景，进行合适的阈值设置，根据处理的结果修改阈值或者算法，才能到达想要的效果。

本次车牌识别系统的设计较简单，基于传统的图像处理方法，实现图像分割和图像定位，当时车牌识别较为复杂，容易收到车辆运动和环境因素的影响，如模板匹配算法受到光照和视角影响较大，如果图像变形程度大，效果将大打折扣。近年来，人工智能技术发展迅速，使用深度学习进行车牌识别中的字符分割以及字符识别能够学习复杂的特征表示，从而提高对光照变化、视角变化、形变以及模糊等因素的鲁棒性，使系统更适用于不同场景，从而提升车牌识别的准确率。

最后，感谢上官老师的辛苦付出，让我学习到了数字图像处理的知识，了解了计算机视觉的应用，在未来的求学路上，我会将所学的知识用于实践之中。

附录

```
1. function varargout = chepai(varargin)
2.     gui_Singleton = 1;
3.     gui_State = struct('gui_Name',       mfilename, ...
4.                         'gui_Singleton', gui_Singleton, ...
5.                         'gui_OpeningFcn', @chepai_OpeningFcn, ...
6.                         'gui_OutputFcn', @chepai_OutputFcn, ...
7.                         'gui_LayoutFcn', [] , ...
8.                         'gui_Callback', []);
9.     if nargin && ischar(varargin{1})
10.         gui_State.gui_Callback = str2func(varargin{1});
11.     end
12.
13.     if nargout
14.         [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
15.     else
16.         gui_mainfcn(gui_State, varargin{:});
17.     end
18.
19. function chepai_OpeningFcn(hObject, eventdata, handles, varargin)
20.     handles.output = hObject;
21.     guidata(hObject, handles);
22.
23. function varargout = chepai_OutputFcn(hObject, eventdata, handles)
24.     varargout{1} = handles.output;
25.
26. function find_Callback(hObject, eventdata, handles)
27. %获取图像    装入待处理彩色图像并显示原始图像
28. global Scolor
29. %获取图像
30. [fn,pn,fi]=uigetfile('*.bmp','选择图片');
31. Scolor=imread([pn fn]);
32. axes(handles.axes1);
33. imshow(Scolor);
34.
35. function ok_Callback(hObject, eventdata, handles)
36. global Scolor
37. axes(handles.axes1);
38. imshow(Scolor),title('车牌原始图像');
39. %将彩色图像转换为黑白并显示
```

```

40. %% 图像灰度化
41. Sgray = rgb2gray(Scolor);%rgb2gray 转换成灰度图
42. %figure,imshow(Sgray),title('原始黑白图像');
43. %c=histeq(Sgray);
44. %figure,imshow(c);title('直方图均衡化图像');
45.
46. %% 图像增强
47. s=strel('disk',13);%strel 函数 13
48. Bgray=imopen(Sgray,s);%打开 sgray s 图像
49. %figure,imshow(Bgray);title('背景图像');%输出背景图像
50. Egray=imsubtract(Sgray,Bgray);%两幅图相减
51. %figure,imshow(Egray);title('增强黑白图像');%输出黑白图像
52.
53. %% 边缘提取
54. grd=edge(Egray,'canny',0.09,'both');
55. %figure,imshow(grd);title('robert 算子边缘检测');
56. se=[1;1;1]; %线型结构元素
57. I3=imerode(grd,se); %腐蚀图像
58. %figure,imshow(I3),title('腐蚀后边缘图像');
59. %% 开闭运算进行滤波
60. bg1=imclose(I3,strel('rectangle',[8,18]));%取矩形框的闭运算即平滑
    8,18
61. %figure,imshow(bg1);title('图像闭运算[5,19]');%输出闭运算的图像
62. bg3=imopen(bg1,strel('rectangle',[8,14]));%取矩形框的开运算 8,18
63. %figure,imshow(bg3);title('图像开运算[5,19]');%输出开运算的图像
64. bg2=bwareaopen(bg3,700);%去除聚团灰度值小于 700 的部分
65. %figure,imshow(bg2);title('从对象中移除小对象');
66. %% 车牌初步定位
67. [y,x,]=size(bg2);
68. I6=double(bg2);
69. %绘制行曲线图
70. Y1=zeros(y,1);%y 行 1 列的零矩阵 累计行像素灰度值
71. for i=1:y
72.     for j=1:x
73.         if(I6(i,j,1)==1)
74.             Y1(i,1)= Y1(i,1)+1;
75.         end
76.     end
77. end
78. % figure();
79. % subplot(1,3,1);
80. % plot(0:y-1,Y1),title('行像素灰度值累计'),xlabel('行值'),ylabel('
    像素和');
81. [temp, MaxY]=max(Y1);

```

```

82. PY1=MaxY;
83. while ((Y1(PY1,1)>=50)&&(PY1>1))
84.     PY1=PY1-7;
85. end
86. PY2=MaxY;
87. while ((Y1(PY2,1)>=50)&&(PY2<y))
88.     PY2=PY2+7;
89. end
90. %绘制列曲线图
91. X1=zeros(1,x);
92. for j=1:x
93.     for i=PY1:PY2
94.         if(I6(i,j,1)==1)
95.             X1(1,j)= X1(1,j)+1;
96.         end
97.     end
98. end
99. % subplot(1,3,2);
100. % plot(0:x-1,X1),title('列像素灰度值累计'),xlabel('列值'),ylabel('
    像数和');
101. PX1=1;
102. while ((X1(1,PX1)<3)&&(PX1<x))
103.     PX1=PX1+1;
104. end
105. PX2=x;
106. while ((X1(1,PX2)<3)&&(PX2>PX1))
107.     PX2=PX2-1;
108. end
109.
110. DW=Scolor(PY1:PY2,PX1:PX2,:);%车牌定位后图像
111. axes(handles.axes2);
112. imshow(DW),title('车牌定位图像');
113. % msgbox('识别部分思路: 连通域分割字符+模板匹配字符识别, 时间关系,
    Q2869939756')
114. %进一步车牌定位
115. num=size(DW);
116. if numel(num)>2
117.     I1 = rgb2gray(DW);    %将 RGB 图像转化为灰度图像
118. else    I1=DW;
119. end
120. g_max=double(max(max(I1)));
121. g_min=double(min(min(I1)));
122. T=round(g_max-(g_max-g_min)/3); % T 为二值化的阈值
123. [m,n]=size(I1);% d:二值图像

```

```

124. imane_bw=im2bw(I1,T/256); %二值化车牌图像
125. [y1,x1,z1]=size(imane_bw);
126. I3=double(imane_bw);
127. TT=1;
128. %去除图像顶端和底端的不感兴趣区域
129. Y1=zeros(y1,1);
130. for i=1:y1
131.     for j=1:x1
132.         if(I3(i,j,1)==1)
133.             Y1(i,1)= Y1(i,1)+1 ;
134.         end
135.     end
136. end
137. Py1=1;Py0=1;
138. while ((Y1(Py0,1)<9)&&(Py0<y1))
139.     Py0=Py0+1;
140. end
141. Py1=Py0;
142. while((Y1(Py1,1)>=9)&&(Py1<y1))
143.     Py1=Py1+1;
144. end
145. I2=imane_bw(Py0:Py1,:,:); %目标车牌区域
146. axes(handles.axes3);
147. imshow(I2),title('车牌区域二值化图像');
148. %% 车牌字符分割
149. % 寻找连续有文字的块，若长度大于某阈值，则认为该块有两个字符组成，需要分割
150. d=qiege(I2);
151. [m,n]=size(d);
152. k1=1;k2=1;s=sum(d);j=1;
153. while j~=n
154.     while s(j)==0
155.         j=j+1;
156.     end
157.     k1=j;
158.     while s(j)~=0 && j<=n-1
159.         j=j+1;
160.     end
161.     k2=j-1;
162.     if k2-k1>=round(n/6.5)
163.         [val,num]=min(sum(d(:,[k1+5:k2-5])));
164.         d(:,k1+num+5)=0; % 分割
165.     end
166. end

```

```

167. % 再切割
168. d=qiege(d);
169. % 切割出 7 个字符
170. y1=10;y2=0.25;flag=0;word1=[];
171. while flag==0
172.     [m,n]=size(d);
173.     left=1;wide=0;
174.     while sum(d(:,wide+1))~=0
175.         wide=wide+1;
176.     end
177.     if wide<y1 % 认为是左侧干扰
178.         d(:,[1:wide])=0;
179.         d=qiege(d);
180.     else
181.         temp=qiege(imcrop(d,[1 1 wide m]));
182.         [m,n]=size(temp);
183.         all=sum(sum(temp));
184.         two_thirds=sum(sum(temp([round(m/3):2*round(m/3)],:))
        ));
185.         if two_thirds/all>y2
186.             flag=1;word1=temp; % WORD 1
187.         end
188.         d(:,[1:wide])=0;d=qiege(d);
189.     end
190. end
191. [word2,d]=getword(d);% 分割出第二个字符
192. [word3,d]=getword(d); % 分割出第三个字符
193. [word4,d]=getword(d); % 分割出第四个字符
194. [word5,d]=getword(d); % 分割出第五个字符
195. [word6,d]=getword(d); % 分割出第六个字符
196. [word7,d]=getword(d); % 分割出第七个字符
197. [m,n]=size(word1);
198. %归一化大小为 22*14
199. word1=imresize(word1,[22 14]);
200. word2=imresize(word2,[22 14]);
201. word3=imresize(word3,[22 14]);
202. word4=imresize(word4,[22 14]);
203. word5=imresize(word5,[22 14]);
204. word6=imresize(word6,[22 14]);
205. word7=imresize(word7,[22 14]);
206. imwrite(word1,'1.jpg');
207. imwrite(word2,'2.jpg');
208. imwrite(word3,'3.jpg');
209. imwrite(word4,'4.jpg');

```



```

210. imwrite(word5,'5.jpg');
211. imwrite(word6,'6.jpg');
212. imwrite(word7,'7.jpg');
213. axes(handles.axes4);imshow(word1);title('1');
214. axes(handles.axes5);imshow(word2);title('2');
215. axes(handles.axes6);imshow(word3);title('3');
216. axes(handles.axes7);imshow(word4);title('4');
217. axes(handles.axes8);imshow(word5);title('5');
218. axes(handles.axes9);imshow(word6);title('6');
219. axes(handles.axes10);imshow(word7);title('7');
220. %% 车牌字符识别
221. liccode=char(['0':'9' 'A':'Z' '藏川甘赣贵桂黑沪吉济冀津晋京警兰辽领鲁
    蒙闽宁青琼陕使苏皖湘新学渝豫粤云浙']); %建立自动识别字符代码表
222. tt=1;
223. I=1;
224. for I=1:7
225.     ii=int2str(I);
226.     t=imread([ii,'.jpg']);
227.     t=255-t;
228.     level=graythresh(t);
229.     t=im2bw(t,level);
230.     SegBw2=imresize(t,[22 14],'nearest');
231.     if tt==1 %第一位汉字识别
232.         kmin=37;
233.         kmax=72;
234.         t=~t;
235.         SegBw2=imresize(t,[22 14],'nearest');
236.     elseif tt==2 %第二位 A~Z 字母识别
237.         kmin=11;
238.         kmax=36;
239.     else %第三位以后是字母或数字识别
240.         kmin=1;
241.         kmax=36;
242.     end
243.     for k2=kmin:kmax
244.         fname=strcat('code module\Sam',liccode(k2),'.jpg');
245.         SamBw22 = imread(fname);
246.         SamBw22 = rgb2gray(SamBw22);
247.         SamBw2=imresize(SamBw22,[22 14],'nearest');
248.         level=graythresh(SamBw22);
249.         SamBw2=im2bw(SamBw2,level);
250.         for i=1:22
251.             for j=1:14
252.                 SubBw2(i,j)=SegBw2(i,j)-SamBw2(i,j);

```

```

253.         end
254.     end
255.     % 以上相当于两幅图相减得到第三幅图
256.     Dmax=0;
257.     for k1=1:22
258.         for I1=1:14
259.             if ( SubBw2(k1,I1) > 0 || SubBw2(k1,I1) < 0 )
260.                 Dmax=Dmax+1;
261.             end
262.         end
263.     end
264.     Error(k2)=Dmax;
265. end
266. Error1=Error(kmin:kmax);
267. MinError=min(Error1);
268. findc=find(Error1==MinError);
269. if tt==1
270.     findc=findc+36;
271. end
272. if tt==2
273.     findc=findc+10;
274. end
275. tt=tt+1;
276. res=liccode(findc);
277. shibiejieguo(1,I)=res;
278. I=I+1;
279. end
280. set(handles.edit2,'string',shibiejieguo);
281.
282. function edit2_Callback(hObject, eventdata, handles)
283.
284. function edit2_CreateFcn(hObject, eventdata, handles)
285. if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'default
    UicontrolBackgroundColor'))
286.     set(hObject,'BackgroundColor','white');
287. end
288.
289. function stop_Callback(hObject, eventdata, handles)
290. ax1=(handles.axes1);cla(ax1,'reset')
291. ax2=(handles.axes2);cla(ax2,'reset')
292. ax3=(handles.axes3);cla(ax3,'reset')
293. ax4=(handles.axes4);cla(ax4,'reset')
294. ax5=(handles.axes5);cla(ax5,'reset')
295. ax6=(handles.axes6);cla(ax6,'reset')

```

```

296. ax7=(handles.axes7);cla(ax7,'reset')
297. ax8=(handles.axes8);cla(ax8,'reset')
298. ax9=(handles.axes9');cla(ax9,'reset')
299. ax10=(handles.axes10);cla(ax10,'reset')
300. set(handles.edit2,'string',' ');
301. msgbox('文档详细解析车牌识别原理')
302. function [ word, result ] = getword( img )
303. %UNTITLED 此处显示有关此函数的摘要
304. % 此处显示详细说明
305.     word = [];
306.     flag = 0;
307.     y1 = 8;
308.     y2 = 0.5;
309.
310.     while flag == 0
311.         [m, n] = size(img);
312.         width = 0;
313.         while sum(img(:, width+1)) ~= 0 && width <= n-2
314.             width = width + 1;
315.         end
316.         temp = my_imsplit(imcrop(img, [1,1,width,m]));
317.         [m1, n1] = size(temp);
318.         if width < y1 && n1/m1>y2
319.             img(:, [1, width]) = 0;
320.             if sum(sum(img)) ~= 0
321.                 img = my_imsplit(img);
322.             else
323.                 word = [];
324.                 flag = 1;
325.             end
326.         else
327.             word = my_imsplit(imcrop(img, [1, 1, width, m]));
328.             img(:, [1: width]) = 0;
329.             if sum(sum(img)) ~= 0
330.                 img = my_imsplit(img);
331.                 flag = 1;
332.             else
333.                 img = [];
334.             end
335.         end
336.     end
337.     result = img;
338. end
339. function [ split_img ] = my_imsplit( img )

```

```

340. %UNTITLED 此处显示有关此函数的摘要
341. % 此处显示详细说明
342. % 获取图像的大小
343. [m, n] = size(img);
344. top = 1;
345. bottom = m;
346. left = 1;
347. right = n;
348. % 获取字符的顶部位置
349. while sum(img(top, :)) == 0 && top <= m
350.     top = top + 1;
351. end
352. % 获取字符的底部位置
353. while sum(img(bottom, :)) == 0 && bottom >= 1
354.     bottom = bottom - 1;
355. end
356. % 获取字符的左边界
357. while sum(img(:, left)) == 0 && left <= n
358.     left = left + 1;
359. end
360. % 获取字符的右边界
361. while sum(img(:, right)) == 0 && right >= 1
362.     right = right - 1;
363. end
364. % 得到宽和高
365. width = right - left;
366. height = bottom - top;
367. % 切割图像
368. split_img = imcrop(img, [left top width height]);
369. end

```