

Saamliu

Matlab-LPR

This project is dedicated to the field of digital image processing. It focuses on the learning and research of license plate detection and recognition implemented using Matlab. The project features a graphical user interface (GUI) for intuitive interaction and enhanced usability.

Samliu

2023-1-2

ABSTRACT

License plate detection and recognition is a common problem and application in digital image processing field. The application of MATLAB in license plate detection and recognition is the purpose of this paper. MATLAB language code depends on the knowledge of digital image processing, such as grayscale processing, contour detection, image corrosion, image clipping and character recognition.

Keywords: digital image processing; MATLAB; Image segmentation; character recognition

目 录

ABSTRACT	1
一、引言	4
(一) 引言	4
(二) 涉及的知识与技术	4
二、环境说明	4
(一) 硬件环境	4
(二) 软件环境	4
(三) 使用的编程语言	4
三、正文	5
(一) 图像采集	5
(二) 图像预处理	5
1. 图像选择	5
2. 灰度处理	6
3. 边缘检测	7
(三) 车牌定位	8
1. 图像腐蚀	8
2. 图像平滑	9
3. 移除对象	10
4. 图像切割	11
(四) 车牌识别	13

1. 灰度处理	13
2. 直方图均衡化	13
3. 二值化处理	14
4. 中值滤波	15
5. 字符识别	15
四、总结语	23
五、参考文献	23

一、引言

（一）引言

matlab 自诞生以来，在数字图像处理领域发挥了重要作用。利用特色的工具箱以及各种专门用于图像处理的函数可以轻松的实现图像的处理。尤其是对于教育领域的教学工作者和想要了解数字图像处理技术的知识的人们带来了极大的便利和帮助。而车牌的检测与识别又一直是数字图像处理领域的典型应用。因此，本文将利用 matlab 这一强大的工具实现车牌的检测与识别……

（二）涉及的知识与技术

MATLAB 基本语法、数字图像处理的获取与基本运算、图像的边缘检测、图像腐蚀、图像的平滑、图像数学形态学、图像分割、图像的空域增强等等。

二、环境说明

（一）硬件环境

机型	Legion R7000
CPU	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
显卡	NVIDIA GeForce GTX 1650Ti
操作系统	Windows 10

表 1 硬件环境说明图

（二）软件环境

MATLAB 7.10.0 (R2010a)

（三）使用的编程语言

Matlab 语言

三、正文

车牌识别系统是一个复杂的系统，从采集图像到识别车牌号要经过一系列的硬件与软件。一般包括四个部分：车辆图像采集、车牌定位、车牌字符分割与车牌字符识别。简要流程图如图 1 所示。为了便于介绍，本文将通过一个随机案例将车牌识别的一系列流程加以体现。

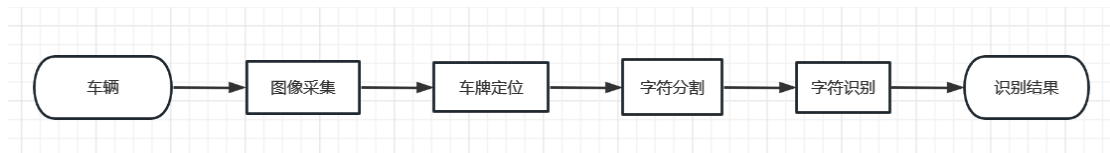


图 1 车牌识别系统流程图

（一）图像采集

获取车辆图像是车牌识别的第一步。由于图像采集主要依赖于拍摄系统，这里只给出文字描述。车辆图像通常由安装在道路上方或两侧的抓拍相机拍摄。因此，图像质量受天气条件和阳光的影响很大。或者车辆车牌本身就粘有泥巴，车牌已经损坏。目前，获取车辆图像的方法主要有两种：一种是彩色相机，另一种是数码相机。

（二）图像预处理

图像预处理是将图像采集得到的车辆图像（图像中包含车牌信息）进行包括灰度处理和边缘检测的一系列操作，为车牌的定位与识别提供良好的条件。

1. 图像选择

利用 `uigetfile` 函数打开文件选择对话框，指定文件类型为 `jpg` 文件和对话框标题为“选择需要识别的图像”。运用 `strcat` 将文件路径与文件名水平串联为 `file`。利用 `imread` 函数读取图像并创建图窗展示选择的图像。如图 2 所示。

```
[filename filepath] = uigetfile('.jpg', '选择需要识别的图像');  
file = strcat(filepath, filename);
```

```
img = imread(file);
figure;
imshow(img);
title('原图');
```



图 2 原图

2. 灰度处理

在图像预处理中，如果不需要处理颜色细节，那么灰度化就必不可少，这样可以加快后续处理的速度。首先，通过常用的 `rgb2gray` 函数将读取到的原图进行灰度化处理。再利用 `imhist` 得到灰度图像的直方图。创建图窗进行展示。如图 3 所示。

```
img1 = rgb2gray(img);    % RGB 图像转灰度图像
figure;
subplot(1, 2, 1);
imshow(img1);
title('灰度图像');
subplot(1, 2, 2);
imhist(img1);
title('灰度处理后的灰度直方图');
```

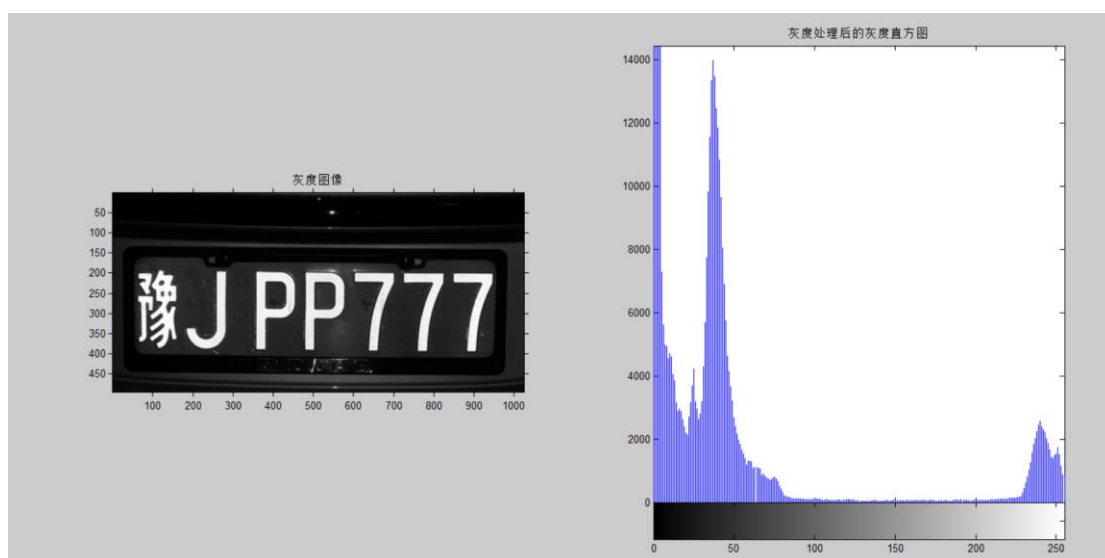


图 3 灰度图像与直方图

3. 边缘检测

在 matlab 中，利用 edge 函数很容易得到二维灰度图像的边缘。可以选择多种算法，例如 Sobel、Prewitt、Canny 等算法，如下表所示。

方法	描述
"Sobel"	使用导数的 Sobel 逼近，通过寻找图像 I 的梯度最大的那些点来查找边缘。（I 为输入图像）
"Prewitt"	使用导数的 Prewitt 逼近，通过寻找 I 的梯度最大的那些点来查找边缘。
"Roberts"	使用导数的 Roberts 逼近，通过寻找 I 的梯度最大的那些点来查找边缘。
"log"	使用高斯拉普拉斯（LoG）滤波器对 I 进行滤波后，通过寻找过零点来查找边缘。
"zerocross"	使用您指定的滤波器 h 对 I 进行滤波后，通过寻找过零点来查找边缘。
"Canny"	通过寻找 I 的梯度的局部最大值来查找边缘。edge 函数使用高斯滤波器的导数计算梯度。此方法使用双阈值来检测强边缘和弱边缘，如果弱边缘与强边缘连通，则将弱边缘包含

	到输出中。通过使用双阈值，Canny 方法相对其他方法不易受噪声干扰，更可能检测到真正的弱边缘。
"approxcanny"	使用近似版 Canny 边缘检测算法查找边缘，该算法的执行速度较快，但检测不太精确。浮点图像应归一化到范围 [0, 1]。

表 2 边缘检测方法

这里我选择 roberts 边缘检测算法。敏感度阈值为 0.15，检测方向为默认的水平垂直两个方向。创建图窗进行展示。如图 4 所示。

```
img4 = edge(img1, 'roberts', 0.15, 'both');
figure('name','边缘检测');
imshow(img4);
title('roberts 算子边缘检测');
```

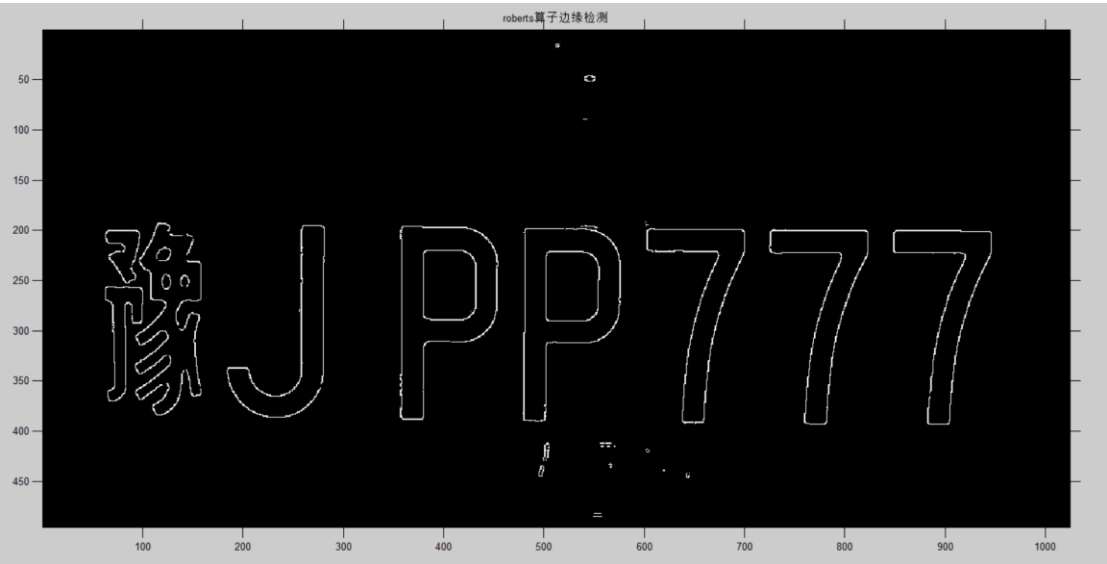


图 4 roberts 边缘检测

（三）车牌定位

车牌定位就是从图像预处理后得到的图像中找到车牌的位置，并将车牌单独切割出来。车牌定位包括图像腐蚀、图像平滑、移除对象与图像切割四个部分。车牌定位对后面字符的识别的准确率有着重要影响。

1. 图像腐蚀

对于腐蚀图像，matlab 中同样有专门的函数，imerode 函数。设置结构元

素 se，使用 se 腐蚀边缘检测得到的图像。最后得到腐蚀后的图像。创建图窗进行展示。如图 5 所示。

```
se=[1;1;1];  
img5 = imerode(img4, se);  
figure('name','图像腐蚀');  
imshow(img5);  
title('图像腐蚀后的图像');
```

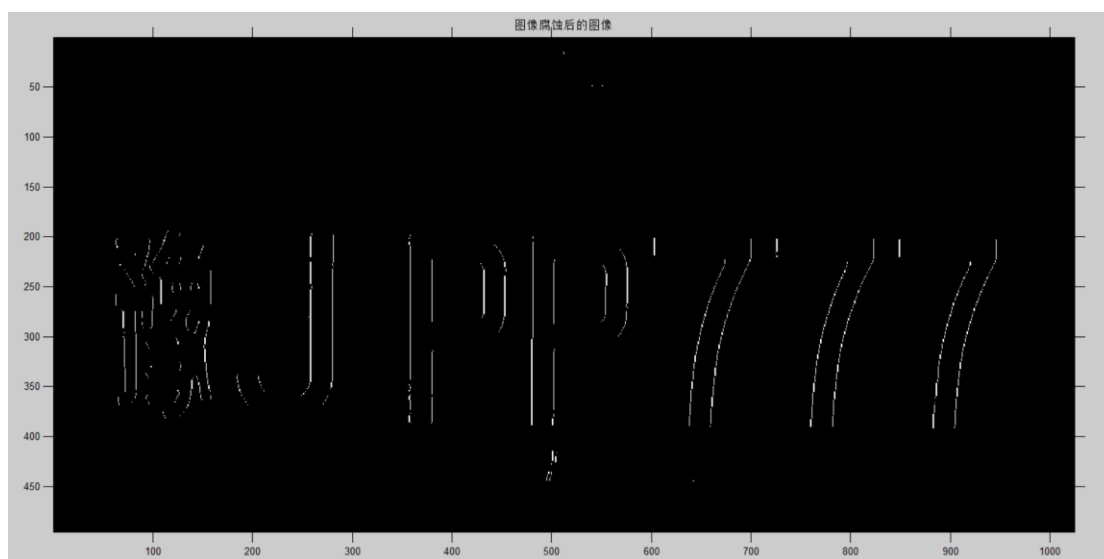


图 5 图像腐蚀后的图像

2. 图像平滑

运用 strel 函数创建一个大小为[30,30]的矩形结构元素 se。然后使用结构元素 se 对灰度或二值图像执行形态学闭运算。形态学闭运算是先进行膨胀后进行腐蚀，这两种运算使用相同的结构元素。最后创建图窗展示得到处理后的图像。如图 6 所示。

```
se = strel('rectangle', [30, 30]);  
img6 = imclose(img5, se);  
figure('name','平滑处理');  
imshow(img6);  
title('平滑图像的轮廓');
```

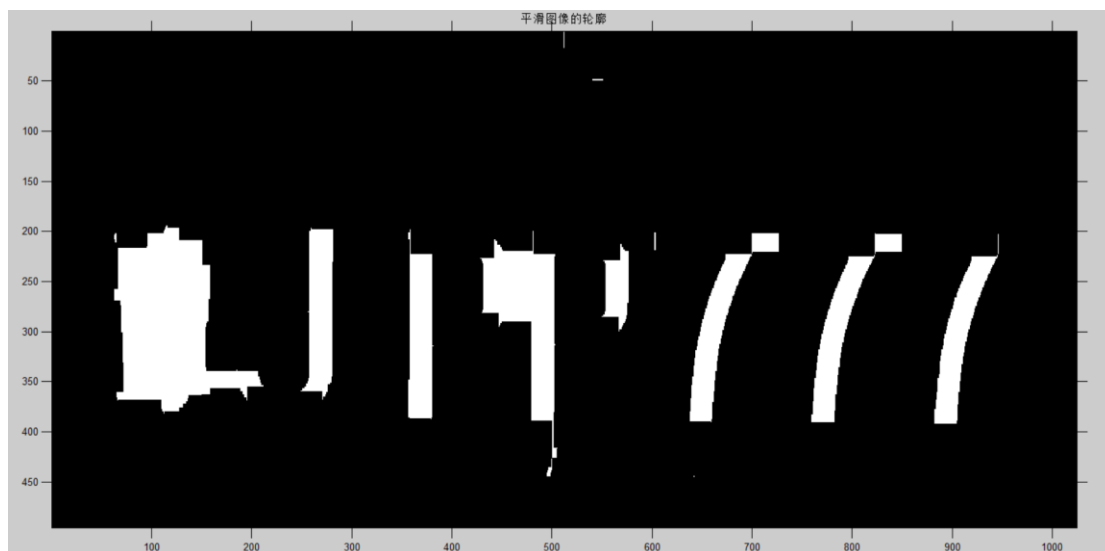


图 6 平滑图像的轮廓

3. 移除对象

使用 `bwareaopen` 函数删除二值图像 `img6` 中所有小于 2200 像素的连通分量，生成另一幅二值图像 `img7`。最后创建图窗展示得到处理后的图像。如图 7 所示。

```
img7 = bwareaopen(img6, 2200);
figure('name', '移除小对象');
imshow(img7);
title('从图像中移除小对象');
```

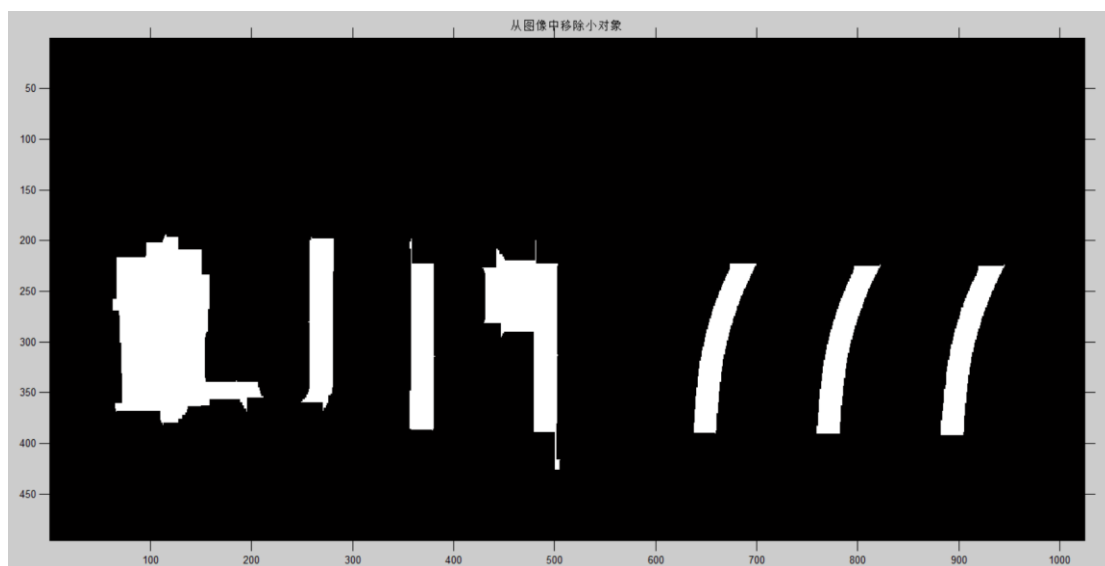


图 7 从图像中移除小对象

4. 图像切割

获取上一步得到的图像的大小，并将其转换为双精度浮点型。对于车牌的蓝色区域，在 y 方向上创建新的全零矩阵，利用双重 for 循环，逐像素判断车牌的位置区域。并统计像素点。如图 8 所示。

```
[y, x, z] = size(img7);  
img8 = double(img7);    % 转成双精度浮点型  
% 车牌的蓝色区域  
% Y 方向  
blue_Y = zeros(y, 1);  
for i = 1:y  
    for j = 1:x  
        if(img8(I, j) == 1)  
            blue_Y(i, 1) = blue_Y(i, 1) + 1;    % 像素点统计  
        end  
    end  
end
```

利用 while 循环找到 Y 坐标的最小值与最大值。

```
img_Y1 = 1;  
while (blue_Y(img_Y1) < 5) && (img_Y1 < y)  
    img_Y1 = img_Y1 + 1;  
end  
img_Y2 = y;  
while (blue_Y(img_Y2) < 5) && (img_Y2 > img_Y1)  
    img_Y2 = img_Y2 - 1;  
end
```

在 x 方向上同理。

```
% x 方向  
blue_X = zeros(1, x);
```

```

for j = 1:x
    for i = 1:y
        if(img8(i, j) == 1)
            blue_X(1, j) = blue_X(1, j) + 1;
        end
    end
end

% 找到 x 坐标的最小值
img_X1 = 1;
while (blue_X(1, img_X1) < 5) && (img_X1 < x)
    img_X1 = img_X1 + 1;
end

% 找到 x 坐标的最大值
img_X2 = x;
while (blue_X(1, img_X2) < 5) && (img_X2 > img_X1)
    img_X2 = img_X2 - 1;
end

最后对图像进行裁剪。

img9 = img(img_Y1:img_Y2, img_X1:img_X2, :);
figure('name', '定位剪切图像');
imshow(img9);
title('定位 裁剪后的 彩色车牌')

```

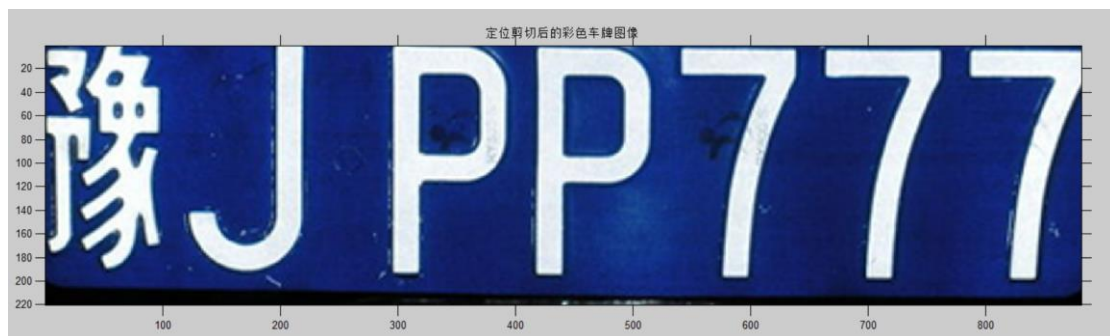


图 8 定位剪切后的彩色车牌图像

（四）车牌识别

车牌识别就是将切割后得到的多个车牌部分进行一系列处理，最后进行字符识别。车牌识别包括灰度处理、直方图均衡化、二值化、中值滤波与字符识别五个部分。

1. 灰度处理

通过常用的 `rgb2gray` 函数将读取到的原图进行灰度化处理。再利用 `imhist` 得到灰度图像的直方图。创建图窗进行展示。如图 9 所示。

```
plate_img1 = rgb2gray(img9);    % RGB 图像转灰度图像
figure;
subplot(1, 2, 1);
imshow(plate_img1);
title('灰度图像');
subplot(1, 2, 2);
imhist(plate_img1);
title('灰度处理后的灰度直方图');
```

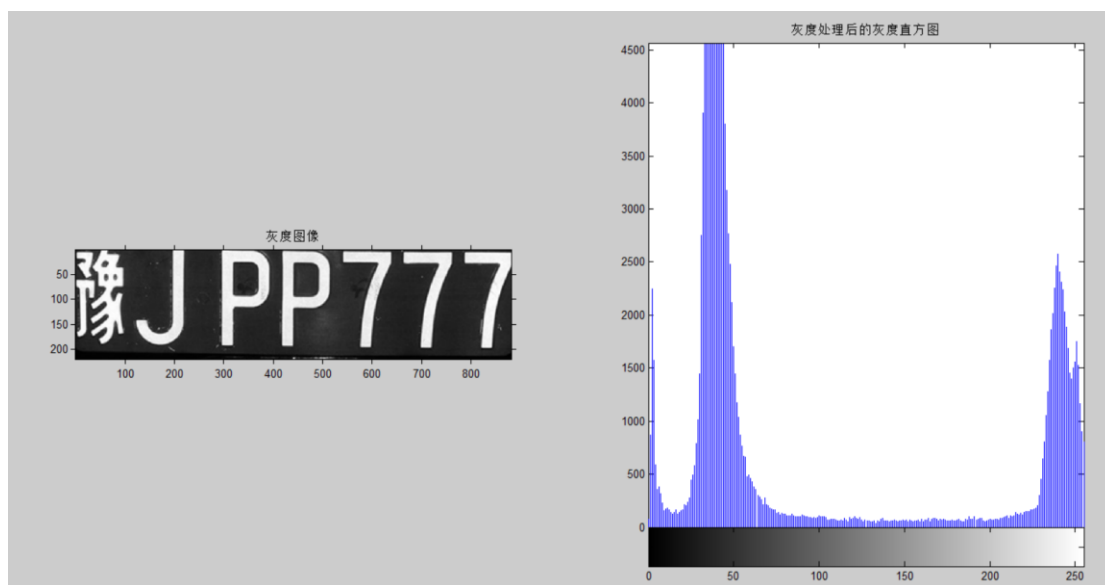


图 9 车牌得到灰度图及其直方图

2. 直方图均衡化

利用 `histeq` 函数使用直方图均衡增强对比度。再利用 `imhist` 得到灰度图

像的直方图。创建图窗进行展示。如图 10 所示。

```
plate_img2 = histeq(plate_img1);  
figure('name', '直方图均衡化');  
subplot(1,2,1);  
imshow(plate_img2);  
title('直方图均衡化的图像');  
subplot(1,2,2);  
imhist(plate_img2);  
title('直方图');
```

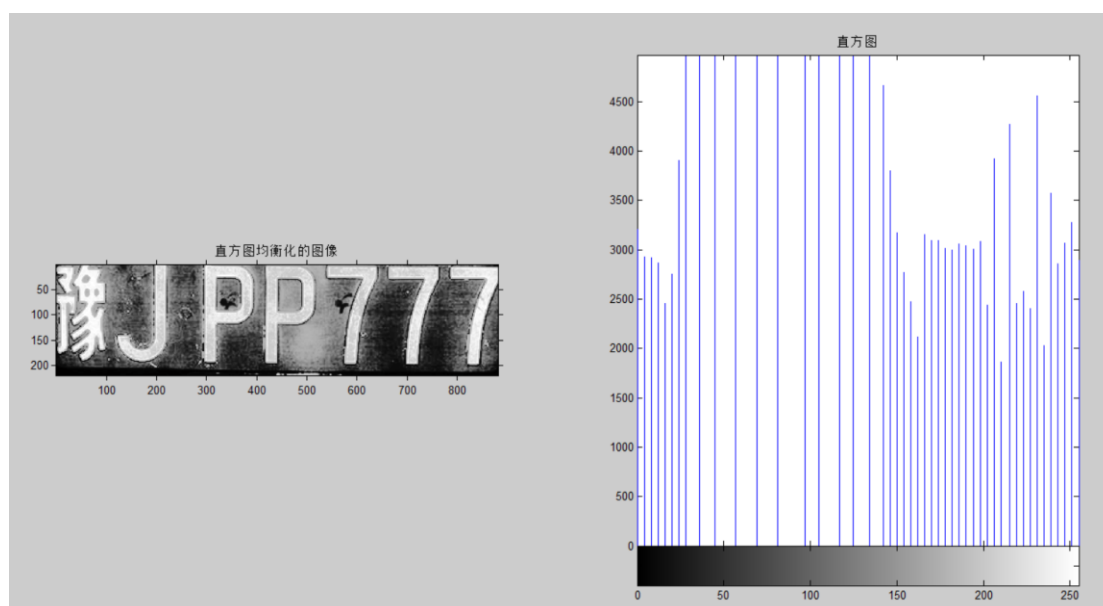


图 10 车牌的直方图均衡化图像及其直方图

3. 二值化处理

使用 `im2bw` 函数，将灰度图像 `plate_img2` 转换为二值图像 `plate_img3`。方法是将输入图像中所有亮度大于 0.76 的像素替换为值 1 (白色)，其他所有像素替换为值 0 (黑色)。该范围与图像类别的可能信号电平相关。因此，级别值 0.5 对应于该类别的最小值和最大值之间的中间强度值。最后，创建一个用于显示的绘图窗口。如图 11 所示。

```
plate_img3 = im2bw(plate_img2, 0.76);  
figure('name', '二值化处理');
```

```
imshow(plate_img3);  
title('车牌二值图像');
```

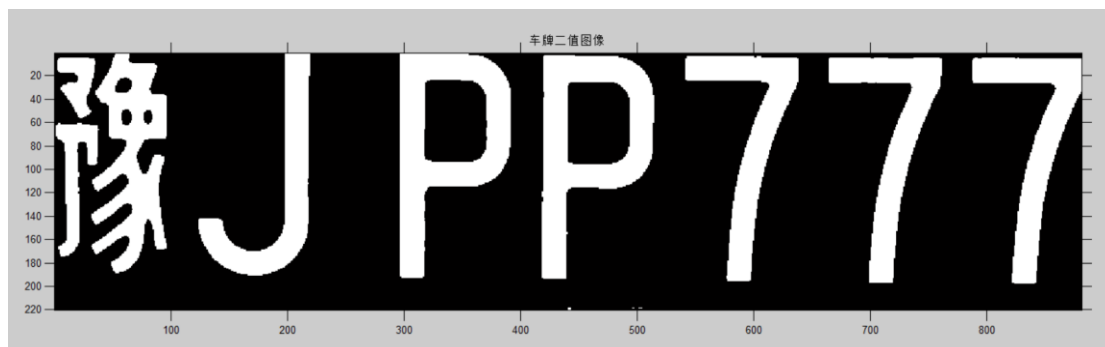


图 11 车牌二值图像

4. 中值滤波

利用 `medfilt2` 函数对图像 `plate_img3` 执行二维中位数滤波。每个输出像素包含输入图像中对应像素周围 3×3 邻域的中位数值。如图 12 所示。

```
plate_img4 = medfilt2(plate_img3);  
figure('name', '中值滤波');  
imshow(plate_img4);  
title('中值滤波后的图像');
```

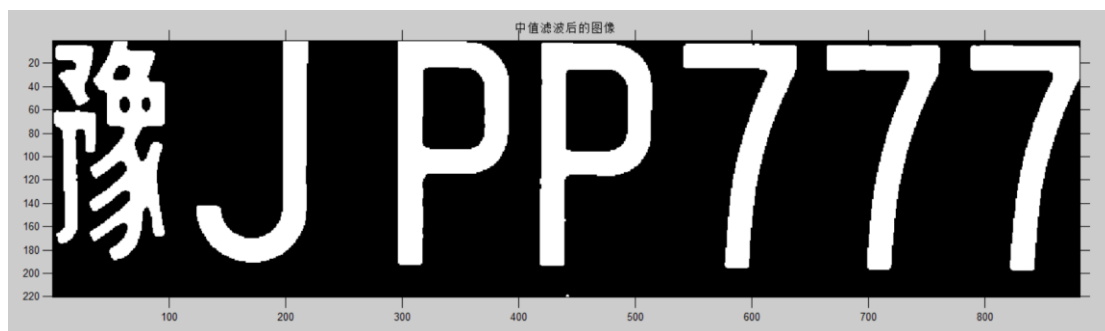


图 12 中值滤波后的图像

5. 字符识别

这里，减法用于寻找与模板中的字符具有最高相似度的字符。汽车的牌照一般有七个字，第 1 个汉字通常代表车辆的省份，后面是字母和数字。车牌字符识别和通用字符识别在字符数量上有限制，包括 50 多个汉字、26 个大写英文字母和 10 个数字。因此，建立字符模板库是一种非常方便的选择。为了减轻项目代

码的复杂程度，根据本次项目选择的车牌的特征点，仅仅建立了七个汉字、26 个英文字母和 10 个数字的模板。如图 13、14、15 所示。

%% 进行字符识别

```
plate_img5 = my_imsplit(plate_img4);
```

```
[m, n] = size(plate_img5);
```

```
s = sum(plate_img5);
```

```
j = 1;
```

```
k1 = 1;
```

```
k2 = 1;
```

```
while j ~= n
```

```
    while s(j) == 0
```

```
        j = j + 1;
```

```
    end
```

```
    k1 = j;
```

```
    while s(j) ~= 0 && j <= n-1
```

```
        j = j + 1;
```

```
    end
```

```
    k2 = j + 1;
```

```
    if k2 - k1 > round(n / 6.5)
```

```
        [val, num] = min(sum(plate_img5(:, [k1+5:k2-5])));
```

```
        plate_img5(:, k1+num+5) = 0;
```

```
    end
```

```
end
```

```
y1 = 10;
```

```
y2 = 0.25;
```

```
flag = 0;
```

```

word1 = [];
while flag == 0
    [m, n] = size(plate_img5);
    left = 1;
    width = 0;
    while sum(plate_img5(:, width+1)) ~= 0
        width = width + 1;
    end
    if width < y1
        plate_img5(:, [1:width]) = 0;
        plate_img5 = my_imsplit(plate_img5);
    else
        temp = my_imsplit(imcrop(plate_img5, [1,1,width,m]));
        [m, n] = size(temp);
        all = sum(sum(temp));
        two_thirds==sum( sum(temp [roud(m/3): 2*roud(m/3)],:));
        if two_thirds/all > y2
            flag = 1;
            word1 = temp;
        end
        plate_img5(:, [1:width]) = 0;
        plate_img5 = my_imsplit(plate_img5);
    end
end

figure;
subplot(2,4,1), imshow(plate_img5);

```

```

% 分割出第二个字符
[word2,plate_img5]=getword(plate_img5);
subplot(2,4,2), imshow(plate_img5);
% 分割出第三个字符
[word3,plate_img5]=getword(plate_img5);
subplot(2,4,3), imshow(plate_img5);
% 分割出第四个字符
[word4,plate_img5]=getword(plate_img5);
subplot(2,4,4), imshow(plate_img5);
% 分割出第五个字符
[word5,plate_img5]=getword(plate_img5);
subplot(2,3,4), imshow(plate_img5);
% 分割出第六个字符
[word6,plate_img5]=getword(plate_img5);
subplot(2,3,5), imshow(plate_img5);
% 分割出第七个字符
[word7,plate_img5]=getword(plate_img5);
subplot(2,3,6), imshow(plate_img5);

figure;
subplot(5,7,1),imshow(word1),title('1');
subplot(5,7,2),imshow(word2),title('2');
subplot(5,7,3),imshow(word3),title('3');
subplot(5,7,4),imshow(word4),title('4');
subplot(5,7,5),imshow(word5),title('5');
subplot(5,7,6),imshow(word6),title('6');
subplot(5,7,7),imshow(word7),title('7');

```

```

word1=imresize(word1,[40 20]);
word2=imresize(word2,[40 20]);
word3=imresize(word3,[40 20]);
word4=imresize(word4,[40 20]);
word5=imresize(word5,[40 20]);
word6=imresize(word6,[40 20]);
word7=imresize(word7,[40 20]);

subplot(5,7,15),imshow(word1),title('11');
subplot(5,7,16),imshow(word2),title('22');
subplot(5,7,17),imshow(word3),title('33');
subplot(5,7,18),imshow(word4),title('44');
subplot(5,7,19),imshow(word5),title('55');
subplot(5,7,20),imshow(word6),title('66');
subplot(5,7,21),imshow(word7),title('77');

imwrite(word1,'1.jpg');
imwrite(word2,'2.jpg');
imwrite(word3,'3.jpg');
imwrite(word4,'4.jpg');
imwrite(word5,'5.jpg');
imwrite(word6,'6.jpg');
imwrite(word7,'7.jpg');

%% 进行字符识别
liccode=char(['0':'9' 'A':'Z' '京辽鲁陕苏豫浙贵']);
subBw2 = zeros(40, 20);
num = 1; % 车牌位数

```

```

for i = 1:7
    ii = int2str(i);
    word = imread([ii, '.jpg']);
    segBw2 = imresize(word, [40,20], 'nearest');
    segBw2 = im2bw(segBw2, 0.5);

    kMin = 37;
    kMax = 44;
elseif i == 2
    kMin = 11;
    kMax = 36;
elseif i >= 3
    kMin = 1;
    kMax = 36;
end

l = 1;
for k = kMin : kMax
    fname = strcat('字符模板\', liccode(k), '.jpg');
    samBw2 = imread(fname);
    samBw2 = im2bw(samBw2, 0.5);    % 图像二值化

    for i1 = 1:40
        for j1 = 1:20
            subBw2(i1, j1) = segBw2(i1, j1) -
samBw2(i1 ,j1);
        end
    end
end

```

```

        Dmax = 0;
        for i2 = 1:40
            for j2 = 1:20
                if subBw2(i2, j2) ~= 0
                    Dmax = Dmax + 1;
                end
            end
        end
        error(1) = Dmax;
        l = l + 1;
    end

    errorMin = min(error);
    findc = find(error == errorMin);
%     error
%     findc

    Code(num*2 - 1) = liccode(findc(1) + kMin - 1);
    Code(num*2) = ' ';
    num = num + 1;

end
% 显示识别结果
disp(Code);
msgbox(Code, '识别出的车牌号');

```

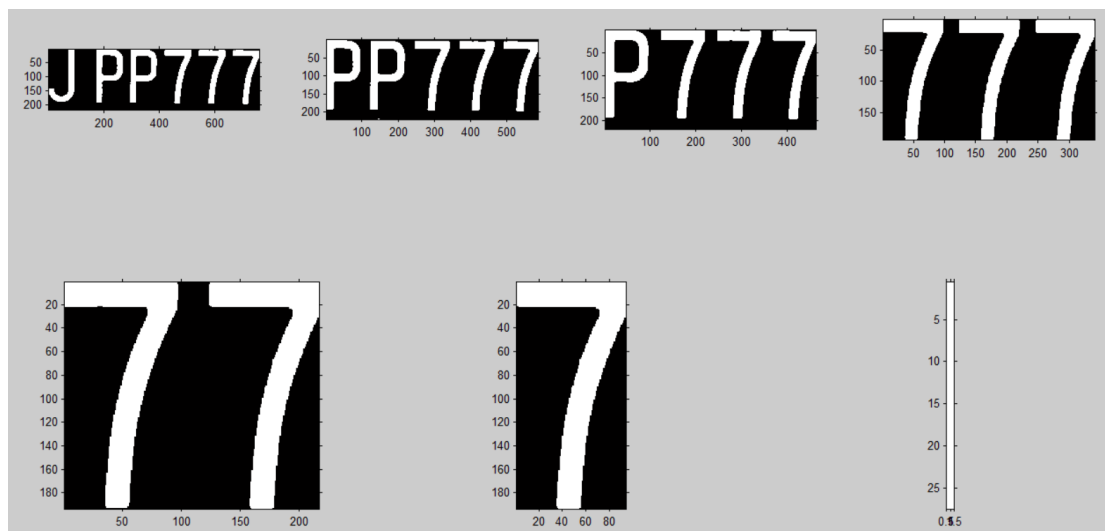


图 13 车牌的字符分割

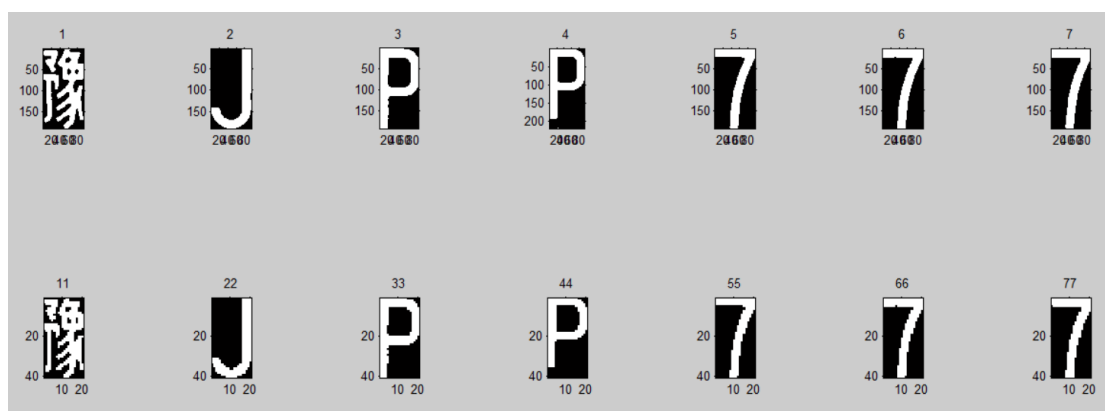


图 14 车牌的分割后的字符

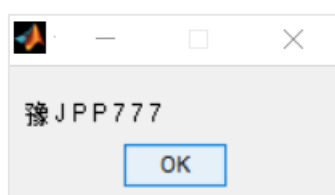


图 15 识别出的车牌号

四、总结语

基于 MATLAB 的车牌检测与识别是对数字图像处理技术的一次综合应用。车牌检测与识别技术是一项值得推广与研究的技术，不管是在交通管理还是刑侦追查方面都有强大的作用。本文通过应用数字图像处理技术的相关知识，例如灰度处理、边缘检测、图像腐蚀、图像平滑、图像切割与字符识别，得到了 matlab 语言的代码实现，也是对 matlab 的一种应用和车牌检测与识别的一次探索。

五、参考文献

- [1]何健东. 数字图像处理（第三版）[M]. 西安：西安电子科技大学出版社，2015：12-240.
- [2]赵云龙. 智能图像处理：Python 和 OpenCV 实现[M]. 北京：机械工业出版社，2022:15-317.
- [3]Rafael C G, Richard E W. 数字图像处理（第 3 版）[M]. 阮秋琦，阮宇智，等译. 北京：电子工业出版社，2013.
- [4]陈天华. 数字图像处理及应用——使用 MATLAB 分析与实现[M]. 北京：清华大学出版社，2019.
- [5] Documentation - MATLAB & Simulink - MathWorks 中国. Ww2.
<https://ww2.mathworks.cn/help/index.html>
- [6]License Plate Recognition System | IEEE Conference Publication | IEEE Xplore. Ieeexplore.
<https://ieeexplore.ieee.org/document/9404706/references>
- [7] License Plate Recognition Systems - EPIC - Electronic Privacy Information Center. Epic. <https://epic.org/license-plate-recognition-systems/>