**数字图像处理技术大作业报告**

**《车牌图像字符分割》**

**学 院（系）：计算机科学与工程学院**

**专 业 ： 计算机科学与技术**

**指导老师： 崔少国**

**学生姓名: 陈松 学号:11703990535**

**时间：从2019年11月18日到2019年12月8日**

目录

[1 课题任务及要求 3](#_Toc26736886)

[2 核心且公共的部分函数 4](#_Toc26736887)

[2.1 前提 4](#_Toc26736888)

[2.2 移除上下边框 4](#_Toc26736889)

[2.3 去除左右边框 5](#_Toc26736890)

[2.4 删除左右空白 7](#_Toc26736891)

[2.5 自动切割图像 8](#_Toc26736892)

[3字符切割 9](#_Toc26736893)

[3.1 I 类图 9](#_Toc26736894)

[3.1.1 军A00101 10](#_Toc26736895)

[3.1.2 川A9999 10](#_Toc26736896)

[3.1.3 鲁JD9309 10](#_Toc26736897)

[3.1.4 KW507 10](#_Toc26736898)

[3.1.5 京V02633 11](#_Toc26736899)

[3.1.6 京V202813 11](#_Toc26736900)

[3.2 II 类图 11](#_Toc26736901)

[3.2.1 蒙 A16777 11](#_Toc26736902)

[3.2.2 兰A20008 12](#_Toc26736903)

[3.2.3 翼BQQ613 12](#_Toc26736904)

[3.2.4 晋F21696 12](#_Toc26736905)

[3.3 III类图 13](#_Toc26736906)

[3.3.1 定位 13](#_Toc26736907)

[3.3.2 切割 14](#_Toc26736908)

[4 实验心得 15](#_Toc26736909)

[5 参考文献 16](#_Toc26736910)

# 课题任务及要求

1.成绩为中者至少完成II类文件夹中3幅图像的分割；

2.成绩为良者至少完成II类文件夹中4幅图像的分割；

3.成绩为优者除了完成II类文件夹中4幅图像的分割外，还需完成III类文件夹中1幅图像的分割；

4.分割效果评价：

优：各字符完整，字符彼此及字符与其它分割物无粘连，车牌区域除字符外其它分割物少或者无。

良：各字符完整，字符间有少许粘连，但不影响辨识。

中：字符不太完整，或者车牌区域有较多非字符分割物，或者字符粘连较多。

差：字符不完整或缺字符，字符粘连较多，无法完整辨识车牌。

5.文档内容包括：（1）课题任务及要求；（2）每幅图像处理步骤或方法，MATLAB代码，分割效果截图等；（3）心得体会

6.文档要求：内容详实，能够体现已掌握的图像处理技术（图像类型转换、图像几何、灰度变换、直方图变换、图像增强、图像分割等）；格式规范，格式不符者不能评优和良。

# 2 核心且公共的部分函数

## 2.1 前提

**跳跃数:对于二值图像，求前后两列之差，即当差值等于一时可认为此处是在水平方向跳跃点，即字符得边界。**

**闭运算：闭运算是先膨胀后腐蚀的过程，用来填充物体内细小空洞、平滑边界，连通车牌区域，以便于取轮廓；**

**开运算：开运算是先腐蚀后膨胀的过程，用来消除细小的噪声、平滑较大物体的边界；**

**结构元素形状选取：车牌区域的形状是长方形的（也可以使用水平和垂直直线），所以选取方形的结构元素进行形态学操作；**

**结构元素大小选取：用字符的标准宽度、高度、车牌的标准宽度、标准高度、图片测量的车牌宽度和高度之间的关系，来大概确定结构元素的宽和长，再经过多次调试进一步确定结构元素宽和长。**

## 2.2 移除上下边框

function out = remove\_sx\_border(I)

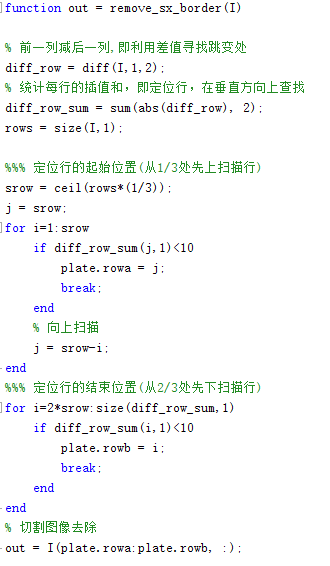




图1 带有柳丁的二值图像



图2 去除上下边框的二值图像

输入:灰度化后的图片

输出: 去除上下边框柳丁

**原理:对于已定位好的灰度图片,考虑的仅仅是车牌及其周边的部分，首先需要知道一个概念，在对前后两列之差的统计,可以在水平方向上检查到跳跃点，同时在水平方向上统计跳跃点数，即计算每行得差值和，可定位柳钉所在行，经观察所得铆钉所在行有6次像素跳变（2个铆钉4次+左右边框2次），字符所在行至少16次跳变（7个字符至少14次+左右边框2次），所以行的1/3处往上扫描，在行的2/3处往下扫描，设定一定的跳变阈值（10），来确定字符的起始行和结束行,如图1所示在水平反向上下柳丁均有4次像素跳变。**

## 2.3 去除左右边框

function out=remove\_zy\_border(I)

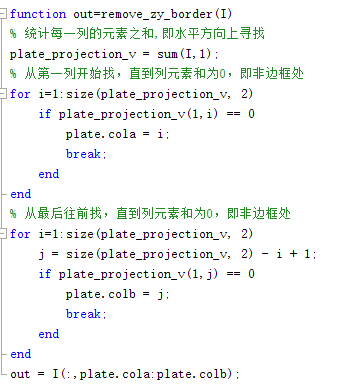




图3 带边框得二值图像



图 4 去除左右边框得二值图像

**输入: 灰度化后的图片**

**输出: 去除左右边框柳丁**

**原理:如图3所示二值图像得左右两边是一片白的‘边框’，为去除这些边框，可利用左右边框和字符间的空白部分(即一列元素全为0时)，可在水平方向上，对每一列的元素和进行统计，然后从左右两边开始寻找便可定位找到空闲的边界，即如图4所示，可定位列，删除左右边框。**

## 2.4 删除左右空白

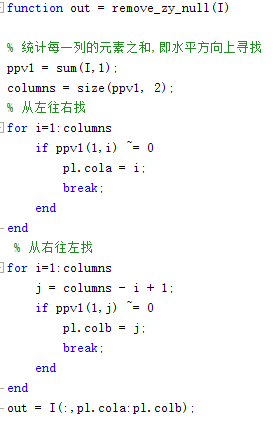


图5 左右两边空白部分较多



图6 去除空白

**输入: 灰度化后的图片**

**输出: 去除左右边框柳丁**

**原理:如图3所示二值图像得左右两边有大量空白部分，为去除这些空白，可对垂直方向上进行线检测，即从左右两边寻找，列元素和不为0的点(定位字符边界)，即可得到图六所示无上下边框，且无空白的二值图，方便最后进行自动切割。**

## 2.5 自动切割图像



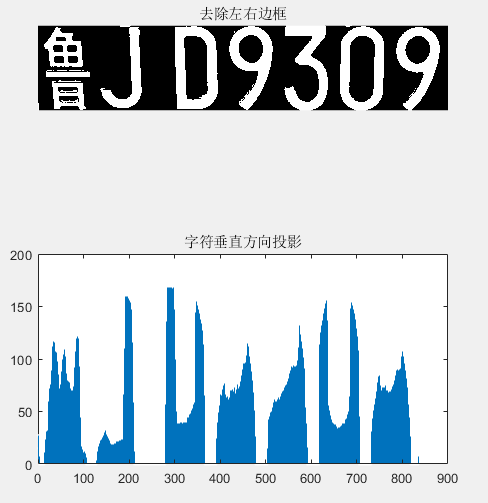


图7 投影图



图8 分割图

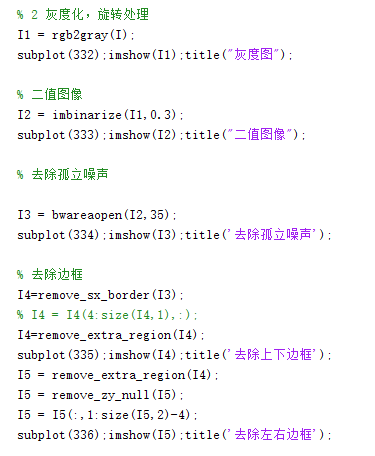
**输入:已去除上下边框，空白的二值图**

**输出:切割好的字符**

**原理:基于规律，如图7投影图所示，每个字符所占比例一致，且除2,3字符外间隔一致，即只要针对特殊的2,3字符外的间隔，其余字符的切割可按比例自动定位。**

# 3字符切割

## 3.1 I 类图



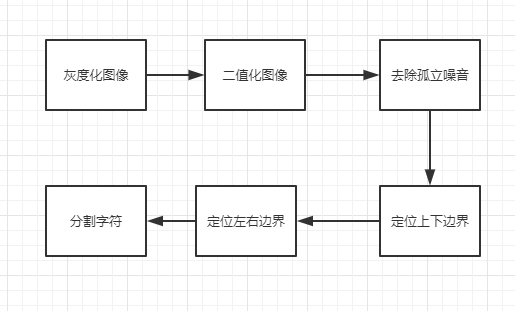


图 9 流程图



图 10 流程展示

**如图9、图10所示，I类图一般按此流程即可得出结果。**

### 3.1.1 军A00101



图 11军A00101

### 3.1.2 川A9999



图 12 川A9999

### 3.1.3 鲁JD9309



图13鲁JD9309

### 3.1.4 KW507



图14 KW507

### 3.1.5 京V02633



图15 京V02633

转为灰度化图像前，imrotate(I1,-10);即顺时针旋转了10度。

### 3.1.6 京V202813



图16 京V202813

转为灰度化图像前，imrotate(I1, -5);即顺时针旋转了5度。

## 3.2 II 类图

**分割步骤如I类图的类似，不过对需要对部分图片进行调整**

### 3.2.1 蒙 A16777



图17 未反转的灰度的二值化图



图 18 蒙 A16777

**处理:二值化前，需对图像反转，I1 = 255 -I1;因为该图片灰度化后，字符部分处于低灰度值部分，如图17所示，不利于后面步骤的边界定位。**

### 3.2.2 兰A20008

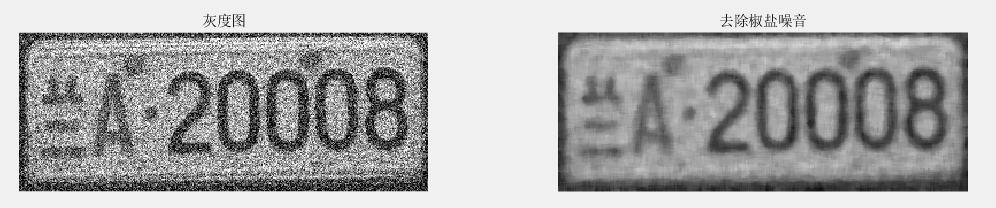


图19 中值滤波



图20 兰A20008

**处理 ： 如图19所示**I1 = medfilt2(f,[8,8], 'symmetric');**中值滤波去除椒盐噪音**

### 3.2.3 翼BQQ613



图21 中值滤波



图22 翼BQQ613

**处理 ： 如图21所示**I1 = medfilt2(f,[2,2],'symmetric');**中值滤波去除椒盐噪音**

### 3.2.4 晋F21696

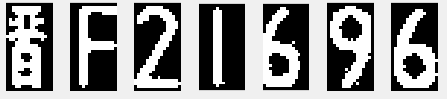


图 23 晋F21696

**处理：**I2 = imbinarize(I1,0.5);**即修改阈值K=0.5即可**

## 3.3 III类图

**对于第III类图片，车牌信息只占是图片的一小部分，则第一步做的是定位图片中的车牌，只要做到这一致，后面的步骤便和I、II类一致。**

### 3.3.1 定位



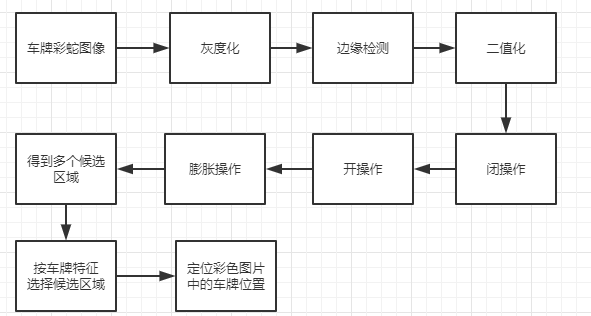


图 24 定位流程图



图 25 定位步骤图

**处理：按图24 流程图所示步骤执行，即可得图25所示的步骤图，从最终膨胀结果可见车牌区域已经大致定位到了。**

### 3.3.2 切割

当车牌位置定位好后，此后车牌的处理和I,II类便差不多了。



图 26 总流程步骤图



图 27 奥A3Y347

**定位后，将车牌分割，然后按I、II、类步骤分割字符即可，如图26所示为分割字符的图示步骤，最后从图26的字符垂直方向投影，图27字符分割图所示，效果不错。**

# 4 实验心得

**这学期以来，我对图像的感觉可谓是跌宕起伏。**

还没有接触这门课的时候，觉得图像处理***高高在上***，肯定非常***难以让人理解***。

在上了理论课的时候，一来就微分积分的，也让我很懵，但随着老师的讲解和平时的练习，觉得其运用的图像上也就简简单单的加减法，瞬间觉得图像处理也***不过如此***，***甚至大作业也让我跃跃欲试***。

但事情总不会这么简单，在设计最简单的第一类问题就遇到了一个难题，***车牌照图像怎么处理才算处理好？*** 百思不得其解后，就常规百度，在阅读了几篇博客后，处理好的图片是最后自动分割效果最好的。这时我扔出几种线检测就解决了此问题。

这时第三类题来了，不是简单的%90左右像素都是车牌信息，而是有只有一小部分是。同样，我又懵了，又是常规百度，又是简单的几次阅读后，发现其实也很简单，一套拳法(边缘检测，膨胀收缩，车牌规律【方形、蓝色像素】)就可以做到定位。

**总结：不要总是把图像处理想的太复杂，这个世界是非常有规律的，理解并运用图像间的规律，便能对图像‘为所欲为’了**。

# 5 参考文献

[1] 基于Matlab的车牌识别系统

<https://blog.csdn.net/zhangquan2015/article/details/78994551>

[2] 基于matlab的蓝色车牌识别（车牌字符分割）

<https://blog.csdn.net/Joseph__Lagrange/article/details/95724053>

[3] 基于matlab的蓝色车牌识别（车牌定位）

<https://blog.csdn.net/Joseph__Lagrange/article/details/95723974>

[4] 基于matlab的蓝色车牌识别（车牌倾斜矫正）

<https://blog.csdn.net/Joseph__Lagrange/article/details/96099117>

[5] 基于matlab的蓝色车牌识别（绪论）

<https://blog.csdn.net/Joseph__Lagrange/article/details/95757863>

[6] 基于matlab的蓝色车牌识别（车牌字符分割）

<https://blog.csdn.net/Joseph__Lagrange/article/details/95724053>