

## 车牌识别:

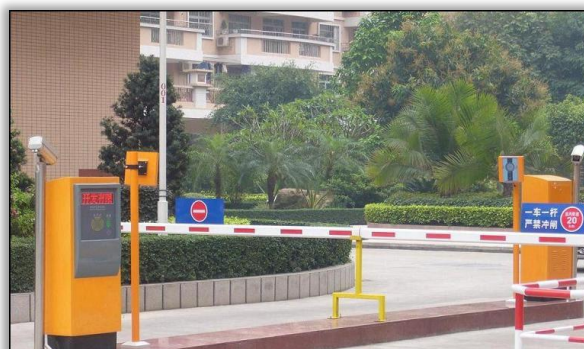
车牌识别是从一张或一系列数字图片中自动定位车牌区域并提取车牌信息的图像识别技术，它以数字图像处理、模式识别、计算机视觉等技术为基础，是现代智能交通系统的重要组成部分，广泛应用于日常生活中。



公路收费站收费



道路车辆违规行为监管



车辆出入门禁管理

# 主要研究成果:

## 1 建立图片集

以1485张图片组成的原始数据集为出发点，制作了用于分类器训练和测试的图片集。

## 2 提取车牌区域候选人

针对车牌区域特点和输入图片的全局特点，提出了四种不同的可能的车牌区域提取方法，经过校正后得到车牌区域候选人。

## 3 字符分析

参考自然场景下的文本检测方法和车牌中的字符分布特点，对车牌候选区域提取改进的最大稳定极值区域，通过使用非极大值抑制和区域校正得到字符候选人，通过字符判断分类器和字符搜索，实现非车牌区域的滤除和车牌区域7个字符的提取，其中字符判断使用基于20个描述性特征的支持向量机实现。

## 4 字符识别

通过使用金字塔梯度方向直方图特征得到了用于数字和字母识别的三层BP神经网络模型，定义了10层的卷积神经网络模型MyLeNet实现汉字的识别。

## 5 程序综合

搭建和编写了一个完整的工程项目，该项目整合了研究过程中的所有方法的程序实现，可以对数据集进行操作也可实现单张图片的自动识别，

## 输入图片的限制条件：

- 第一：处理的车牌为440mm\*140mm的蓝底白字的机动车车牌；
- 第二：输入图片最大边长为1600个像素点，整个车牌区域占整张图片中的像素点数目不得超过1/8；
- 第三：车牌区域在原图片中肉眼可以清晰分辨，没有严重的模糊现象，车牌中的所有字符没有缺损和遮挡；
- 第四：车牌区域像素点数在1000到10万之间，车牌的倾斜角度不超过 $\pm 45^\circ$ ，图中车牌的长宽比在1.3到6.0之间；
- 第五：图片没有较大的水印和人为的涂画，天气状况均为晴天。

## 限制存在的主要原因：

车牌候选区域的获取依然是基于传统的数字图像处理方法。

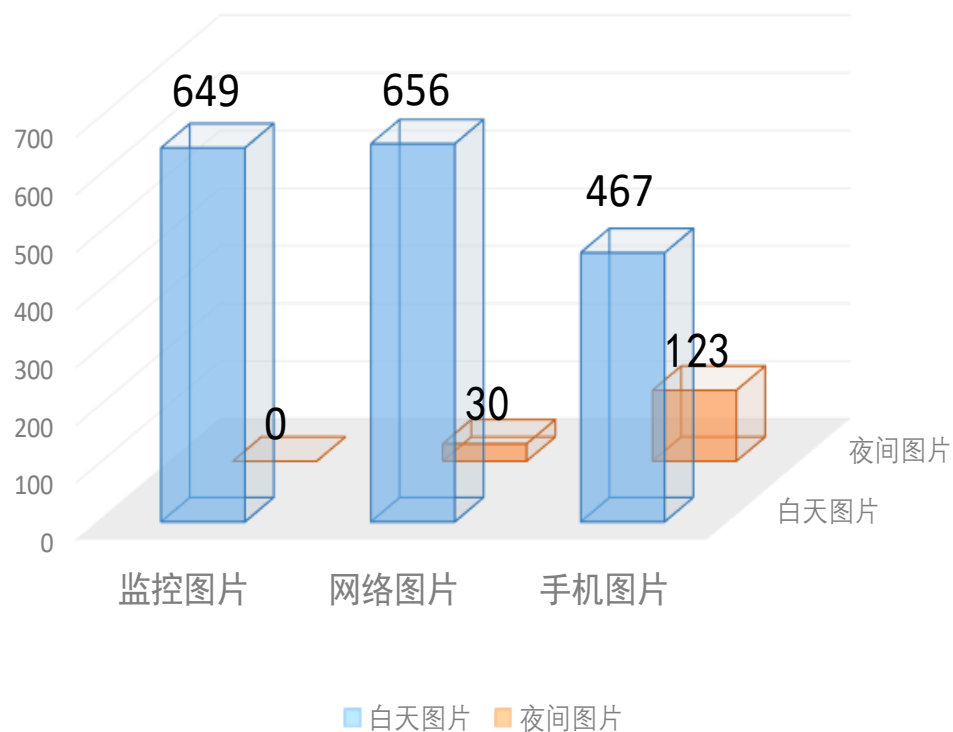
• 采集原始图片集

• 建立车牌区域候选人图片集

• 建立字符判断图片集

• 建立数字和字母识别的图片集

• 建立汉字识别图片集





a) 高速抓拍视角一



b) 高速抓拍视角二



c) 网络图片



d) 手机拍摄白天场景图片



e) 手机拍摄夜间场景图片

- 采集原始图片集

- 建立车牌区域候选人图片集

- 建立字符判断图片集

- 建立数字和字母识别的图片集

- 建立汉字识别图片集

**制作目的：**减轻字符判断图片集标记工作量

**制作方法：**对1485张原始图片依次使用四种定位方法

**元素类型：**153×48的灰度图片

**图片分类：**完整车牌区域+干扰项



6candi3.jpg



7candi0.jpg



7candi1.jpg



10candi0.jpg

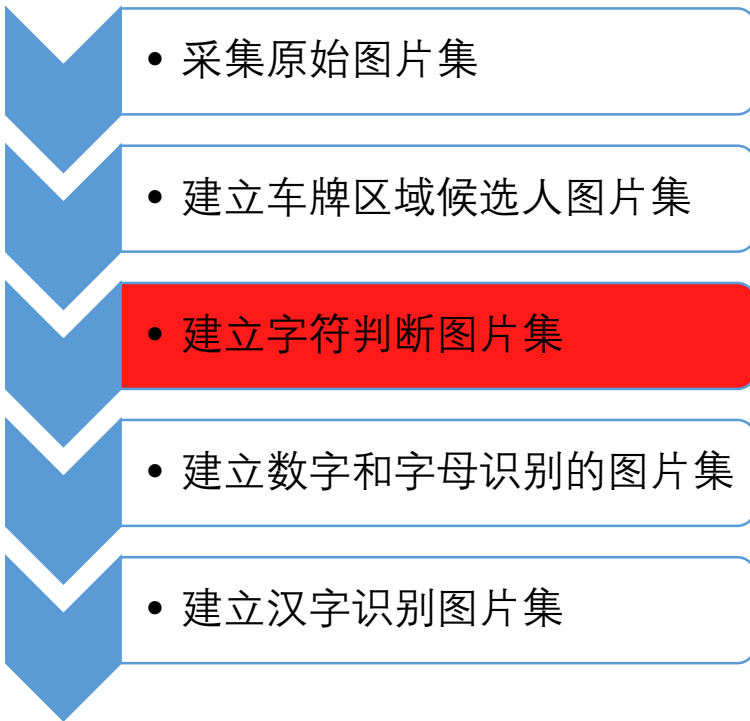


10candi1.jpg



10candi2.jpg

车牌区域候选人图片展示



- 采集原始图片集

- 建立车牌区域候选人图片集

- 建立字符判断图片集

- 建立数字和字母识别的图片集

- 建立汉字识别图片集

**制作目的：**得到用于字符判断分类器的训练集和测试集

**制作方法：**对车牌区域候选人图片集中的所有元素依次提取最大稳定极值区域

**元素类型：**24×24的二值图片

**图片分类：**“字符”+非字符




a) 字符图片



b) 非字符图片





- 采集原始图片集

- 建立车牌区域候选人图片集

- 建立字符判断图片集

- 建立数字和字母识别的图片集

- 建立汉字识别图片集

**制作方法：**对字符判断图片集中的字符图片进一步分类和筛选。

**图片类型：**24×24的二值图片。

**数据结构：**由数字0，2-9这9种数字和不含“0”和“1”的24个英文字母组成，共33类，每类字符各300张图片用于训练，100-300张不等的图片用于测试，测试集图片总数为6281。

- 采集原始图片集

- 建立车牌区域候选人图片集

- 建立字符判断图片集

- 建立数字和字母识别的图片集

- 建立汉字识别图片集

**制作方法：**从网络下载一个车牌汉字图片集，经过筛选、阈值操作后归一化为 $24 \times 24$ 的图片，经过程序随机加入缩放、平移、旋转、噪声扩充样本。

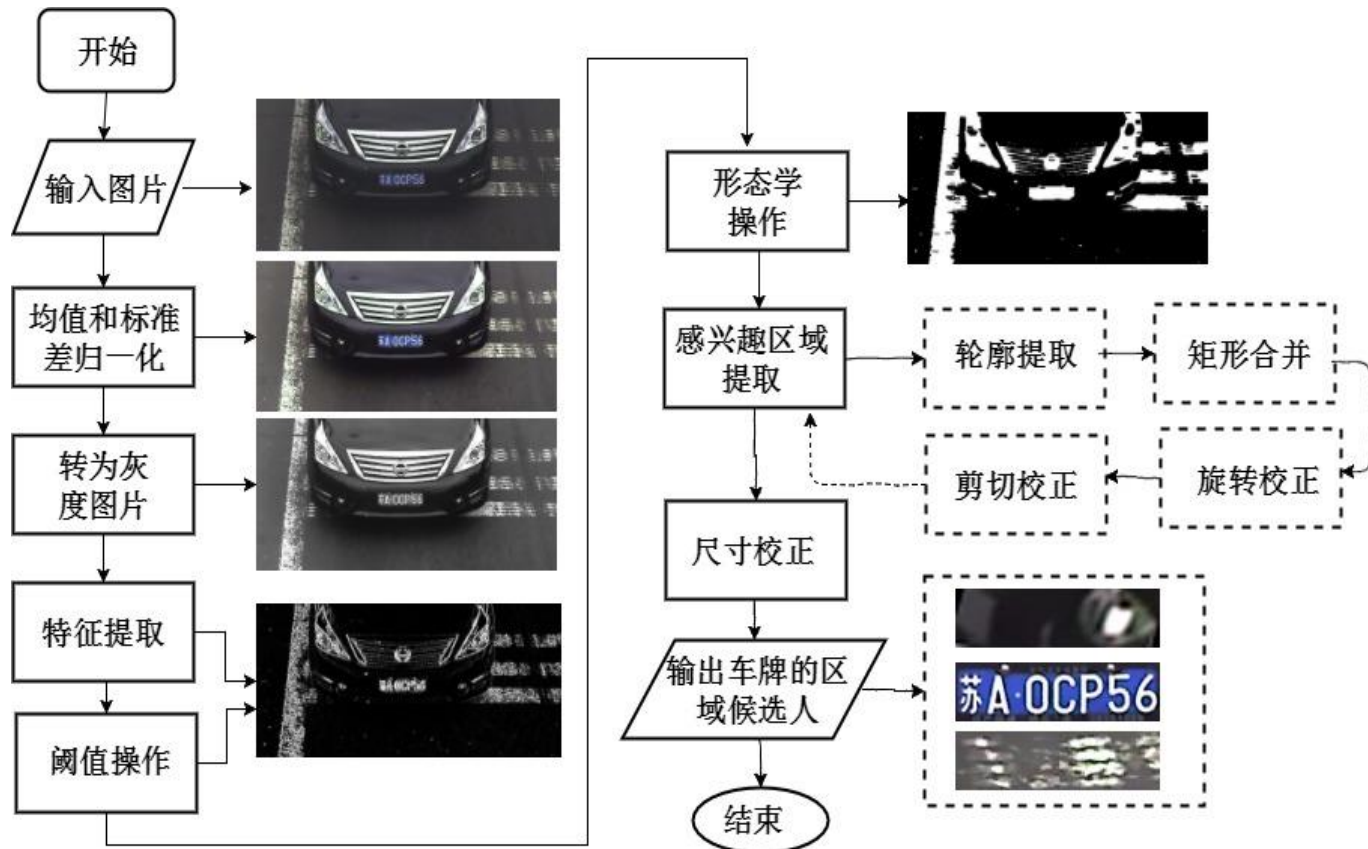
**图片类型：** $24 \times 24$ 的二值图片，为了与卷积网络结构适应，通过四周补零的方式扩充为 $32 \times 32$ 的图片。

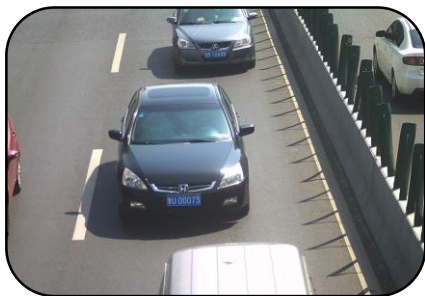
**数据结构：**共有31类，每类字符600张用于训练，200张用于测试。



24×24的中文字符图片

# 车牌区域候选人提取的完整流程：





光照充足，边缘特征突出，但图片色彩失真。

基于边缘特征的检测方法

基于颜色和边缘特征的检测方法



蓝色特征鲜明，背景中包含复杂的边缘信息。

基于区域连通性的检测方法

基于灰度值动态拉伸的边缘增强检测方法



车牌区域反光，颜色偏白，边缘特征被弱化，但车牌区域与周围像素对比度高。

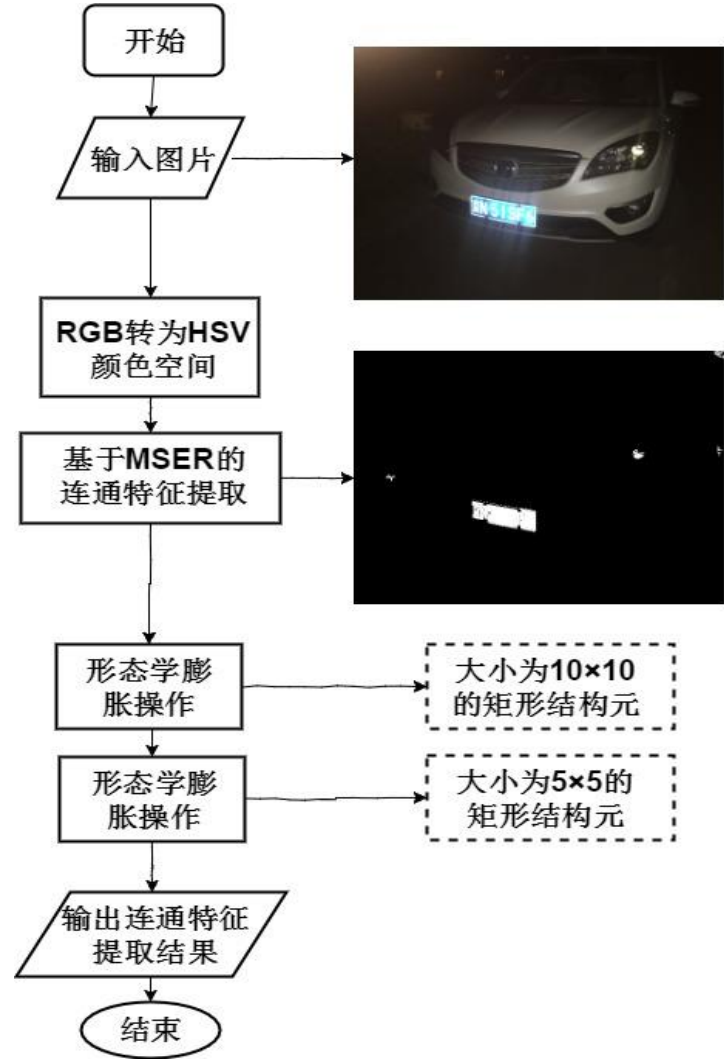


车牌区域较暗，背景中存在较亮区域。

## 基于区域连通性:

**图片类型描述:** 车辆开前灯或者手机开闪光灯时拍摄, 车牌的背景区域过亮, 字符边缘特征被弱化, 车牌颜色偏白, 直接使用基于大津算法的阈值操作会将车牌区域归类为弱边缘, 导致定位失败。

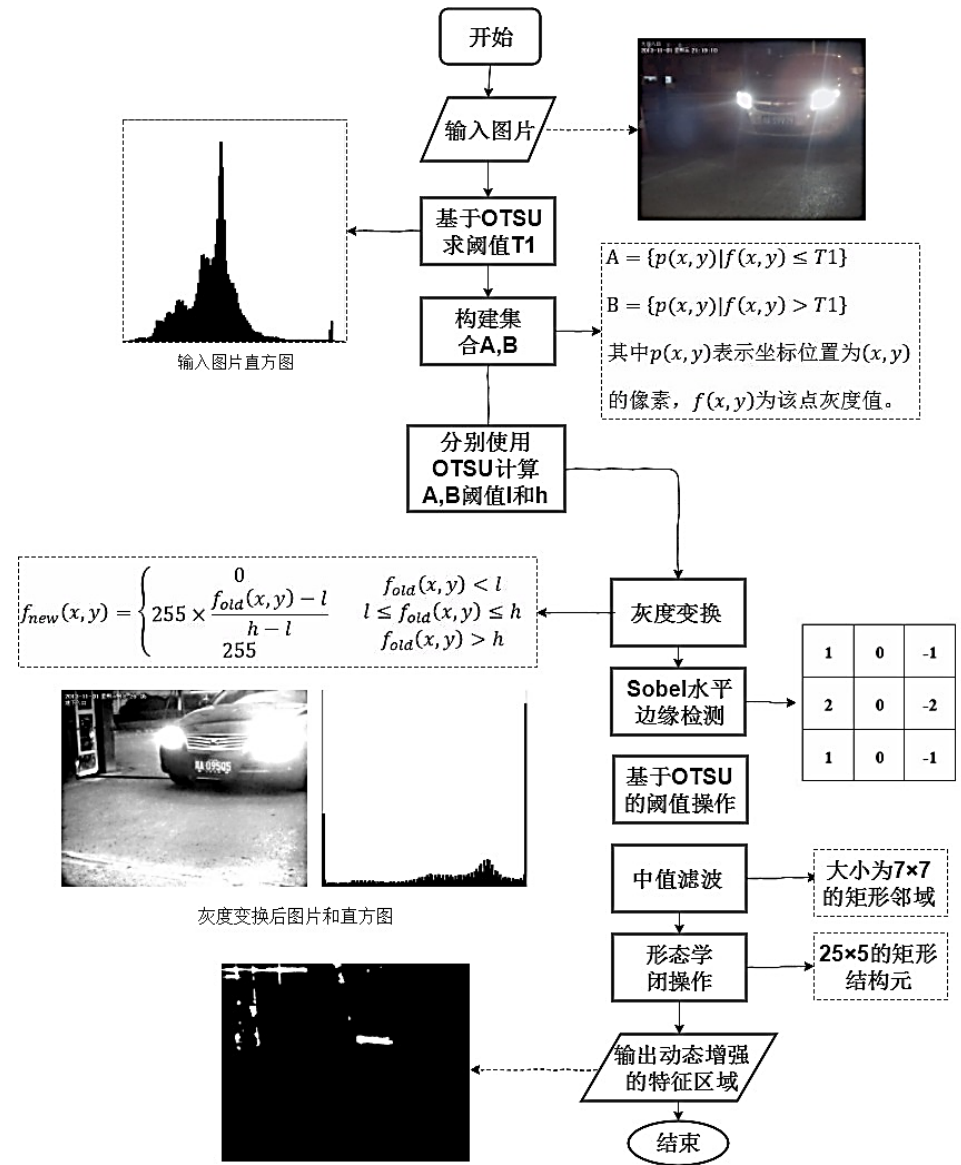
**基本实现思路:** 车牌区域连通, 连通性标记可以跳过阈值操作。



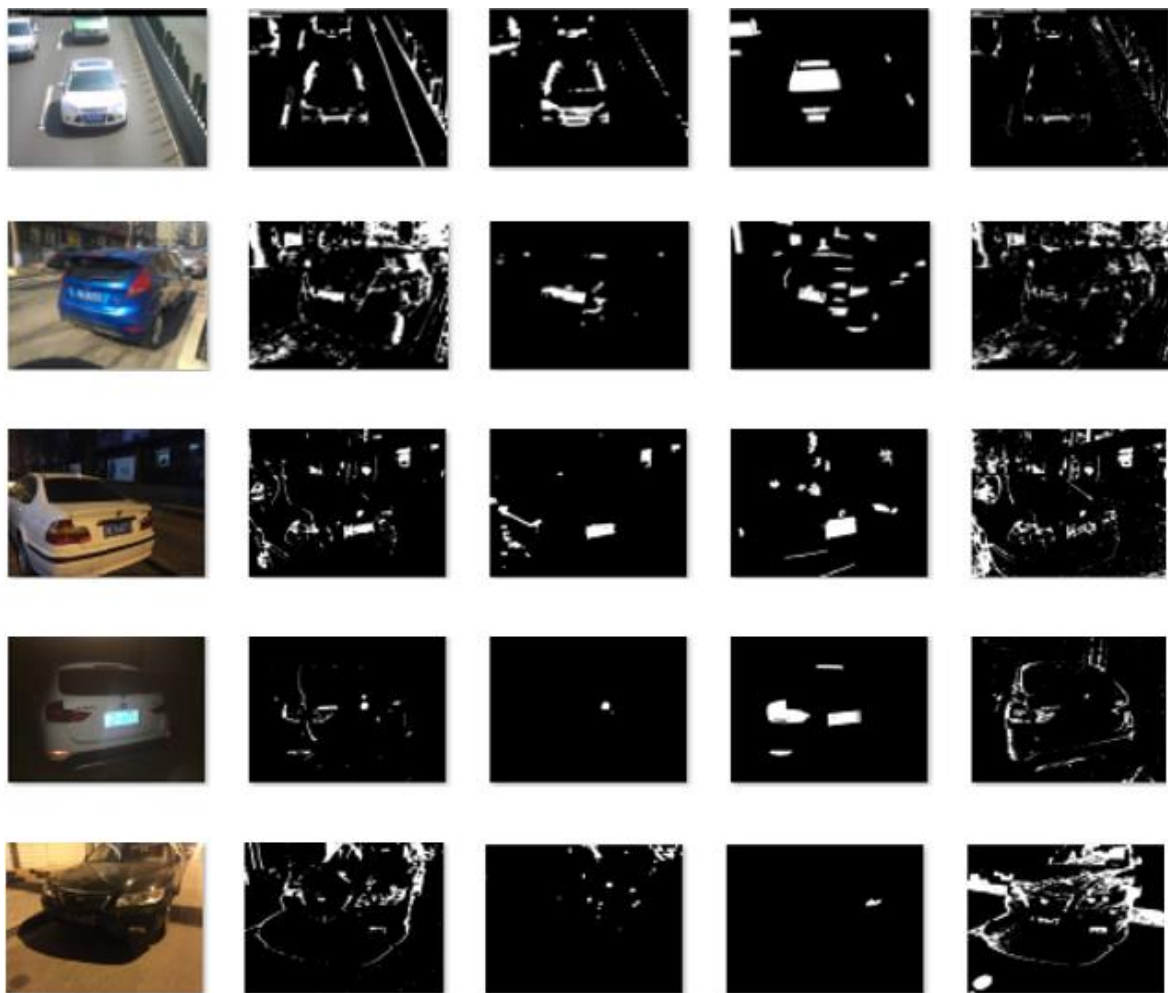
# 基于灰度值动态增强:

**图片类型描述:** 背光条件下拍摄的图片，车辆区域很暗，而背景中存在亮的物体，图片的直方图会存在多个极值。

**基本实现思路:** 车牌区域强度值介于很暗和亮之间，只对中间区域的灰度值拉伸，对很暗和明亮的区域的像素进行抑制，使它们不对边缘做贡献。



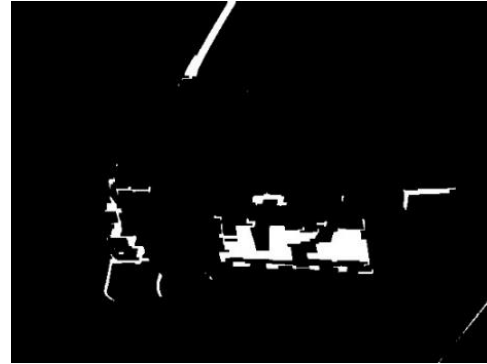
## 不同类型图片的四种特征提取效果展示:



## 矩形合并操作效果:



a) 输入图片



b) 形态学处理结果

c) 轮廓提取结果

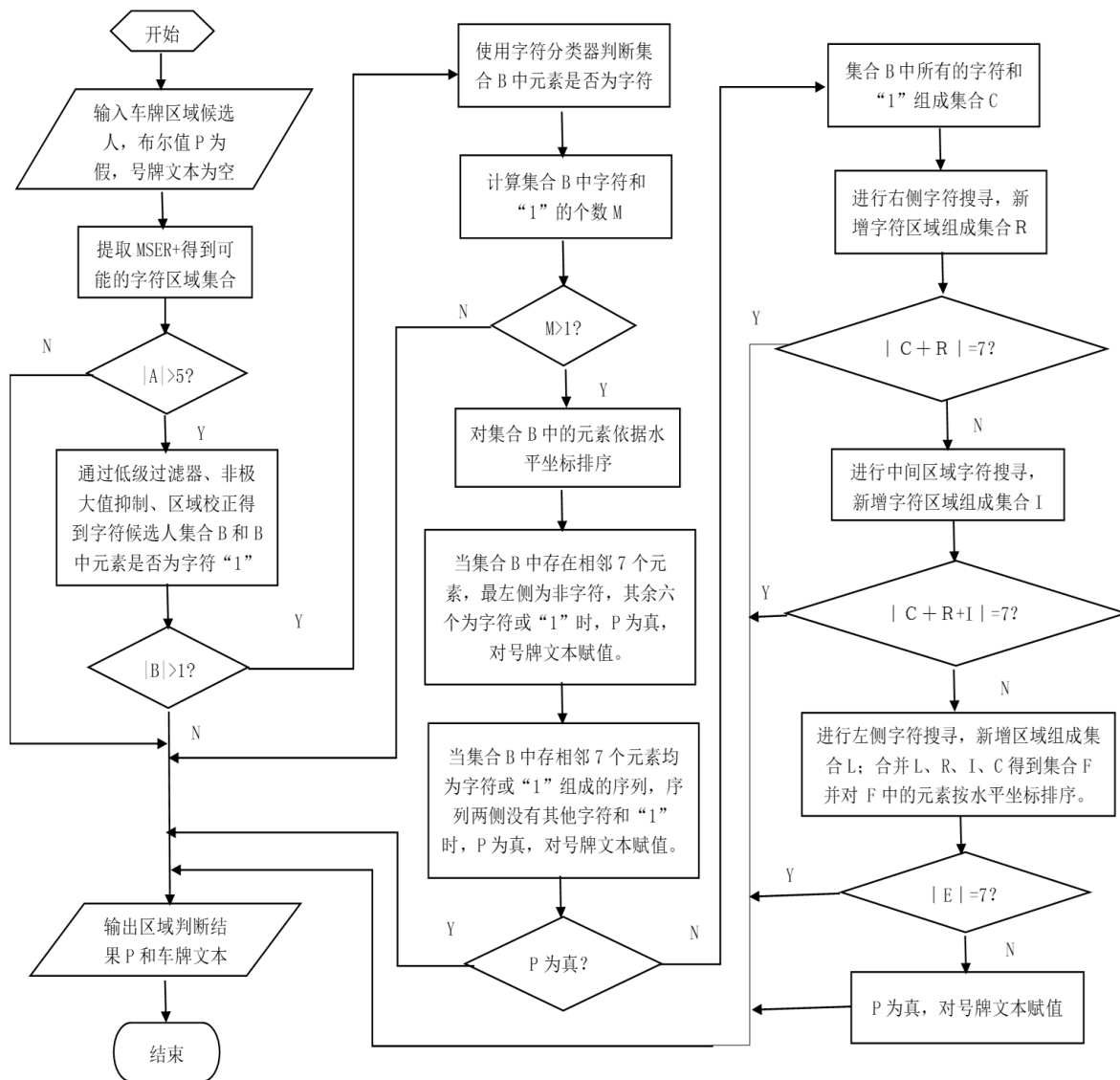


d) 加入矩形合并操作后的结果





# 字符分析完整流程图:



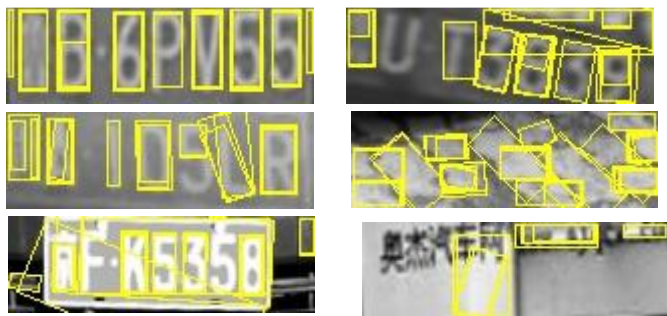
## 基于SVM的字符判断:

**20个描述性特征:** 区域面积、周长、面积与周长的比值、孔洞数、水平穿越次数中值、孔洞大小与字符面积的比值、凸包面积与面积的比值、凸缺陷点的数目、外接矩形左上端的坐标值 $x$ ,  $y$ 、矩形的长度和宽度、笔画宽度的字符和方差、两个一阶矩和三个中心矩。

**模型的训练与测试:** 使用线性核和归一化操作, 经过交叉验证和网络寻参后, 当正则化系数为156时, 模型在测试集识别精准率为95.18%, 召回率为96.60%, F数为95.53%。

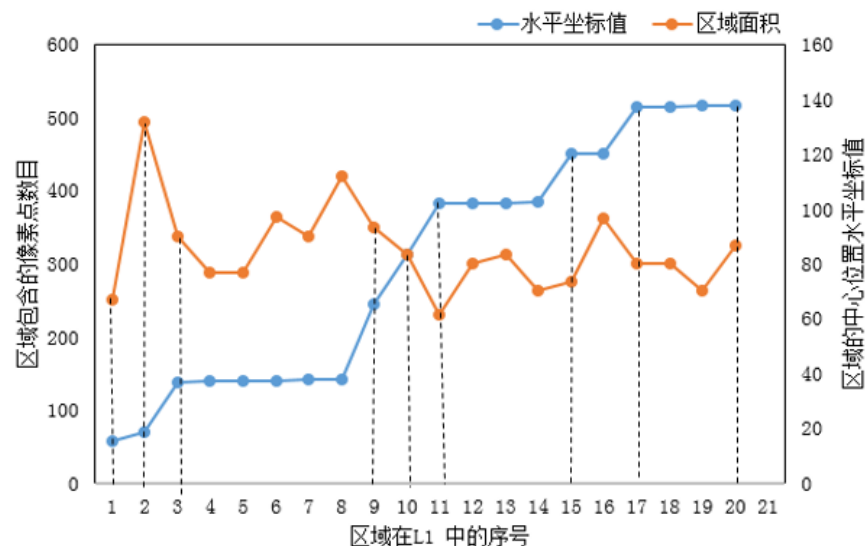
# 基于连通性的区域提取与校正:

1. 使用改进MSER提取连通区域:



3. 非极大值抑制:

3.1 同一区域的重复提取:



2. 低级过滤器处理:



a) 输入效果

b) 过滤器排除的区域

c) 输出效果

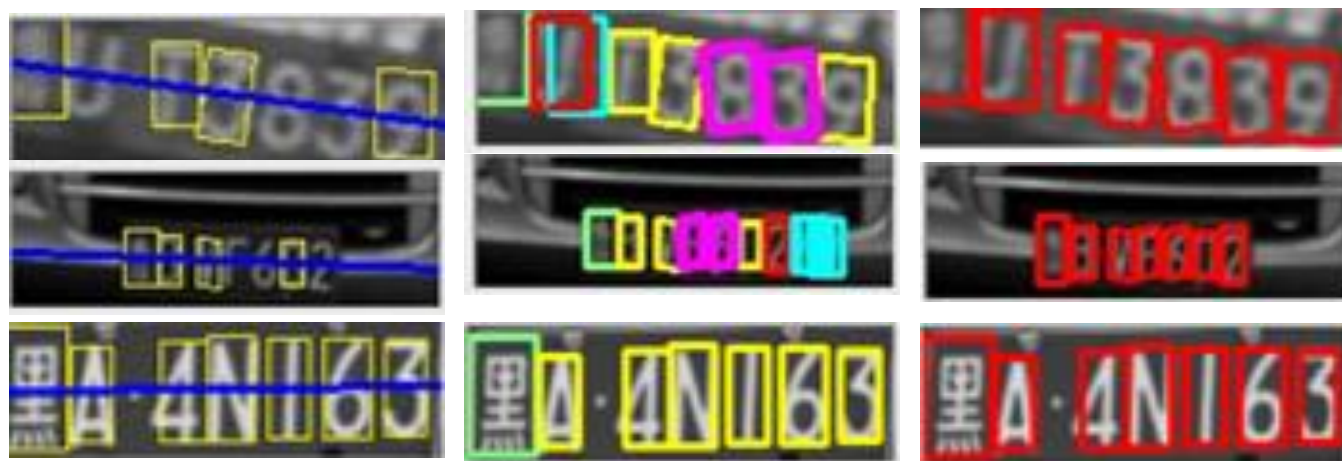
4. 区域校正:



3.2 嵌套问题:



## 字符搜索和最终输出:

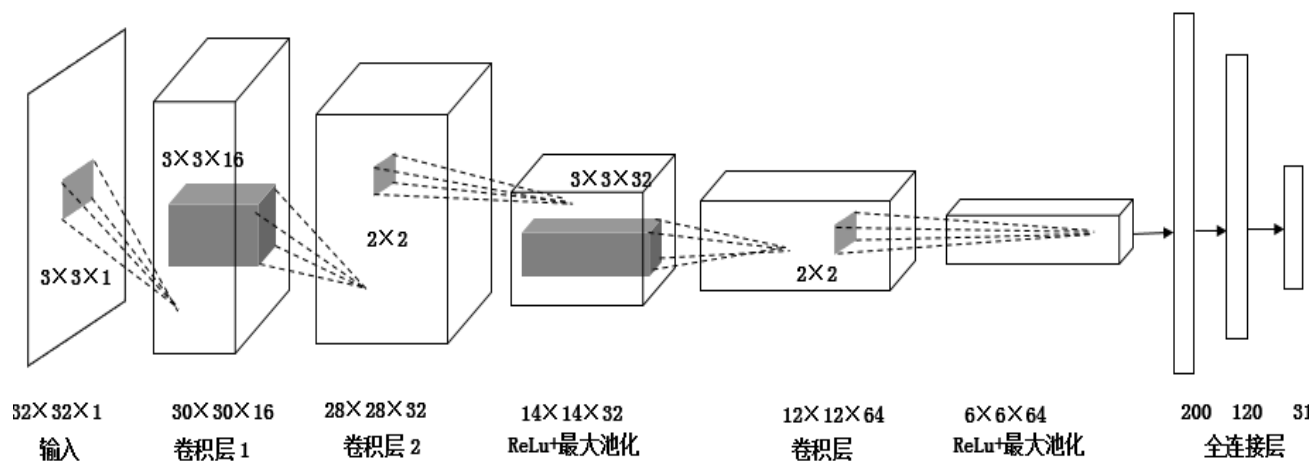


a) 连通区域提取结果   b) 字符搜索结果   c) 最终提取区域

图b中黄色矩形框为图a中区域经字符判断后为字符的对象，被称为“字符种子”，浅绿色矩形框为为非字符对象，粉红色矩形框为在“字符种子”之间被搜索为字符的对象，蓝色矩形框为在最左和最右侧“字符种子”两侧进行“滑窗搜索”但检测结果不是字符的对象，红色是判断为字符的对象；

# 中文字符的识别：基于卷积神经网络

MyLeNet网络结构：



**网络的训练与测试：** 设置批处理的大小为64，输入图片的强度值乘1/255的预处理，学习速率的初始值为0.05，动量矩系数为0.9，迭代衰减系数为0.07，即每隔1000次学习速率下降7%，经过2万次迭代，在测试集上实现了96.4%的正确率。

## 程序编写平台：

计算机程序设计语言：C++11

集成开发环境：Microsoft Visual Studio 2015专业版。

计算机视觉库：使用混合编译OpenCV3.2的源码文件和V3.1版的opencv-contrib源码文件得到的32位Debug模式的OpenCV；Vlfeat3.2.0。

深度学习框架：Windows版本Caffe，编译工具为cmake3.8.1，安装平台为window10，附加了Anaconda3.4.2作为python3.5的管理平台和cuda8.0工具包作用来调用GPU加速应用程序。

解决方案“vlpr”(1 个项目)

vlpr

引用

外部依赖项

头文件

location

basic\_class.hpp

platelocate.hpp

segment

chars\_judge.hpp

erfilter.hpp

utils

kv.hpp

mser2.hpp

util.hpp

characterR.hpp

cnn.hpp

creatDataset.h

源文件

erfilter.cpp

main.cpp

platelocate.cpp

资源文件

CPlate()

operator=(const CPlate & other)

charMats

is1

isPlate

plateMat

plateRotatedRect

plateStr

```
-----../resources/image/lifeCar/TEST_ONLY1/冀000001.jpg
The recognition result is: 川K00000
-----../resources/image/lifeCar/TEST_ONLY1/新H68999.jpg
The recognition result is: 闽H63999
-----../resources/image/lifeCar/TEST_ONLY1/津NRL118.jpg
The recognition result is: 津NRL118
-----../resources/image/lifeCar/TEST_ONLY1/浙AA2270.jpg
The recognition result is: 浙AAF2XT
-----../resources/image/lifeCar/TEST_ONLY1/浙F397C0.jpg
The recognition result is: 浙F391C0
-----../resources/image/lifeCar/TEST_ONLY1/浙G70000.jpg
We don't find any plate in this picture!
Maybe you can choose other method to locate your plate.....
-----../resources/image/lifeCar/TEST_ONLY1/渝B777C9.jpg
We don't find any plate in this picture!
Maybe you can choose other method to locate your plate.....
-----../resources/image/lifeCar/TEST_ONLY1/湘A3685C.jpg
The recognition result is: 湘A3685C
-----../resources/image/lifeCar/TEST_ONLY1/湘AYN355.jpg
The recognition result is: 湘A1N355
-----../resources/image/lifeCar/TEST_ONLY1/皖A1T235.jpg
The recognition result is: 皖4T1P35
-----../resources/image/lifeCar/TEST_ONLY1/皖ATH859.jpg
The recognition result is: 皖ATH819
-----../resources/image/lifeCar/TEST_ONLY1/粤A5DP12.JPG
The recognition result is: 粤AU1P12
-----../resources/image/lifeCar/TEST_ONLY1/粤B3RS91.jpg
The recognition result is: 粤B3RS91
-----../resources/image/lifeCar/TEST_ONLY1/粤SD050L.JPG
The recognition result is: 粤SD050L
-----../resources/image/lifeCar/TEST_ONLY1/苏A6E176.jpg
The recognition result is: 苏ATE1U6
-----../resources/image/lifeCar/TEST_ONLY1/陕AD2N68.jpg
The recognition result is: 陕AD2N68
-----../resources/image/lifeCar/TEST_ONLY1/鲁B010LU.jpg
The recognition result is: 鲁B010HU
-----../resources/image/lifeCar/TEST_ONLY1/鲁B018N8.jpg
The recognition result is: 鲁868059
```