作品介绍

车牌识别：主要使用OCR光学字符识别技术

识别流程：

1、输入图片

2、扫描图片，粗定位到有车牌的图像区域

3、检测车牌颜色

4、图像矫正，车牌精确定位

5、字符分割

6、字符识别

7、输出结果

1. 车牌粗定位

1.1粗定位原理

基于目标检测的方法，训练出Haar特征分类器：Haar-like特征+积分图方法+Adaboost算法+级联，将车牌与非车牌区域分开，从而实现车牌粗定位。

1.2算法优势

以前常见的方法是使用**边缘检测**来定位车牌，首先找到疑似车牌的连通域，然后对目标区域进行筛选，主要是使用一些特殊的知识或信息，直接检测单个图像。

例如：车的颜色，长宽比，边框，等形态学特征，来筛选车牌。但是此算法识别率不高，只能识别较为清晰的图片，对不同环境和不同角度的车牌识别准确率较差，所以我们采用基于统计的车牌检测方法：**Haar特征分类器算法**，它对车牌图像质量要求不敏感，即使图像中车牌处于倾斜、多个并存、夜光或车牌字符出现污渍、模糊等情况，本算法均能快速准确地定位出车牌区域。

1.3粗定位步骤

粗定位可分为**样本创建、样本训练**和**识别测试**两个主要部分。

1.3.1样本创建：

正样本：只包含检测目标的图片即车牌

所有图片尺寸必须相同，例如36\*9

进行灰度处理

noplate_519

负样本：不包含检测目标的任何图片

不要求样本尺寸，但样本量要大于正样本；且负样本不能重复，增大负样本差异性。

进行灰度处理

A03_LF8567_1

1.3.2样本训练：

对Haar 级联分类器进行训练。选取4700个正样本12000个负样本，调用 opencv\_ haartraining. exe和opencv\_createsamples.exe命令工具进行样本训练，该工具封装了 Haar 特征提取以及 Adaboost 分类器训练过程，训练后的Haar分类特征可记录在对应的 xml 文档中。

1.3.3识别测试：

使用opencv中的cascadeclassifier函数进行多尺度检测即可。

二、车牌精定位

2.1精定位算法流程

训练完cascade分类器之后，会输出三个参数，x(宽)、y(高)、尺度。但仅凭这三个参数进行定位是远远不够的，在车牌有角度倾斜的情况下，可能定位的不是很精准。所以为了最大程度的保留车牌图像，我们对cascade目标检测后的区域进行扩展。然后对上下边界进行裁剪，在对其进行位置矫正，最后确定左右边界，来实现精确的车牌定位。

**确定上下边界：**

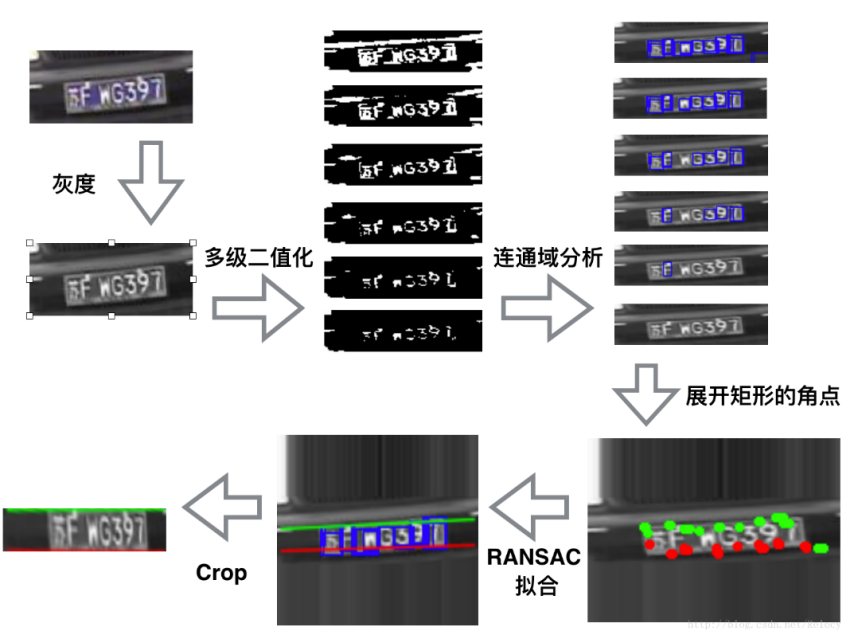
1、使用多个参数对这个区域进行多次自适应二值化。我们对opencv中 adaptiveThreshold函数的k的参数从选择从-50变化到0，做15次二值化

2、对每次二值化的图像进行连通域分析寻找满足字符长宽比的轮廓

3、将矩形框的两个点画出来对下面的点做直线拟合，要用到随机抽样一致(RANSAC) 算法。RANSAC算法能帮助我们剔除做连通域分析时产生的噪声点。

4、找到了上边界和下边界，只需要把这个区域crop出来

流程图如下：



**确定左右边界：**

在做完精定位之后，接着确定左右边界。我们知道存在车牌这类文字密集的地方，往往垂直边缘的聚集就特别强烈，由于在精定位后，车牌几乎是在一个尺度下，所以就无需关心多尺度的问题。

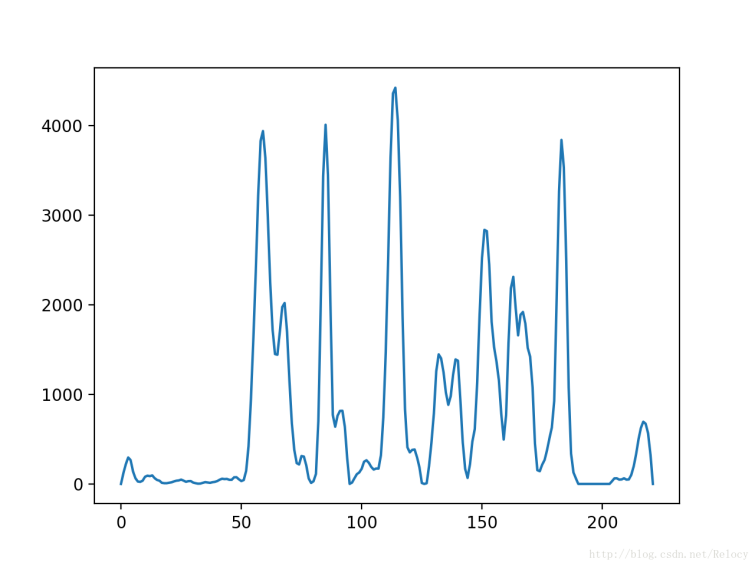
使用垂直边缘来进行车牌精定位：

IMG_256

这种方法和基于边缘的车牌定位方法差不多，只不过不再需要考虑多尺度的问题。我们对上图做Sobel算子找到垂直边缘。

IMG_256

接着我们对垂直方向像素求和，画出投影直方图。



最后对左右两边的数值做一个判断即可得出左右边界，这样我们就成功的精确定位了车牌

IMG_256

三、字符分割

我们使用基于滑动窗口的分割方法。利用一个训练好的正负样本分类器来进行在图像上滑动来获取最佳分割路径。此方法对中文字符的分割效果理想。

**其他算法：**

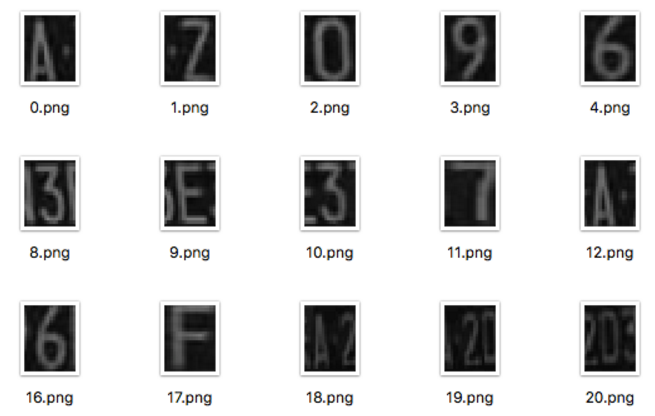
通常也采用垂直投影对字符进行分割，其原理是将二值化图像上的白色像素点垂直累加起来，做出分布直方图，从而找到相邻字符的分界点进行分割。但是对于一些中文字符，分割的情况很不理想，如：川字等。



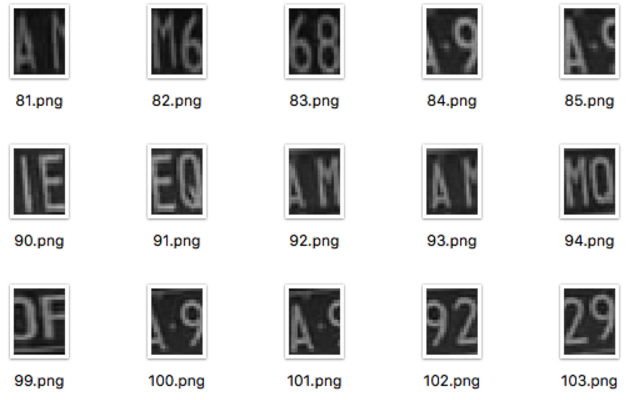
滑动窗口的字符分割法具体流程：

利用一个训练好的正负样本分类器来进行在图像上滑动来获取最佳分割路径。将样本分为下面三类放到CNN模型中进行训练：

正样本：



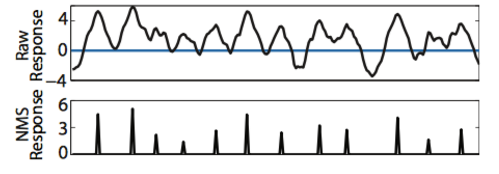
负样本：

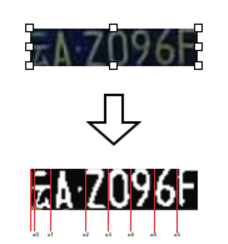


中文字符：



训练完得到一个三分类的分类器，我们把它运行在如下的车牌中，用滑动窗口进行扫描，使得第一个字符的中文概率+后6个字符的正样本概率最大，这个最大的位置即为最佳分割位置。





最后再将分割好的字符放到字符识别的模型中进行识别。

1. 字符识别及颜色识别

4.1算法原理：

使用卷积神经网络CNN来训练模型，它不再是对图片的一个个像素点进行处理，而是使用卷积核对一块像素区域进行扫描，这种方法是为了加强图像信息的连续性，加深神经网络对图像的理解。CNN 包含以下层次：输入层、卷积层、激励层、池化层、全连接层。

4.2字符识别

流程：

1、搭建神经网络模型，进行图像分类时，要设定好类别数量。

2、创建字符与数字的数据集，并贴上标签。例如：27-6、17-7、14-0-3

3、把数据放入建立好的模型中训练，得到训练好的模型

4、把分割好的字符用模型做预测

5、输出结果

4.3颜色识别

与上面类似，车牌颜色判断时可以设定，黄、蓝、绿、白、黑五个类别。我们将图像输入到计算机，计算机最终输出的是图像主要内容分别属于每种类别的概率。如 [ 黄 0.01, 蓝 0.96, 绿 0.01, 白 0.01，黑 0.01] 识别与分类的任务将由卷积神经网络完成。