

# 车牌检测与识别的实现

## 一、课题概述

随着汽车保有量的持续增长，交通拥堵状况和车辆管理压力也越来越严重，高效准确的智能交通管理系统由此产生，并朝着高清化、视频智能分析技术、多媒体调度等趋势发展。车辆是智能交通系统的核心研究对象，拥有自身唯一 ID。车牌号码和车辆之间存在一对一的关系，可以用来进行信息获取、管理和监控。因此一种高效、精确的车牌识别系统是智能交通的核心。车牌识别系统是一个有着广泛实用价值的视觉任务，其主要应用方向有：①城市电子眼监控系统；②停车场智能管理系统；③高速公路自动监管系统等。

车牌识别系统的简单定义为：要求能够将运动中的汽车牌照信息从复杂背景中提取并识别出来。基本工作原理首先对采集的原始图象进行预处理，运用特定定位算法提取出车牌区域，最后采用字符分割和识别算法识别出车牌号信息。

近年来，国内外专家学者和企业对车牌自动识别系统进行了深入研究。但是由于外部环境、车牌自身和其它因素的影响，系统的运行速度、准确性、鲁棒性都有待改进。本实验主要利用边缘特征和颜色特征实现车牌的检测，运用 SVM 和 CNN 模型实现车牌的识别。

## 二、方法介绍

### ● 我国车牌特征

清楚了解需求目标的特征，有利于后续实现。我国标准民用车牌是由一个省份汉字、一个代表地级行政区的字母代码、一个分隔符，后跟 5 位字母或阿拉伯数字组成的 7 字序列。绝大多数尺寸为长 44cm，宽 14cm。此外，不同类型的机动车车牌的颜色特征也是不同的，本实验主要研究了最常见的蓝底白字车牌、黄底黑字车牌、和渐变绿色的新能源车车牌。

## ● 主要流程

了解车牌特征后，我们可以得出车牌识别系统的实现思路。主要流程分为以下三个部分：

- 1) 车牌检测。通常拍摄到的原图含有大量复杂的无关背景干扰，识别系统直接从车辆图片中读取车牌信息较为困难，所以需要先定位车牌；
- 2) 字符分割。对车牌进行水平和垂直方向的校正处理后，通过适当算法把车牌图图像分割成单个字符图像；
- 3) 字符识别，从单个字符图像中提取特征数据，输出字符。

### 1、车牌的检测

车牌检测输入原始车辆图像，输出车牌图像，为后续字符识别建立基础，所以需要保证较高的准确率。传统的车牌检测算法包括基于边缘信息的检测算法、基于颜色信息的图像分割算法、基于小波分析、基于纹理特征和基于形态学运算的检测算法等。这里经过比较，选择了基于边缘和颜色两种定位方法。

#### • 基于边缘检测的定位

第一种是基于车牌边缘轮廓特征的车牌检测算法。

首先对输入的车牌图像进行预处理，目的是为边缘检测做准备。高斯模糊对图像去噪，通俗来讲就是产生模糊的效果，严格意义上是一种数据平滑技术（可理解成图片中每个像素都取周边像素的平均值，中间点失去细节，产生模糊的效果），取值范围越大效果越强烈。高斯模糊中的模糊半径会给结果带来明显的变化。不宜过高或过低，经过后续的实验，确认 BlurSize 取 5 个像素。灰度化处理为边缘检测算法准备灰度化环境。与人类记忆辨别色彩的能力不同，对于计算机而言，色彩图像相对于灰度图像难处理得多



图 1 原图；图 2 高斯模糊后；图 3 灰度化处理后

边缘检测是该检测算法的核心，便于后续区分车牌。边缘检测的原理是考察像素变化最明显的点，车牌形状规则，适合使用边缘检测。这里的三张图，从左到右分别是 Sobel/Canny/Laplacian 算子边缘检测的效果。从中可以发现：Sobel 算子产生的边缘有强弱，抗噪性好，但是对边缘定位不是很准确，图像的边缘不止一个像素；Canny 算子产生的边缘很细，实际会导致车牌区域和附近连在一起，对后续的闭运算产生干扰；Laplacian 算子抗噪声能力较差。因为这一步边缘检测的主要目的不是检测边缘，而是为了框出车牌区域，所以综合考虑，最终选择抗噪声能力强、速度快、实现简单的 sobel 算子。



图 4 Sobel/Canny/Laplacian 算子边缘检测效果

接下来对检测到的边缘，进行二值化处理，为后续的形态学算子 Morph 等准备二值化的图像。原理是对图像的每个像素做一个阈值处理，在灰度图像中，每个像素的值是 0-255 之间的数字，代表灰暗的程度。如果设定一个阈值 T，规定像素的值  $x$  满足一定条件时，则每个像素的值仅有 {0,1} 两种取值，0 代表黑、1 代表白，图像就被转换成了二值化的图像。二值化图像中的白色是没有颜色强与暗的区别。直接使用 opencv 的二值化函数 threshold 时加上自适应阈值 OTSU 参数实现。其中大津法（最大类间方差法）对图像进行阈值化处理，剔除一些梯度值较小的像素，使图像边缘更清晰。该方法按照图像的灰度特性，将图像分为目标部分和背景部分，通过最大化目标和背景的类间方差来获取分割的最佳阈值。

再对二值图像进行数学形态学操作。闭运算：先膨胀后腐蚀，使得车牌字符区域连接起来；开运算：先腐蚀后膨胀，消除车牌噪声。目的是将车牌字母连接成为一个连通域，便于取轮廓。形态学操作的对象是二值化图像。腐蚀，膨胀是许多形态学操作的基础。观察白色区域面积。闭操作的结果一般是可以将许多靠近的图块相连称为一个无突起的连通域。所以先进行一次闭运算，对图像进行先

膨胀后腐蚀，使得车牌的字符区域连接起来，便于取轮廓；然后进行一次开运算，对图像进行先腐蚀后膨胀，来消除车牌噪声。经过形态学运算，原图像的候选车牌区域变为长方形的连通区域。



图 5 二值化处理后；图 6 闭操作后；图 7 开操作后

取轮廓将连通域的外围勾画出来，便于形成外接矩形。取轮廓操作就是对得到的白色连通区域的轮廓进行处理得到矩形边界框，将图像中的所有独立的不与外界有交接的图块拿出来。然后根据这些轮廓，求最小外接矩形。注意这里用的矩形是 RotatedRect，意思是可旋转的。因此我们得到的矩形不是水平的，这样就为处理倾斜的车牌打下了基础。

尺寸判断是为了排除不可能是车牌的矩形。尺寸判断操作是对外接矩形进行判断，以判断它们是否是可能的候选车牌的操作。考虑到我国车牌的长宽比特征为 22:7，在车牌定位过程中可能会损失部分车牌信息，故取长宽比阈值为 2-5.5，进一步缩小范围。

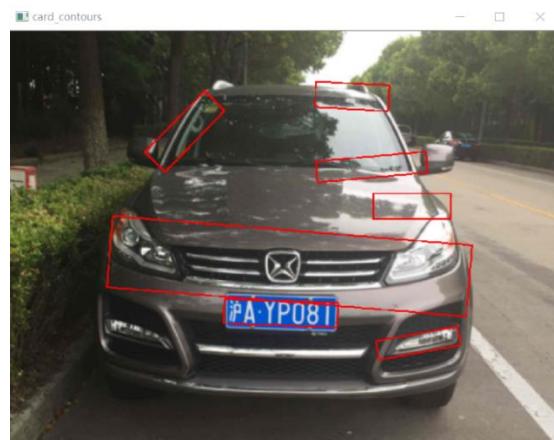


图 8 基于边缘检测算法效果

存在的问题：1) 轮廓不精确；2) 可能的定位过多，仍需要进一步筛选

- 基于颜色分割的定位

介绍的第二种车牌定位算法基于车牌的颜色特征。利用车牌的颜色特征信息（字符颜色或者背景颜色）快速定位到与车牌颜色有关的区域。

关于颜色定位首先我们自然想到的解决方案就是：利用 RGB 值来判断。如果我们想找出一幅图像中的蓝色部分，那么我们只需要检查 RGB 分量（RGB 分量由 Red 分量--红色，Green 分量--绿色，Blue 分量--蓝色共同组成）中的 Blue 分量就可以了，Blue 分量是个 0 到 255 的值。如果我们设定一个阈值，检查每个像素的 Blue 分量是否大于它，就可以知道这些像素是不是蓝色的了。但是这个阈值的确定、复合色原色的比例难以确定。

将图像的颜色空间从 RGB 转换到 HSV。HSV 是一种圆锥模型，三个分量分别代表色调 H、饱和度 S、亮度 V，更符合人的视觉感知结果。尤其是 HSV 颜色空间把明度作为一个单独的分量提取出来，应用在车牌定位中能实现对颜色敏感而对光照不敏感的算法，从而有效克服车牌褪色和光照不理想的恶劣条件。

第二步是对每种颜色设置对应的掩模（mask）以去除背景。这一步主要是利用 OpenCV 自带的 `inrange` 函数。对于给定的 HSV 图像，设置一个颜色的范围，就可以把这个不在这个颜色范围的像素点全部设为 0，在这个颜色范围内的像素点全部设为 255。H 分量是 HSV 模型中唯一跟颜色本质相关的分量。只要固定了 H 的值，并且保持 S 和 V 分量不太小，那么表现的颜色就会基本固定。为了判断蓝色车牌颜色的范围，可以固定了 S 和 V 两个值为 1 以后，调整 H 的值，然后看颜色的变化范围。通过一段摸索，可以发现当 H 的取值范围在 200 到 280 时，这些颜色都可以被认为是蓝色车牌的颜色范畴。于是我们可以用 H 分量是否在 200 与 280 之间来决定某个像素是否属于蓝色车牌。黄色车牌也是一样的道理，通过观察，可以发现当 H 值在 30 到 80 时，可以作为黄色车牌的颜色。

固定了 H 的值以后，如果移动 V 和 S 会带来颜色的饱和度和亮度的变化。当 V 和 S 都达到最高值，也就是 1 时，颜色是最纯正的。降低 S，颜色越发趋向于变白。降低 V，颜色趋向于变黑，当 V 为 0 时，颜色变为黑色。因此，S 和 V 的值也会影响最终颜色的效果。

第三步对仅有黑白两色的二值图参照之前车牌检测算法，使用形态学操作、取轮廓等方法截取车牌外接矩形。

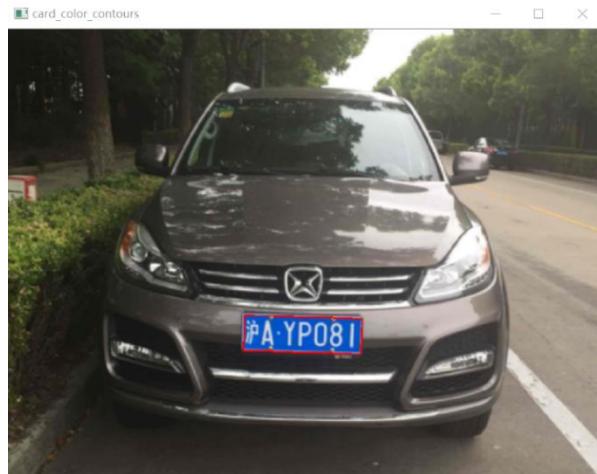


图 9 基于颜色检测的效果

## 2、车牌的识别

车牌的检测已经定位出车牌区域，接下来包括单个字符字符的分割和识别。

### 2.1 字符分割

提取出来的车牌区域内图像通常存在倾斜，因此需要倾斜校正。该算法主要依据仿射变换来保持二维图形的形状。首先确定矩形框的四个顶点，根据坐标判断旋转角度。计算得到其中三个顶点变换后对应的坐标。利用变换前后坐标求变换矩阵计算变换后图像。倾斜校正的结果如下图所示：

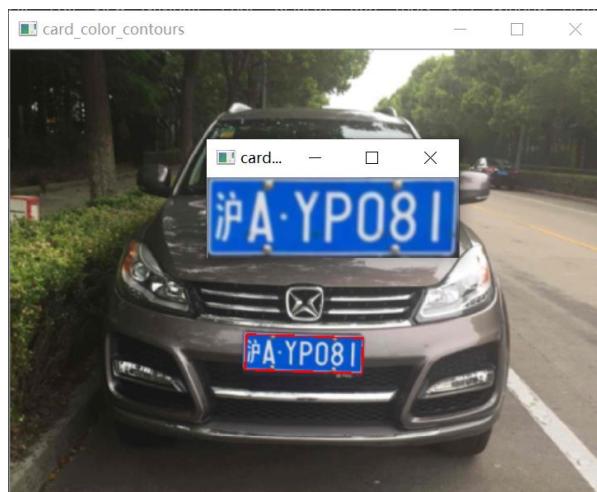


图 10 倾斜校正

由于黄色和蓝色车牌使用的二值化策略不同，所以需要进行一次颜色判断。切除四周的边可以排除干扰。扫描图像，统计出来最多像素点的颜色就是车牌的颜色。为使车牌字符更加清晰，统一进行二值化处理。黄绿色车牌字符比背景暗，蓝色车牌字符比背景亮，对黄绿色车牌进行按位取反。二值化后的灰度图如下：



图 11 灰度化处理

中国车牌有妨碍识别的铆钉，因此在二值化之后还需要一个去除铆钉的操作。扫描每行时判断跳变次数，若某行跳变次数较少，可以把该行所有像素赋值为 0。

字符的分割采用投影分割法，对二值化图片的像素分布直方图进行分析，利用其波峰波谷特征，找出相邻字符的分界点进行分割。对二值化图像，画出其水平方向投影直方图，即统计每一行白色像素点的个数。设定合理的阈值后对这张直方图进行峰谷分析。如果车牌图像含上下边框，也可能有多余的峰出现。再画出垂直方向投影直方图。每个峰对应一个字符，按照这些竖线进行切割，得到单独的字符图像。把外接矩形截取出来后归一化到统一格式。

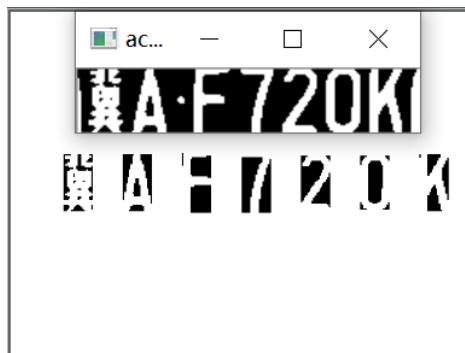


图 12 字符分割结果

## 2.2 字符识别

因为仅包括有限的汉字、大写英文字母和数字，车牌的字符识别与一般的文字识别相比难度较低。但要求识别率尽量高，识别速度尽量快。本实验通过对汉字和数字字母分别训练模型，比较了 SVM 和 CNN 的表现。

- SVM

首先对图片进行预处理抗扭矩操作，再提取方向梯度直方图特征（HOG）作为描述符。把图像分成小的连通区域 cell，采集每个 cell 中各个像素点的梯度或边缘方向的直方图，把直方图组合起来即可构成特征描述符。采用的训练及测试样本为 20\*20 大小的二值化字符图片。经过训练和测试发现，SVM 方法在汉字、数字和字母的识别中都取得了较高的准确率，识别速度非常快。

- CNN

对数字&字母训练时迭代了 100 次，对汉字训练时迭代了 300 次。训练图像大小为 40\*32，在验证集和测试集上都取得了很高的准确率，但是耗时过长。

### 三、实验与讨论

#### 实验设置和实施方案

对比分析之前两种方法的效果，最终方案为先使用颜色定位，失败则用边缘特征方法。输入数据集，经过检测和识别算法，输出成功识别每个车牌的信息或者报告失败，最后统计成功率。

数据集：

- ① 自行实地拍摄的 20 张理想条件下的车辆图片；
- ② 包含不同拍摄角度、不同光照条件、不同分辨率的 250 张模拟真实摄像头获取的图像。

#### 评价指标

##### 1) 识别成功率

- ① 主观判断：唯一准确识别车牌位置，允许倾角和略微边框存在，不允许缺失
- ② 客观检查：可识别出 7 个字符则检测成功，全部正确则识别成功

##### 2) 识别用时

## 实验结果

	总数	客观检测成功率	客观识别成功率	耗时
SVM	200	59.5% (119)	38.5% (77)	94s (0.47s/张)
CNN	200	59.5% (119)	38% (76)	820s (4.1s/张)

理想图像检测识别成功率为 100%

## 评述效果和优缺点

- 优点：

- 1) 充分利用我国车牌特点
- 2) 对输入图像的处理效果较好
- 3) 算法较为简单

- 缺点：

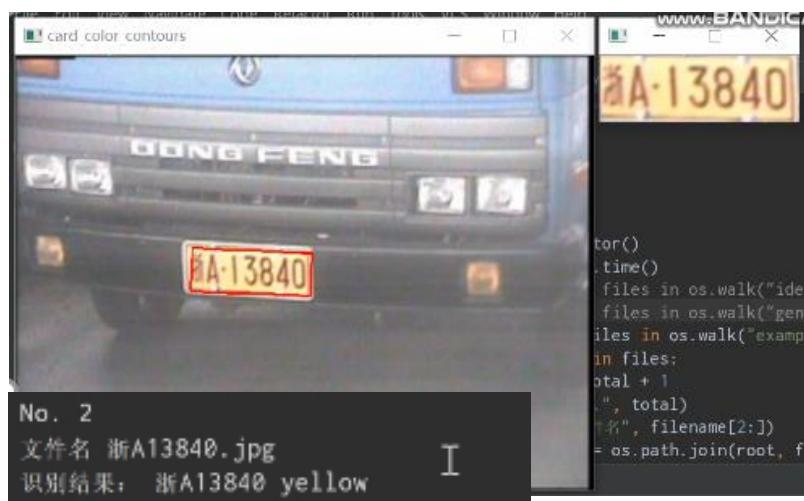
- 1) 识别成功率受测试环境（图片清晰度、拍摄角度、天气条件）影响较大
- 2) 无法识别多行、不规则形状车牌

## 基于颜色特征算法车牌识别成功实例

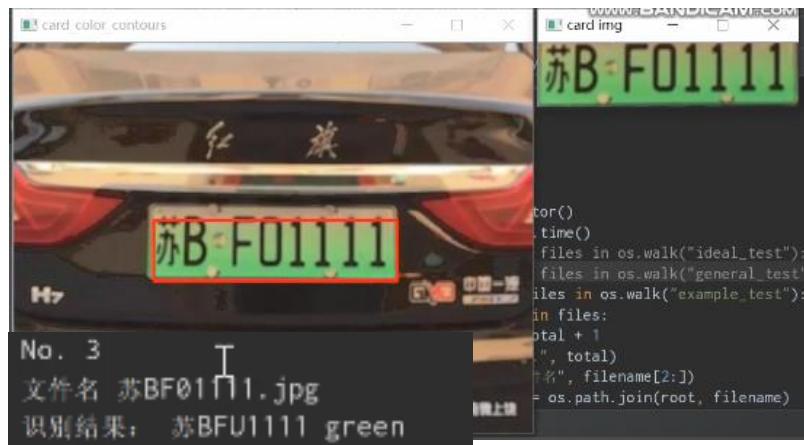
### 蓝色车牌



### 黄色车牌

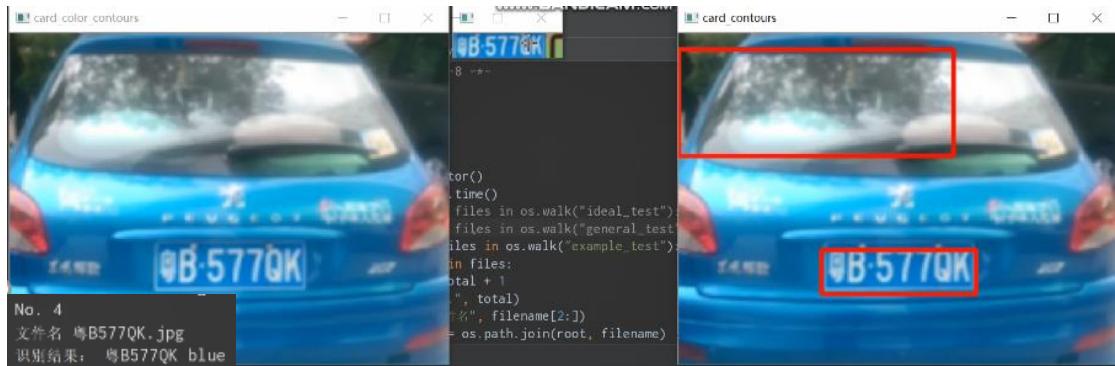


### 绿色车牌



## 基于边缘检测算法车牌识别成功实例

### 蓝色车辆



### 光照条件不理想失败实例

