

**毕业实习报告**



**题目： 基于OpenCV的车牌识别系统设计与实现**

**系 名 计算机科学与技术系**

**专 业 软件工程**

**学 号 6017208033**

**学生姓名 刘**　**晓**　**康**

**指导教师 王**　**国**　**振**

**2021年 11月 28日**

目　　录

[第一章　　绪论 1](#_Toc89448369)

[1.1　课题来源及意义 1](#_Toc89448370)

[1.2　国内外发展现状 1](#_Toc89448371)

[1.2.1　国外发展现状 1](#_Toc89448372)

[1.2.2　国内发展现状 2](#_Toc89448373)

[1.3　课题研究目标和内容 2](#_Toc89448374)

[第二章　　车牌定位 4](#_Toc89448375)

[2.1　车牌特征 4](#_Toc89448376)

[2.2　定位流程 5](#_Toc89448377)

[第三章　　字符分割 11](#_Toc89448378)

[3.1　切割方式 11](#_Toc89448379)

[3.2　切割流程 11](#_Toc89448380)

[第四章　　字符识别 15](#_Toc89448381)

[4.1　字符识别方式 15](#_Toc89448382)

[4.2　识别流程 15](#_Toc89448383)

[第五章　　总结与展望 18](#_Toc89448384)

[5.1　总结 18](#_Toc89448385)

[5.2　展望 18](#_Toc89448386)

[参考文献 19](#_Toc89448387)

# 

# 第一章　　绪论

## 1.1　课题来源及意义

　　随着人们的生活水平的不断提高以及对日常出行需求的不断增长，汽车成为越来越多人出行所选择的交通工具。汽车行业和公共交通的迅速发展，汽车已经逐步走入千家万户之中。伴随着汽车数量的迅速增长，道路交通的管理需求也不断扩大，道路管理中的一些问题也逐渐显示出来。大量汽车投入到交通之中，道路疏通，车辆管理等问题依靠传统的人工方式所付出的财力、物力以及时间成本也越来越高。现代计算机及网络的发展使得利用计算机代替传统的人工方式来对车辆进行管理变得可行起来。车牌作为汽车最显著的身份特征，是现在选定特定车辆目标的最主要手段之一，因此如何从监控或其它汽车相关图片之中处理分析识别出车牌号码是现代车辆管理中至关重要的一节。通过车牌识别的技术，实现对车辆的自动化管理方式是未来交通发展的趋势。

　　对与车辆的行为管理最主要依靠方式之一则是以车牌号码所对应的车辆信息进行录入。所以对车牌识别的实现对此有着重要的意义，借此可以运用在道路交叉路口、各处的停车场、加油站、高速公路的收费站、ETC等多种场合之中。车牌识别通过汽车图像获取车牌号码，实现高效率低成本的智能化管理方式，有效的节约各类成本。车牌的字符识别本子上其实是通过对车牌上的数字、字母与汉字进行一定区分对比之后，将获得到的结果以计算机中字符串的形式展现出来，而字符得识别中，汉字的识别表现得最困难，主要原因在于汉字是我国专用的，这就导致了很多国外的先进的车牌识别系统大都不能够适用于我国。本次选题是为了更好的运用现代科技及网络资源来达到国内车牌的车牌识别。

## 1.2　国内外发展现状

　　由于识别系统在各个领域之中的火热崛起，对与车牌识别的系统在国内外也有不算少的相关研究。车牌识别的技术在国外起步较早，之后再国内也开始跟上。

### 1.2.1　国外发展现状

车牌识别最先是由国外所发起的，无论是研究时间还是先进性都较于领先我国。在二十世纪的中后期，国外就已经开始了对车牌识别算法的研究，并且根据这些算法设计了汽车牌照识别的软件，在实际生活中进行了应用，取得了很好的效果。有着许多人员在这片领域之中投入了大量的资源开展实验和总结。在国外的部分国家之中，例如德英日意以及加拿大这几个国家，早在上个世纪的后期就已经把这种技术研究发展得很成熟了，也提出了大量关于车牌的检测识别方面的高技术方法，并且已经开始在市场上投入使用，虽然说他们的方法不大一样，但是对于这项研究也投入了大量的精力和资金[1]，许多国家都有着不错得成果展现。如以色列的Hi-Tech公司所推出See/Car系统，通过视觉传感器获得车辆车牌信息对车辆管理效果显著；新加波的OptAsia公司的VECON-VIS车牌识别系统，据说准确度也相当高；以及著名的日本的LUIS车牌识别系统[2]等等。

### 1.2.2　国内发展现状

　　车牌识别的技术虽然在国外起步较早，但之后再国内也开始跟上步伐。在早期是用一些较为零散的图像处理方法来应用与车牌识别之中，之后逐渐形成系统般的方法。因为我国的车牌与国外的车牌有着较大差异，所以对于国内的车牌识别方法大都无法直接使用国外的研究技术。虽然无法直接使用国外相对要成熟的技术，但是其中的方法与思想却是仍然有非常多值得去向其借鉴的地方。

在2018年11月份，公安部所发布的汽车保有量已达到了2.35亿辆,同比约增长12.0%,其中新能源车101万辆,同比增长83%[3]。新能源车和普通车的使用量日益增多,智能交通识别变的越发困难。为了解决这一现状，我国相关的研究人员在二十世纪九十年代开始逐步地探索研究汽车车牌照的识别技术。虽然我国的车牌识别虽然要较晚于其他国家，但是在我国大量的研究人员的努力之中，仍然可以看到一系列效果优异的研究成果存在。例如在众多我国相关的研究人员中，胡雪会,王波涛[4]等各研究人员利用OpenCV作为使用工具提出了一种改进的灰度跳变的车牌定位算法，有效的提高了车牌定位的准确率。黄社阳,刘智勇,阮太元[5]等研究人员则是提供了另一种方法，提出了一种基于HSV颜色空间和SVM相结合的车牌提取算法。该方法对多种车牌能够实现较好的精确定位与提取。甘玲,林小晶[6]等研究人员是在字符的分割上提出了一种改进过的连通域提取车牌字符的分割算法，对字符的分割变得相对更加的准确、稳定。

近些年来，人民生活水平的富裕，车辆的拥有人数飞快上升，车牌识别的相关技术在我国也变得愈加重视。除去交通领域内的研究人员、研究公司的不断深入研究，学术、教育领域的人员也越来越多开始对此领域开始了自己的研究工作。

## 1.3　课题研究目标和内容

车牌号码的识别方面,在以前，传统的解决的方法是分为了字符的分割和字符的识别两个阶段来实现。如今大都分为车牌定位、字符分割、字符识别三个阶段。

　　本文以OpenCV为基础，能够较为准确的定位到带有汽车图像的汽车车牌区域。将定位到的汽车车牌区域的图像进行处理切割为单个字符的图像，最后将分割后的各个字符图像能够较为正确的识别处理。即以下的几个目标：

* 1. 如何实现车牌的定位功能：

对车牌识别系统的基本功能，车牌定位功能进行设计。通过对车牌的基本特征分析进行车牌位置的定位。

* 1. 如何实现车牌号码的字符分割功能：

通过垂直投影法，对车牌图像字符间隔之间的特征进行分析，实现将一副图像中的各个字符给分割成一个个单独的字符图像。

* 1. 如何实现字符识别功能：

通过OpenCV的sample函数，对字符图像进行处理并识别出字符。

* 1. 整体实现对车牌号码的识别完成上述功能。

# 第二章　　车牌定位

## 2.1　车牌特征

　　车牌号码是在机动车前后两面分别悬挂着的板材，各国之间使用的材质各不相同。在车牌上会登记显示着车辆的登记地区、号码或者其他的基本相关资料。各国之间的车牌外观、材质和内容之间都有些各自的区别及特征存在。

　　想要准确的定位到车牌所在区域，车牌的特征是车牌定位的关键信息，车牌的识别技术在各方面大多要依靠车牌自身所特有的一些特征来进行分析。以一般我国的车牌来说，首先，在车牌的外观上，一般的车牌都是长方形的形状，并且在长度与高度上都有着统一的标准规定；其次，在车牌的字符上，字符由左到右依次排列，通常情况下第一个字符为汉字，第二个字符为英文字符，各个字符的大小都是由统一的规定，各个字符之间的间隔大小也是相同的；最后，在车牌颜色上看，我国的车牌颜色不同与外国，呈多种颜色，如最常见的目前是普通的小型客车的蓝底白字与最近几年迅速发展的新能源汽车的绿底黑字。它们的车牌底色与字符颜色通常不一样，切相差较大，使得在计算机处理及灰度图上会有一些较明显的变化。

　　车牌定位方法大致有几种，第一种是通过形态学处理和车牌的颜色特征的方法；第二种为根据车牌的形状特征为基础的定位方法，根据车牌的矩形特征和矩形的长宽比来进行；第三种方法是基于Lab的颜色模型和聚类分割以及车牌的性质特征的定位方法[7]。

　　目前本文所使用的车牌定位最主要依靠两点进行。其一为颜色，在我国，最常见的车牌的颜色特征大致可以分为几类，主要由大型汽车车牌的黄底黑字，普通小型汽车的蓝底白字以及在最几年开始广泛使用的新型能源汽车车牌的绿底黑字。其二为车牌的尺寸、字符特征，车牌第一个字符为省、自治区、直辖市的汉字字符简称，第二个字符为发牌机关代号(通常为英文字符)，之后为英文数字之间混合组成的字符。各个字符之间隔，整体车牌的长宽比等车牌尺寸的细节特征如下图2.1的标准车牌尺寸图所示。

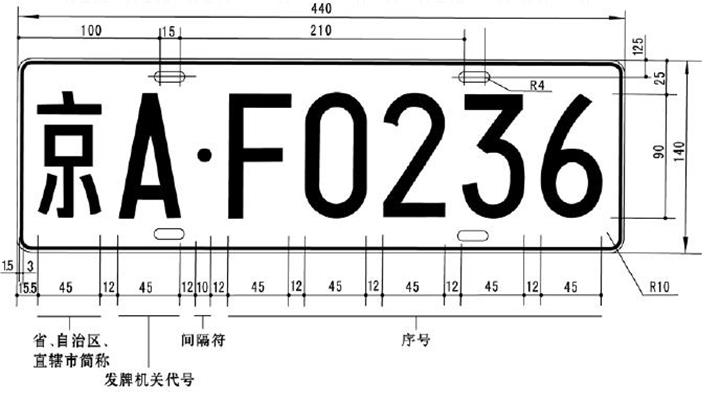


图2.1 标准车牌尺寸图

因拍摄角度，外界的各种干扰因素，实际所获得到的汽车图像的尺寸特征并不会严格如图2.1一般，各标准之间会有一定的小范围内变化。下面大致介绍具体的定位流程。

## 2.2　定位流程

　　定位的大致流程如下图2.2定位流程图所示：



图2.2 定位流程图

　　对图像的预处理部分，首先对输入的汽车图像使用高斯滤波去噪，使图像变得平滑去除部分噪声。再将图像转化为灰度图并进行图像的二值化处理。灰度化就是把摄像头拍摄的RGB类型的图像上每一个像素点的R、G、B三个通道的相同位置像素点的像素值相等。灰度共有256个灰度级别，255是白色，0是黑色。使拍摄到的车牌图像的R、G、B三个通道的像素值相等，即灰度化。在OpenCV的Python接口（以下均默认为Python的接口）中分别对应GaussianBlur()、cvtColor()、threshold()函数。经过预处理后的灰度图大致如图2.3所示



图2.3 灰度化图像

　　再通过各种形态学操作对得到的灰度图像进行进一步的处理，如礼帽操作。先开运算消除图像中的毛刺，后闭运算使图像变为一个整体可分的轮廓。接着进行二值化，将灰度值转换成只有255和0的像素值。二值化过程中，需要选取一个最佳阈值。最佳阈值的选取方法有试错法、Otsu’s算法。本项目使用Otsu’s算法。OpenCV的cv2.threshold() 方法已经实现了这一过程。在寻找最佳阈值时，只需调用OpenCV模块的threshold()方法，此时需要传递参数cv2.THRESH\_OTSU，这个参数的含义是使用Otsu’s算法来计算最佳阈值，并返回最佳阈值。经过预处理后的二值化图像大致如图2.4所示。



图2.4 二值化图像

部分代码如下：

img = cv2.GaussianBlur(img, (5, 5), 0) # 高斯滤波

img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) # 转换成灰度图像

img\_opening = cv2.morphologyEx(img, cv2.MORPH\_OPEN, Matrix) # 开运算

ret, img\_thresh = cv2.threshold(img\_opening, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU) # 二值化图像

　　通过对图像的一些基本的预处理方式可以明显的突出图像的一些基本特征，将图像的明亮与较暗的区域明显的给划分出来，使较明亮的区域变得更加明亮，在灰度图之中即变白，较暗的部分变得更加暗，在灰度图之中呈现更黑。方便之后的进一步操作。

　　接着对处理过后的图像进行边缘检测，在OpenCV中为使用Canny函数。通过边缘检测，能够将能够获取图像的边缘轮廓，再通过开闭运算能够使相近区域的轮廓变为一个整体。所谓开运算，是先进行腐蚀再进行膨胀就叫做开运算，可以用来去除噪音，而闭运算则是先进行膨胀再进行腐蚀操作，可以用来填充前景物体上的小黑点。腐蚀单个操作能简单的去除白噪声也能将两个连在一起的物体给断开来，而膨胀可以用来连接两个分开的物体。因此，通过开闭运算我们能够消除一些噪声的干扰并获得各个部分的填充区域的轮廓图形，大致如图2.5所示。

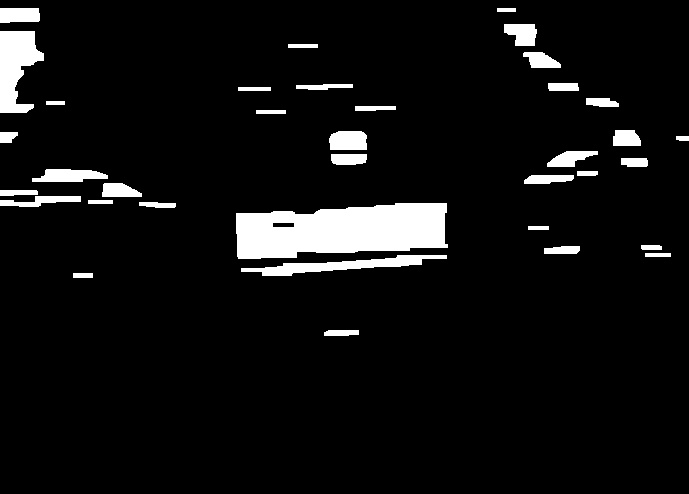


图2.5 轮廓图像

部分代码如下：

img\_edge = cv2.Canny(img\_thresh, 100, 200) # 边缘检测

Matrix= np.ones((4, 19), np.uint8)

# 使用开运算（膨胀再腐蚀）和闭运算（腐蚀再膨胀）让图像边缘成为一个整体

img\_edge1 = cv2.morphologyEx(img\_edge, cv2.MORPH\_CLOSE, kernel)

img\_edge2 = cv2.morphologyEx(img\_edge1, cv2.MORPH\_OPEN, kernel)

　　通过使用findCountours函数将获取的图像为参数输入可获得各个轮廓的边界点坐标与旋转角度。通过我们对车牌尺寸的长宽比（大致为2-5.5左右）以及设定默认最小图形面积信息可以从这些图形之中进一步地筛选出符合条件的图形区域。在获得符合条件的轮廓图形之后，根据之前所获得到的轮廓角度对图形进行矫正。

　　如上图2.5所示，我们所获得的轮廓图形不一定都是相对与图像水平平行的，大致多多少少都有一定的倾斜角度。为了后续操作的方便即顺利进行，我们需要对获得的符合条件的轮廓图形进行一定的倾斜矫正，调用矩形矫正方法img\_Transform()，具体实现过程为：判断变量car\_contours中存储的旋转角度的值，如果介于-1和1之间，将角度变量angle设为1。否则angle中存储的就是之前findCountours函数所收集的轮廓信息中，带有轮廓的倾斜角度的一些信息。首先，我们根据OpenCV中的boxPoint函数，将获得的轮廓作为参数输入可以获得该轮廓的四个顶点坐标数据。根据这四个坐标，选取该轮廓区域图形的左右上下的最值顶点数据。再根据左右最值的顶点坐标，我们可以根据左右的坐标Y的差值，可以判断出该区域图像是向左倾斜或是向右倾斜。经过图2.4观察，可以发现，虽然车牌本身有一定的倾斜角度，但是其字符本身并没有受到太大的干扰，也就是字符本身并没有受到倾斜的影响。因此，我们选则使用warpAffine函数，传入要变换的图像oldimg，仿射变换矩阵M，变换后的图像大小(pic\_width, pic\_hight)三个参数，实现对车牌图像的矩形矫正。选择使用平移原始图像操作而不是使用旋转原始图像的操作来避免一些对车牌字符的干扰，通过传入原始图像的左右与上方顶点坐标与新的左右与上方顶点坐标参数实现。矫正后的矫正图像大致如图2.6所示。



图2.6 矫正图像

部分代码如下：

car\_rect = (car\_rect[0], (car\_rect[1][0] + 5, car\_rect[1][1] + 5), angle)

if left\_point[1] <= right\_point[1]: # 正角度，根据左右顶点判断

new\_right\_point = [right\_point[0], heigth\_point[1]]

pts2 = np.float32([left\_point, heigth\_point, new\_right\_point]) # 字符只是高度需要改变

pts1 = np.float32([left\_point, heigth\_point, right\_point])

elif left\_point[1] > right\_point[1]: # 负角度，根据左右顶点判断

new\_left\_point = [left\_point[0], heigth\_point[1]]

pts2 = np.float32([new\_left\_point, heigth\_point, right\_point]) # 字符只是高度需要改变

pts1 = np.float32([left\_point, heigth\_point, right\_point])

# 仿射变换

M = cv2.getAffineTransform(pts1, pts2)

dst = cv2.warpAffine(oldimg, M, (pic\_width, pic\_height)) # 平移图像

　　矫正之后根据之前的各个符合条件的轮廓坐标信息从矫正后的图像单独截取出该部分。之后将截取出的图像转换为HSV模式，统计颜色信息。

　　HSV色系即代表色调（H），饱和度（S）,亮度（V）。能够更直观的表示图像颜色信息。通过多次上述操作，发现蓝色车牌的色度H的值的范围大致在100-130之间，饱和度S的值的范围在110-255之间，亮度V的值的范围在105-255之间。故inRange()方法中参数设置为np.array([100, 110, 105])和np.array([130, 255, 255])的效果最好。黄色车牌、绿色车牌的识别过程中的inRange()方法中的参数也是这种方法获得的。使用inRange()方法获得车牌对应的蓝色掩码图、黄色掩码图、绿色掩码图，无论拍摄到的车牌图像是蓝色、黄色或绿色车牌，都调用bitwise\_and()方法将原图与三色掩码图mask逐位相与，实现了车牌通过颜色定位的归一化，具体实现为：cv2.bitwise\_and(hsv, hsv, mask=mask\_blue + mask\_yellow + mask\_green)。接着使用轮廓查询方法找到图像中存在的所有的外部轮廓，使用生成最小外接矩形法计算出所得到的所有的外部轮廓的最小外接矩形，存储下来，使用面积和长宽比阈值，作最后的验证，筛选出车牌区域，并将车牌区域信息存储下来，从而实现车牌的定位功能并按照矫正后的车牌位置信息调用img\_color()方法，得到车牌的颜色字段和车牌位置信息。如图2.7所示。



图2.7 定位区域图

部分代码如下：

width, height = ant[1]

if width < height:

width, height = height, width

ration = width / height

if 2 < ration < 5.5:

car\_contours.append(ant)

box = cv2.boxPoints(ant) # 获得要绘制这个矩形的 4 个角点

# 第三章　　字符分割

## 3.1　切割方式

　　在获得比较精确的车牌定位区域后，开始进一步对获得到的图像进行字符分割。车牌的字符分割主要的作用是为了方便之后所要实现的识别功能。在准确的分割算法下,车牌识别能够获得较高的准确率,但分割算法要达到较高的准确率是比较困难的,其表现结果受光线强度、车牌扭曲等的影响较大,停车场的应用场景下车辆往往可以保持良好的“姿态”且距离采集设备较近,因此采用字符分割的方法可以获得优秀的识别效果,但对于无人机摄像头或交通摄像头来说环境会更加复杂[8]。

　　现在，最常见最常用的分割方法为两种，一种是基于连通域的分割方法，另一种是基于车牌的投影图像的分割方法通俗来说即是垂直投影法。本文所采用的方法则是基于垂直投影的方法来进行。

垂直投影法是目前应用较多的字符分割的方法，其特点为比较简单且相对要容易实现。它是根据图像在垂直方向上的投影，（在车牌字符分割中，理想状态下只有在有字符的位置上垂直投影是有数值的，而字符与字符之间的间隙是为0的）车牌之间的字符是由一段间距存在的，因此根据投影之间的数值，可以判断出字符与字符之间所在的间隔位置，依此可以将字符之间给分割出来。在本文，分割方法的大致的实现步骤为，首先对车牌二值化图像进行逐行扫描，记录其数值的跳变次数。因为车牌之间的间距存在，车牌二值化图像的数值横向跳变次数基本是保持在一定的次数以上（因噪声的存在，图像预处理的不完善，字符有一定几率存在粘连等多种因素无法保证为固定数值）可以大致的确定上下边界。之后再对图像进行逐列的扫描，分别记录下黑白像素的数量值。根据数量值，我们可以设定一个较为宽松的阈值，通过修改阈值可解决一些字符粘连上的问题。

## 3.2　切割流程

　　在使用垂直投影法进行车牌字符分割操作的流程大致如下流程图3.1所示。



图3.1 分割流程图

　　首先是图像的预处理部分。同车牌定位流程部分的预处理大致相同，先是使用高斯滤波平滑图像去除噪声，再将图像转化为灰度图并二值化处理，需要注意的则是，与蓝色作为底色的车牌不同，绿色和黄色为底色的车牌字符颜色较暗而背景较亮，与蓝色车牌正好相反，直接对转化好的灰度图像进行二值化处理会使得背景部分为白色而字符部分会呈现黑色，并且通常带有大量的干扰因素不利于后面的处理。所以，根据之前车牌定位所得到的车牌颜色，若为黄色或者绿色，首先使用OpenCV中的bitwise\_not函数对灰度图的数值先进行取反。大致效果图如下图3.2所示。



图3.2 车牌二值化图

部分代码如下：

# 黄、绿车牌字符比背景暗、与蓝车牌刚好相反，所以黄、绿车牌需要反向

if color == "green" or color == "yellow":

gray\_img = cv2.bitwise\_not(gray\_img)

ret, gray\_img = cv2.threshold(gray\_img, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)

　　要正确高效率的识别车牌字符，最重要的是对定位到的车牌区域上的中文、英文及数字的准确分割。对中文、英文及数字的分割的目的是把经过上述操作定位到的车牌区域中的每个字符依照次序的从大图像中分割出只包含单个字符的小图像，对分割下来的包含单个字符的图像依照分割的顺序进行字符识别。分割得到的车牌字符与分割得到的车牌底色之间的灰度差异明显，所以基于投影的对中文、英文及数字的分割方法比较常见。系统使用的对中文、英文及数字的分割算法是基于投影的字符分割法。在二值化过程中计算每一列上目标像素的数目，得到表现为一系列波峰和波谷的车牌图像的垂直投影曲线。波谷对应着对中文、英文及数字的分割的位置，而波峰的位置对应着相应的字符。分割算法首先构建直方图，调用波峰查找方法find\_wave()找出波峰，对波峰过滤，去除车牌上的分隔点。对于find\_wave()中阈值和直方图的选取，直方图由np.sum(gray\_img, axis=0)得到，其中gray\_img是去除白边的蓝色车牌区域，阈值由y\_min = np.min(y\_histogram)、y\_average = np.sum(y\_histogram) / y\_histogram.shape[0]、y\_threshold = (y\_min + y\_average) / 5三条语句获得，除5操作是由于类似U、0这类字符的分割操作要求阈值偏小，否则会造成分割不完整。接着调用分割字符的seperate\_card()方法，分割图片，从而得到逐个字符图片。然后去除车牌上的铆钉的干扰，使用常量扩充边缘的方法copyMakeBorder()，将分割后的字符使用常量扩充边界，如下图为不扩充边界的字符区域。



图3.3 不扩充边界的字符区域

　　若不使用常量扩充边界的方法，会产生错识或者多识。接着使用重写后的resize()方法将识别后的图像调整为固定格式。分割结果如下图3.4所示。

图3.4 分割结果图

部分代码如下：

wave\_peaks = img\_math.find\_waves(x\_threshold, x\_histogram)

part\_cards = img\_math.seperate\_card(gray\_img, wave\_peaks)

　　自此，车牌分割到此基本完成，基于垂直投影的分割法可以基本达到了我们所预期的分割效果，成功的将车牌区域的字符给分割了出来。但是，这个方法在一定程度上有着一些不足之处，首先，对初始图像的质量要求要高一些，汽车图像本身的模糊，字符褪色问题，或者一些其他的外在干扰信息都可能会对分割效果造成影响；其次，对于图像的预处理，干扰噪声的去除要求也会相对高一些，若存在较多的干扰条纹，就会对投影结果产生影响，进一步则会影响到分割效果。对于前一点我们无法做出修正，因此对后一点需要稍微着重处理。

# 第四章　　字符识别

## 4.1　字符识别方式

　　将车牌定位与车牌字符分割说做车牌识别整体的基石，那么字符识别的性能好坏也是直接关系到了整体车牌识别的成果。由于计算机的整体普及、计算能力的进一步发展以及大数据智能化的时代到来，字符识别的方法变得多种多样，各个方法之间得优劣各有千秋。例如基于字符特征的方法，基于模板匹配的方法、基于神经网络的识别方法与支持向量机法等等。基于车牌的特征统计方法是目前在车牌识别的应用中使用也是比较广泛的一种，这种方法主要是依靠在统计学之上的一种识别方法。在之前的车牌定位中，我们获得到了车牌的具体区域，在字符切割中得到了车牌的单个字符图像。最后，在本文中，根据现有的资源以及OpenCV中附带有的函数功能，本文中采取使用支持向量机的方法进行字符识别。

　　支持向量机，也被称为SVM，是一种能够构造出线性和非线性分类器的方法。作为一种二分类模型时，它的基本做法是求解出能够正确地划分训练数据集并且使几何间隔最大的分离决策面，此外SVM包括一种核技巧，可以通过核方法使它达到非线性分类器的效果。简单来说，SVM方法是基于结构风险最小化原理的统计学方法[9]。SVM分类器的优点在于它能够相对缩短时间复杂度，只需要进行一次模型的训练，对模型可以重复地进行使用。虽然这种方法有着许多优势，却也有着缺点存在，为了提高识别精度，需要大量的训练集当作样本，训练的时间以及对计算机本身的要求较大。

## 4.2　识别流程

　　在OpenCV中提供了相应的训练的sample样例函数，仅仅是使用的话非常方便。OpenCV所附带的sample函数的本质为将灰度图、二值图、16值图的HOG特征在一定的权重下融合为联合HOG特征，使用核主成分分析法（KPCA）对联合HOG特征进行降维，进而利用支持向量机进行分类[10]。若需要训练模型，只需要通过cv2.ml.SVM\_create()函数创建SVM类，将所需要训练的图片通过OpenCV中附带的sample样例转化为相应的格式后，再将转化好的训练样本图片以及图片的标签传入所创建的SVM类中的.train()函数之中即可。最后在训练完成之后，将训练好的模型保存至本地即可。由于设备以及无法获得足够的训练样本集，因此本文采用已经训练完成的sample模型作为字符识别模块使用。大致的流程如下图4.1所示。



图4.1 识别流程图

　　在图片预处理方面在舍弃掉上下黑边之后还需要做两点，首先将每个字符图像的四个边界可以缩小一个像素或者将其像素值给置0，因为分割出来的字符四周的边界也还是有可以能会存在干扰，处理过后若能消除影响则能进一步提高识别的准确率。此外，需要每个字符图片统一归一化为通过sample训练时的大小格式，也是有助于在之后调用时提高识别的准确率。在此例中格式为20\*20。此外在改变图像尺寸的时候，图像本身是会变得一定的模糊化。经过一些简单的测试，这种一定程度上的模糊化的效果是有助于消除干扰效果的影响的，也是对识别率有一定帮助，因此可以直接以此形式直接进行之后的操作。

　　获得处理好的图像后，创建SVM类实例，读取本地或者实时训练好的模型，将图像经过OpenCV的sample样例函数处理，最后调用实例的.predict()函数即可获得识别结果。

部分代码如下所示：

# modelchinese与model分别为获得到的汉字模型与英文数字模型

part\_card = img\_recognition.preprocess\_hog([part\_card]) # 将图片传入样例进行函数转换

if i == 0: # 第一个字符为汉字

resp = self.modelchinese.predict(card)

ans.append(provinces[int(resp[0]) - PROVINCE\_START])

else: # 其余为英文或数字

resp = self.model.predict(card)

ans.append(chr(resp[0]))

# 判断最后一个数是否是车牌边缘，假设车牌边缘被认为是1

if charactor == "1" and i == len(part\_cards) - 1:

if part\_card\_old.shape[0] / part\_card\_old.shape[1] >= 7: # 1太细，认为是边缘

continue

　　本例中识别结果如下表4-1所示。

表4-1 识别测试表

|  |  |
| --- | --- |
| 测试车牌图 | 识别结果 |
|  | blue | 苏DS0000 |
|  | blue | 冀T88888 |
|  | blue | 吉AA266G |
|  | blue | 云A555TX |
|  | blue | 渝AN7968 |
|  | blue | 京KP1111 |
|  | blue | 苏EC62N8 |
|  | blue | 豫M00000 |
|  | blue | 苏BU129L |
|  | yellow | 苏D11111 |
|  | yellow | 川A00001 |
|  | green | 鲁AD25557 |
|  | green | 苏BE01111 |
|  | green | 京AD77972 |

可以看出，识别的成功率基本是令人满意的，但是在汉字的识别上却有些令人遗憾的出错存在。分析后其主要原因可能在于两点，第一个原因是因为训练模型的原因。汉字的训练集本身不好寻找，不同于英文汉字在全世界范围内的流行，中文的汉字基本是我国所特有的文化，所以训练集所训练出来的模型本身的识别率有可能不算很高。第二点则是对图像的处理本身问题，从图3.4中可以看到，虽然汉字的字符图像周围虽然噪声消除的比较好，但是汉字的自身的特征也被处理而导致信息损失过多，从而导致了汉字的识别率会整体有所下均。

# 第五章　　总结与展望

## 5.1　总结

本文所设计的车牌识别系统是在基于OpenCV为主进行开发的设计，尽力的运用了Python语言于OpenCV本身所提供的便捷性。汽车车牌的识别可谓是当今智能化交通发展道路上的基础技术，在本文所设计的一个简单的汽车车牌识别系统可以借由如今的各路计算机软硬件技术，为一些软硬件，如自动车辆门禁管理系统、高速公路的车辆管理、智能交通识别管理等主要在依靠汽车车牌进行车辆管理的场所提供一些基本的设计思路于解决方法。本文对车牌识别系统的背景、意义及现今的一些国内外研究现状的一些叙述表示，车牌识别在我国依然是拥有一定的意义存在。对于车牌识别的三部分重点，车牌定位、车牌字符分割、字符识别通过对车牌的尺寸特征、颜色特征、基于垂直投影的方法、SVM的识别方法，分别进行了一定程度上的概述并给出了一个较为简单的实现方法或思路，确定了本系统的基本完成。

## 5.2　展望

虽然本文所叙述的汽车车牌识别系统设计基本成功，其成果也基本达到了预期要求，但是可以发现，因本人的知识水平有限以及设计不充分，本文的方法仍有着许多不足之处存在。如多车牌的应用场景没有设想多，在今后的交通发展，多车牌的场景必然频繁出现，其次目前只能识别黄、绿、蓝三种种类的车牌，而在我国肯定不只有这几种分类等多方面需要进行进一步改进。

参考文献

[1]基于图像的车牌识别系统设计与实现，作者：唐智强，电子科技大学 2018

[2]车牌识别系统的研究与实现，作者：管文，湖北工业大学 2018

[3]基于深度学习的车牌识别系统研究，作者：张佑贤，广西师范大学 2019

[4]胡雪会，王波涛．基于OpenCV的车牌图像定位[J]．计算机测量与控制，2016，24(09)：206-210．

[5]黄社阳，刘智勇，阮太元．基于HSV颜色空间和SVM的车牌提取算法[J]．计算机系统应用，2014，23(08)：150-153．

[6]甘玲，林小晶．基于连通域提取的车牌字符分割算法[J]．计算机仿真，2011，28(04)：336-339．

[7]邓运生，郑晨霞，尹安．车牌定位和字符分割方法对比研究及实现[J]．兰州文理学院学报(自然科学版) ，2019，33(06)：78-83+106．

[8]实时车牌识别系统的设计与实现，作者：刘洁，山东大学 2019

[9]余棉水，黎绍发，基于边缘与SVM的车牌自动定位与提取[J] ．计算机应用研究，2004(10)：131-133．

[10]殷羽，郑宏，高婷婷等．基于联合HOG特征的车牌识别算法[J]．计算机工程与设计，2015(2)：476-481．