

## 基于 python OpenCV 车牌识别计费系统

摘要：随着我国日渐强盛，交通问题也越来越大，人们需要一款高科技的可靠性高的车牌识别系统。本文即是针

对这一问题开发了一款基于 python OpenCV 车牌识别计费系统，本系统的图像处理主要来源于计算机视觉库 OpenCV，采用了 python 语言来对我国常见车牌进行识别。该识别系统通过对车牌的文本图像信息进行了车牌图像信息预处理、车牌图像定位、车牌字符图像分割和车牌文本字符识别等多种图像处理方式，从而基本实现了汽车车牌识别系统中的各种重要功能。根据实验结果可以看出，该系统可以识别我国常见的大部分车牌。

关键词：OpenCV，图像的文字预处理，车牌号的定位，图像的字符分割，图像的字符识别

### 1 绪论

#### 1.1 研究背景及意义

##### 1.1.1 课题研究背景

随着当前阶段我国新型现代城镇化工程建设工作进程的深入和不断加速，市场经济和国家工业技术现代化等社会科学信息技术的广泛引入和应用迅猛发展，人们对于自己汽车出行的需要日益扩大，机动车已经逐渐地成为现代人们在日常生活中必须使用的一种重要交通方式。截至 2021 年第一季度，全国累计发放新增驾驶注册机动车 966 万辆，同比上一季度上一季度多了 67.31%，比 2019 年同期上一季度多了 31.96%，创同期的历史最高水平。机动车虽然给人们的日常生活方式带来极大的方便，但也在其中滋生了一个社会性的问题。即是说，机动车虽然在给人们的工作和日常生活都带来极大的方便，但也在其中滋生了一个社会性的问题。也就是说，私有机动车辆总量的迅猛增长直接导致了我国许多大城市和中小型农村地区的机动车停放困难问题越来越凸显，而且机动车辆停放困难的问题也成为长期困扰人们的社会问题，给当前我国的交通运输和管理工作带来了极大的压力。要想完全解决机动车停放困难的问题，不能再简单地用一个停车场有偿使用或者是加大对停车场基础设施工程的建设投资来进行调整。所以这些都不是很好的解决方法。而是需要充分运用以人民为中心的发展理念的现代科学技术，利用先进的现代科学技术来构造高效的停车场计时和计费系统，提高停车场的计时和计费工作效率，实现整个停车场的智能化。此类问题的管理工作已经是政府、运输行政主管部门、物流行政主管部门和道路交通行政执法机构的一项重要工作。

基于 python opencv 车牌识别计费系统，本研究课题主要来基于对停车场的安保管理措施薄弱、计时和计费不合理、停车资料保存不健全，且与当代信息自动化、智能化等城市建设的推进方向不适应和协调，影响了整个城市温暖和力量的品牌形象等众多问题而提出的。本套系统设计旨在开发出一个具有较高技术含量、成本较低、维修简单、计时和收费准确等优势。

##### 1.1.2 车牌识别技术国内外发展现状

车牌识别技术最早起源于上世纪 80 年代,相关专家学者当时针对车牌识别技术的研究只是将其范围仅仅停留在对于汽车牌号图像本身的处理与分析上,该技术发展阶段的汽车牌号识别技术尚未完全形成任何科学研究的体系。随后我们步入上世纪 90 年代,源自电子计算机和视觉技术对汽车快速发展的驱动和推动,美、日、德、法等地区和国家都投入了巨大的资金、财力对汽车车牌识别技术的应用进行了研究,经过努力,车牌识别技术的科学研究框架逐渐建立和完善,最近几年以来我国车牌识别在这些地区和国家的应用研究已相当成熟。其中我国目前具有重要代表性的技术包括:基于车辆颜色特征和边缘检测的数字化车牌辨认算法正在得到应用;发现字符与背景之间颜色的不同或者是色差也都可以从而在车牌识别方面得到应用,拓宽了我们的车牌识别理论研究视野;介绍了一种新型的车牌识别系统,该模块在其中利用 opencv 提取的车牌特性和对其他车牌信息进行了二值化的操作,让该模块系统在实际应用中对于车牌的识别效率有了很大的改善。在复杂的环境中如何进行汽车牌号辨认的研究[7],是的车牌识别及时有了更加广阔的应用范围。

由于智能交通的推广、智慧城市的建立以及人民对停车场服务质量的要求日益提高,我国现代化停车场早已朝着科技化和自动化发展,停车场的各项技术在国内发展快速,越来越多的公司将把车牌识别技术专注运用于到停车场的计时计费系统当中,在技术和市场占有率上做到领先的公司有:川大智胜、成都臻识、北京汉王、北京文通等等。

### 1.1.3 课题研究意义

基于前面提到的有关于停车和交通方面的问题,本研究课题主要目的是基于自动化的车牌辨认技术来研究设计一种用于停车位的计时和算费系统,其主要的优点有以下几方面:

(1)系统收费的规范合理:严格地设计规范了系统收费的具体内容,实现了自动系统收费,自动信息查询等收费功能,解决了系统收费的信息流失,以及系统收费的不合理等潜在的收费问题。

(2)有效地节约了时间:通过改变 ic 卡车主在自动取卡和插入手机的过程中浪费了时间的情况,节约了其他车主宝贵的资金。

(3)停车场的安全性高:具备数据库和存储管理功能、便于日后检索和查询机动车在该地区停放的情况和记录,保证了停车场的安全。

(4)施工经济性高:由于停车场的软硬件投入比较低,停车场的投资者仅仅是根据自己所需的智能路口、摄像头、聚光灯以及电脑等设备,降低了建筑物的施工困难,节约了建筑物的资源,具有相对优势和经济效益。

因为车牌号是机动车唯一具有不能更改的标志性编号特征,所以车牌标志性编号特征有时候会犹如也就是当地居民“身份证”的车牌编号,构建高效合理、智能化的汽车计时和运行计费管理信息综合管理服务系统对于汽车的车牌标志性特征编号建设应用来说也是必不可少的。

## 1.2 车牌识别技术的应用

车牌自动识别系统技术主要目的是为了能够实现定位车辆自动识别确认需要定位的所有车牌号和地点,能够自动识别进行对定位车牌上的所有字符符号进行自动分割和识别辨认。该项创新技术主要研究应用领域范围涉及包括光学电子信息科学技术、光学电子工程技术、图像信号处理、计算机工程视觉、自动化等多个专业领域,在许多实际应用领域中都对其有着重要指导作用:

该系统主要应用于停车场对于车辆的识别与计费除此之外还可以在高速公路和日常交通管理上发挥作用。

### 1.3 车牌识别技术难点

车牌识别的最大一个特点便是在人为环境和自然条件等特点下对采集得到的各种汽车牌照信息进行了识别,车牌辨认技术中的主要难点即其重要性就是在如何高效地快速识别得出车牌,车牌辨认的精确性和速率主要受多种因素的影响,难点分析方法如下:

(1)汉字、文本、字母、数码的混合:只有我们知道第一个字符为汉文、第二个字符为文本、后五个字符为数码、文本、字母等的随机组合。

(2)汽车颜色变化多:目前我国常见的汽车牌号和颜色变化多。

(3)由于人为管理原因的复杂:往往有一些交通行政执法机关为了完全逃脱对道路交通安全法律运行规则的严厉惩戒,故意地严重污损了普通车牌、掩盖普通车牌、不当地按规定标准地禁止悬挂普通车牌。

(4)各种车牌形态和格式多:许多专门的职能部门都具有自己独特的车牌形态。比如:私人汽车、消防用的汽车、大使馆的汽车牌、军事和机动车的汽车牌等。

(5)车牌悬挂形式的多元化:重型汽车、大客车、小轿跑、农用汽车等各类型的不同车牌悬挂形式都是千差万别的。外部环境影响:

(1)对汽车光照环境条件的直接影响:在白天与夜间黑夜、晴天、雨天或者说是阴天的任何时候,光照环境条件都有可能完全是不同的,光照环境条件的发生改变也可能会直接性地影响你看到一辆车牌灯光信号信息识别的速度准确率、辨认车的速度,光照环境条件也可能是重要的光照测量检验指标之一。

(2)对监控摄像机中的信号进行采集和检测角度的控制问题:在实际的监控工程当中,一般来说会把一个摄像机信号设置在一辆机动车或汽车的正前方或者也就是正上方的左侧或者右侧。由于所有用监控摄像机对其进行信息采集处理得到的标志图像都必须是经过动态准确地进行采集处理得到的一整套系列化的图像,而且用一辆车牌上的标志图像信息系统对其图片进行图像处理后的采集图像效果并非一定量就能够完全达到最理想的图像效果。

### 1.4 论文安排

#### 1.4.1 系统框图

本文的研究课题主要对对我国汽车专用牌照的文字图像信息预处理与对比及车牌号码的准确定位、文字图像信息源的分割、文字信息识别、车型趋势判断、计时和自动算法收费等相关系统基础设计几个重要工作过程的进行深入分析研究,其系统基础设计工作完成流程以及框架设计如软件图 1.1 所示。

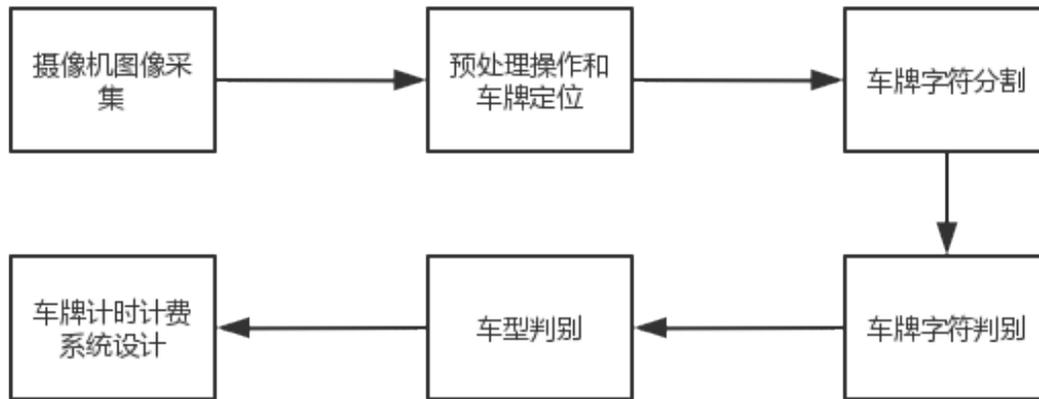


图 1.1 本课题的基本流程

#### 1.4.2 本文的章节安排

本文的研究课题主要内容是详细介绍了一款基于停车自动化的通用车牌计时识别管理技术的新型公共停车场自动车牌计时和停车收费服务管理信息系统,总计六十个章节的研究内容,具体各研究章节的课题安排内容如下:

第一章为绪论。主要阐述了这个课题的择优选项背景,研究意义,贡献的价值。介绍了我国汽车牌号自动识别技术在国内外的研究发展现状、存在的困难及其具体应用方面,最后详细介绍了这篇论文的基础理论内容及对全文章节的编辑安排。

第二章主要是介绍了图像预处理的操作、分析了比较传统特点的车牌定位算法,提出了本文所采用的车牌边缘特点提取与数学形态科技技术相结合的车牌定位算法,完成了车牌定位的目标。

第三章主要是介绍了各种车牌字符的应用特点、常见的各种车牌字符分割技术和方法、提出了本文所需要使用的垂直投影相结合的车牌字符先验基础知识的各种车牌字符分割技术和方法;完成了对车牌号和字符进行分割。

第四章主要内容是详细介绍了归一化的基本原理,分析和应用比较了常见的字符识别系统方法,考虑了软件系统对于字符识别系统实时性的基本需求,利用系统模板字符匹配识别方法等进行字符识别。

第五章节是系统设计的一个组成部分,介绍了该系统使用的各种开发工具、环境等,介绍了如何使用数据库进行交互的技术和方法,分析了该系统所需的功能进行了调试和实现。

最后阶段主要是对这个课题中的全部工作内容进行了总结和分析,以及在接下来关于该项目的改善、升级、精简等方面的工作提出了展望。

### 1.5 本章小结

本章是全文的绪论部分,介绍了基于 Python OpenCV 的计时计费系统的研究的背景、意义、研究状况,本章还对全文的写作思路和章节安排进行了介绍。

## 2. 车牌图像预处理和车牌图像准确定位的设计与实现

车牌预处理的目的是为了接下来的车牌定位的操作提供良好的基础,消除后期不必要的麻烦。本文主要研究的车牌为社会中常见的蓝、绿、黄三种颜色的车牌图像,并不包含特殊车辆,大多数情况下由视频输入或图像输入的照片为彩色 RGB 图像,为方便程序进行处理要先将图像转换为单通道图像,这就是图像的预处理。图像预处理的步骤主要先获取图像的 HSV 通道图像,再将图片的 HSV 通道图像通过高斯滤波,图像边缘检测,图像二值化以及图像形态学处理等步骤转化为灰度图像。

### 2.1 HSV 通道图像的获取

首先,为了方便程序对图片的处理,我们要先将彩色图像从 RGB 通道转化到 HSV 通道 (H 是图像色调 (Hue), S 是图像饱和度 (Saturation), V 是图像亮度 (Value))。

$$h = \begin{cases} 0^\circ, & \max = \min \\ 60^\circ \times \frac{g-b}{\max-\min} + 0^\circ, & \max = r, g \geq b \\ 60^\circ \times \frac{g-b}{\max-\min} + 360^\circ, & \max = r, g < b \\ 60^\circ \times \frac{b-r}{\max-\min} + 120^\circ, & \max = g \\ 60^\circ \times \frac{r-b}{\max-\min} + 240^\circ, & \max = b \\ \cdot & \cdot \\ & \cdot 360 \end{cases}$$

以下是图像的处理方法: 设 (r, g, b) 为某一像素的坐标值, r、g、b 的值是 0 到 1 之间的实数。设 r、g 和 b 中的最大值为 max, 最小值为 min。在公式 2.1 中,  $h \in [0, 360)$  表示色相的角度,  $s, v \in [0, 1]$  分别表示图像的饱和度和亮度。

### 2.1

h 的取值在 [0 ] 之间, 当 r, g, b 的最大值最小值相等即  $\max=\min, h=0$  时, s 和 v 的表达式如公式

2.2, 2.3:

$$s = \begin{cases} 0, & \max = \min \\ \frac{\max-\min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

v = max 2.3

2.2

$$\begin{cases} \max \\ \min \end{cases}$$

经过设置，图像的颜色阈值设置如表 2.1 所示：

阈值	黄色	蓝色	绿色
阈值上限	[50, 255, 255]	[130, 255, 255]	[70, 255, 255]
阈值下限	[15, 55, 55]	[110, 100, 100]	[70, 100, 100]

表 2.1 图像黄色，蓝色，绿色的阈值设置

图片由 RGB 通道图像转化为 HSV 通道图像结果如图，其中图 2.1 为原 RGB 通道图像，图 2.2 位转化后的 HSV 通道图像：



图 2.1 原 RGB 通道图像 图 2.2 转化后 HSV 通道图像

从图 2.1 和图 2.2 的对比中可以知道，图像的转换过程比较成功，基本上实现了图像的颜色分割，而且这一步也除去了大部分的复杂背景以及噪声点，为下一步对图像的灰度处理打下了比较好的基础。

## 2.2 对 HSV 通道图像的灰度处理

为了能够使系统对图像处理更加方便也更加准确，还要将由 RGB 通道图像转化来的 HSV 通道图像进一步的转化为更简单的单通道的灰度图像，单通道的灰度图像每个像素点所包含的信息更少，更加便于系统对图像进行处理，这样不仅可以加快系统的工作效率，还能提高系统的准确率，使系统更加的可靠，图 2.3 为原图处理后的灰度图像：



图 3 原图处理后的灰度图像

### 2.2.1 高斯滤波

高斯滤波的目的是使图像像素的灰度值更加的平滑，高斯滤波原理简单来说就是将图像中的每一个像素都取其相邻像素的加权平均值，这样就能达到噪声的效果。这样处理后的图片系统更加容易处理，可以提高系统的工作效率降低系统的错误率，提高系统可信度。执行高斯滤波的示意图如图 2.4 和 2.5 所示，方格中的数字表示像素值，图

2.4 为图像高斯滤波平滑前的像素值，图 2.5 为图像高斯滤波平滑后的像素值。

1	1	1
1	2	1
1	1	1

图 2.4 图像高斯滤波平滑前的像素值

1	1	1
1	1	1
1	1	1

图 2.5 图像高斯滤波平滑后的像素值

在本文中我使用的是 Python—OpenCV 中的函数——cv2. GaussianBlur() 对图片进行高斯平滑。平滑前后的图像如图 2.6 和 2.7 所示，图 2.6 为使用高斯平滑前的图像，图 2.7 为使用高斯平滑后的图像。



图 2.6 使用高斯平滑前的图像

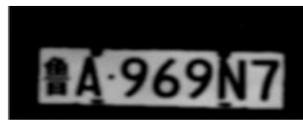


图 2.7 使用高斯平滑后的图像

### 2.2.2 边缘检测

我国车牌上的字符和车牌的底色之间是不同的，而图像的灰度值会在图像颜色交接的地方发生跳变，跳变的方式可粗略分为阶跃变化和屋顶状变化如图 2.8 和图 2.9 所示，边缘检测就会将这种发生的特别明显的跳变现象标记出来，这样就方便了系统对图像的预处理。



图 2.8 阶跃变化图 图 2.9 屋顶状变化

本未使用的是索贝尔图像边缘检测算子，索贝尔图像边缘检测算子在实际生产生活中的应用十分广泛，该算子是通过卷积核  $g_1(x, y)$ 、

卷积核  $g_2(x, y)$ 、对原始图像  $f(x, y)$  卷积得到的，其数学表达式如 (2.8) 所示： $S(x, y) = \text{MAX}$

$$\left[ \begin{array}{l} \sum_{m=1}^M \sum_{n=1}^N f(m, n) g_1(i-m, j-n), \\ \sum_{m=1}^M \sum_{n=1}^N f(m, n) g_2(i-m, j-n) \end{array} \right] \quad (2.8)$$

索贝尔算子是  $3 \times 3$  的模板，该算子是通过加权平均，而且索贝尔算子是有方向的，可分为垂直算子

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \text{和水平算子} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

两个算子，这两个算子可以在两个方向形成强烈的边缘，即分别在水平方向和垂直方向形成了强烈的边缘。索贝尔算子采用了差分、滤波运算，所以该算子既可以检测边缘点，又能去除噪声对边缘检测的干扰，效果较好。其数学表达式如 (2.9)-(2.11) 所示：

$$S_1(i, j) = f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1) - f(i+1, j-1) - 2f(i+1, j) - f(i+1, j+1) \quad (2.9)$$

$$S_2(i, j) = f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1) - f(i-1, j-1) - 2f(i, j-1) - f(i+1, j-1) \quad (2.10)$$

$$S(i, j) = |S_1(i, j)| + |S_2(i, j)| \quad (2.11)$$

由于索贝尔算子的边缘检测具有定位精度高处理速度快的好优点，所以本文之所以采用索贝尔算子，是因为由于索贝尔算子的边缘检测与其他算子比较具有定位精度高处理速度快的好优点，即函数 `cv2.Sobel()` 进行边缘检测。得到灰度边缘检测的图像如图 2.10 所示：



图 2.10 灰度边缘检测的图像

### 2.2.3 图像的二值化处理

为了进一步的精简图像像素的信息，本文将混度处理后的图像惊醒了二值化处理。进行二值化处理的灰度图像信息更加精简，系统的工作效率和准确性也将进一步得到提升。

灰度图像二值化处理方法如公式 2.12 所示：

$$g(x,y) = \begin{cases} 0, & g(x,y) < T \\ 255, & g(x,y) \geq T \end{cases} \quad (2.12)$$

经过测试，将阈值设置在 80 的时候，灰度图像二值化效果最好，二值化结果如图 2.11:

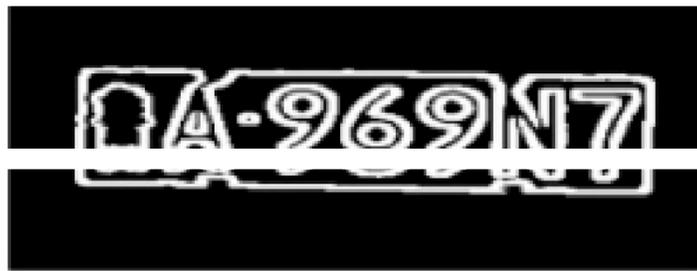


图 2.11 灰度图像二值化效果图

#### 2.2.4 形态学处理

形态学处理中的最基本操作就是腐蚀与膨胀。在二值化图像中，使用腐蚀操作后，图像的边缘界限会因此变小，像素带也会因此变窄，这样就能把那些比较小的像素块腐蚀掉，避免对之后图像处理的影响。使用了膨胀这个动画操作后，图片当中一个物体的边缘界限可能会向外扩展，即使这个物体的边缘界限仍然存在着严重的缺损，这些缺损的边界部分也会被像素补充。具体的操作步骤如下：

膨胀:用一个元素给定的物体结构图将元素物体 b 对于在目标点的物体所在 a 的每个点像素进行快速平移,将在物体 a 某一点的每个像素快速增加和高度扩展的平移操作过程变成了一个膨胀的平移运算。记:

$$a \oplus b = \{(x,y) | b_{(xy)} \cap a \neq \emptyset\} \quad (2.13)$$

腐蚀:用到的结构点是元 b 需要平移一个靠近目标点的物体 b 和 a 中某一点(x, y)后,满足需要平移这个条件的(x, y) 地点的结构点是元素物体 b 的一个最大参考点(通常来说是一个原点)和其中一个与靠近目标点的物体 b 和 a 的最大有关点的集合,记作:

$$a \ominus b = \{(x,y) | b_{(xy)} \cap a^c \neq \emptyset\} \quad (2.14)$$

因为膨胀和腐蚀这两种操作是可逆的，所以通过他们两种操作的不同组合，可以将它们分为两种，即“开启”和“闭合”的概念。先腐蚀再膨胀这就是“开启”，与之相反的先膨胀再腐蚀就是“闭合”，数学表达式分别为

$$A \circ B = (A \ominus B \oplus B) \quad (2.15)$$

$$A \cdot B = (A \oplus B \ominus B) \quad (2.16)$$

扩大目标对象像素是膨胀操作的作用，收缩目标对象像素是腐蚀操作的作用，结构元素的大小和形状决定了扩大和收缩像素数量的多少。以为知道了本文中的图片都是形状规整的矩形，在数学形态

学形成连通域的操作过程中，首先要用水平方向上的  $2 \times 6$  结构元素来对车牌进行膨胀运算，然后是要消除一部分连通域中的一些小缝隙间隔，这里用到的是用一次闭运算，得到的效果图如图 2.12 所示。



图 2.12 灰度图像形态学处理效果图

从图中可以看出，车牌的粗定位的任务实现了，下面的工作就是完成车牌细定位的工作，提取出正确的车牌区域，但图像仍有边框和铆钉的存在，处理办法会在本文的后面进行阐述。

### 2.3 车牌图像的准确定位

对于图像的边缘检测只能大体定位到车牌，并不能获取准确的车牌位置，这里面除了包含着真实的车牌位置还有一些虚假的拆牌区域，而在这些区域里面，我们只需要一个区域，就是真实的车牌所在的区域。要准确的定位车牌的位置就一定要除掉图像的虚假车牌区域，具体操作如下：

第一步：对候选连通域逐个进行长宽对比，这会使结果在一定区间上，将结果的区间设定在  $[2, 5]$ ，将计算结果不在这个区间的区域删除。

第二步：将经过第一步处理后的区域分别计算面积，这样计算结果也会在一定的区间里，将结果区间设定在

$[2500, 7500]$ ，然后排除那些结果不在这个区间的区域。第三步：再将经过前两步排除剩下的区域进行色彩分量的分析，将其与我国车牌色彩饱和度特征进行对比，排除那些不匹配的区域。

剩下的就是真实的车牌所在的区域，将这些区域用最小外接矩形标记并记下坐标，可以使用函数 `cv2.findContours()` 对图像进行矩形检测，具体的效果图如图 3.1 所示：



图 3.1 车牌精准定位后图像

### 2.4 本章小结

本章第一步对采集到了图像进行了预处理操作，图像经过由彩色 RGB 通道图像转化成 HSV 通道图像，再把 HSV 通道图像进一步简化为单通道的灰度图像，再将灰度图像经过高斯滤波、边缘检测、二值化处理和形态学处理，尽可能的简化了原图像中所携带的信息，降低了系统在图像识别中的难度，

提高了系统的工作效率和可靠性，同时本章还对车牌进行了精准的定位。本文下一步就是对车牌图像的字符分割。

### 3. 车牌字符分割技术的设计与实现

将车牌图像中的字符分割成一个个单独的字符这就是车牌图像的字符分割，车牌图像的字符分割是为了下一步的字符识别打下基础。

#### 3.1 车牌图像的水平校正

经过定位后，所获得的车牌难免存在某种程度的倾斜情况，最终会影响到车牌识别正确率。所以，首先应该检查倾斜角度，做倾斜矫正。

##### 3.1.1 车牌图像倾斜角度检测

本文使用的是 OpenCV 中的函数 `HoughLines` 和函数 `HoughLinesP`，`HoughLines` 函数可以调用标准的 `HoughLine` 转换和多尺度的 `HoughLine` 转换。`HoughLinesP` 函数用于调用累积概率 `HoughLine` 转换。

要在二维坐标轴上表示一条直线，可以用方程  $y = a * x + b$ 。这里我们用通过函数的极坐标来求出函数中的  $a$  和  $b$ ，函数极坐标的表达式如公式 3.1 所示：

$$\rho = x * \cos \theta + y * \sin \theta \quad (3.1)$$

这些都是一条直线的基本特征。只要在 OpenCV 里调用霍夫线变换，就可以得到这两个值。

##### 3.1.2 车牌图像的倾斜矫正

本文对于图片倾斜矫正处理的方法是先利用霍夫变换，找到与垂直方向夹角不大于  $30^\circ$  的直线，接着对这些直线与垂直方向的夹角取平均值  $avAng$ ，然后再将图像旋转  $avAng$ ，以此来实现车牌校正，车牌图像校正前后对比如图 4.1 和 4.2 所示。



图 4.1 车牌图像水平校正前 图 4.2 车牌图像水平校正后经过两图对比可以看出，经过水平校正后的图片更加规整，更加利于系统对其进行处理。

#### 3.2 车牌边框和铆钉去除

经过 2.2.4 节的操作后我们得到了车牌图像的二值图像，但是二值化的图像里仍然惨在这边框和铆钉，这会对我们的垂直投影造成很大影响。为了使系统能够更加准确的处理车牌图像，我们要去除

二值化图像中的边框和铆钉。根据我国车牌特点，在车牌的第二个的右上方和第六个字符的左上方有二个铆钉。在目前去车牌边框和铆钉的方法非灰度跳变法莫属[21]。

灰度跳变法基本原理：将灰度变化平缓的边框区域看成背景，把灰度变化快的字符区域看成主题。设定一个阈值，是系统在扫描过程中可以分辨出背景与主体的边界。在对调处理的车牌图像进行水平扫描的过程中，把灰度跳变的次数记录下来；根据跳变规律提取字符区域。灰度跳变为 0 的背景行没有铆钉，灰度跳变加 1 的背景有铆钉。设定阈值分割出车牌。一个字符会发生不小于两次的 2 次跳变，一行像素会发生不小于 12 次跳变。将前面提到的阈值设定为 9，将不大于 9 次的像素值跳变次的行投影值全设为 0。具体步骤：首先令相关参数如下：

H: 二值车牌图像的高度；

L: 车牌图像的长度；

N: 水平方向投影的次数，其初始值为 0；

S: 字符上边界坐标；

D: 字符下边界坐标；

(1) 统计行像素上发生跳变的次数，发生一次跳变对 N 加 1；

(2) 设置阈值 T 对步骤 (1) 的 N 进行筛选。从中间值  $\frac{N}{2}$  开始向两端扫描。再向上扫描，当 T 大于跳变次数时扫描结束，记录位置 L。接着向下扫描，同理 T 大于跳变次数时扫描结束，记录位置 L。

(3) 将图像从 H 开始扫描一直到 L，再一次获取图像，如图 4.3 所示：



图 4.3 除去边框和铆钉车牌图像

观察本图可知，车牌图像在进行边框和铆钉的去除工作后更加简洁了，去除了图像中的干扰因素，系统的可靠性进一步得到了加强。

### 3.3 车牌字符分割的具体步骤与操作

#### 3.3.1 车牌字符的特点

我国现阶段实行的机动车号牌规范为《我国机动车号牌》(GA36-2007)。如图 4.4 所示，是一幅 44cm×14cm 的汽车牌照中，在此牌照中，有 7 个字符：省、直辖市、自治区的简称（汉字字符）排在第一个；大写字母，即所在城市的代号排在第二个。后五个字符可是数字和字母。牌照和其中字符的

高度与宽度确定。第二个字符与第三个字符的间隔比其他字符间隔大，上述信息可以作为先验知识简化字符分割的难度，提升字符分割精确度。

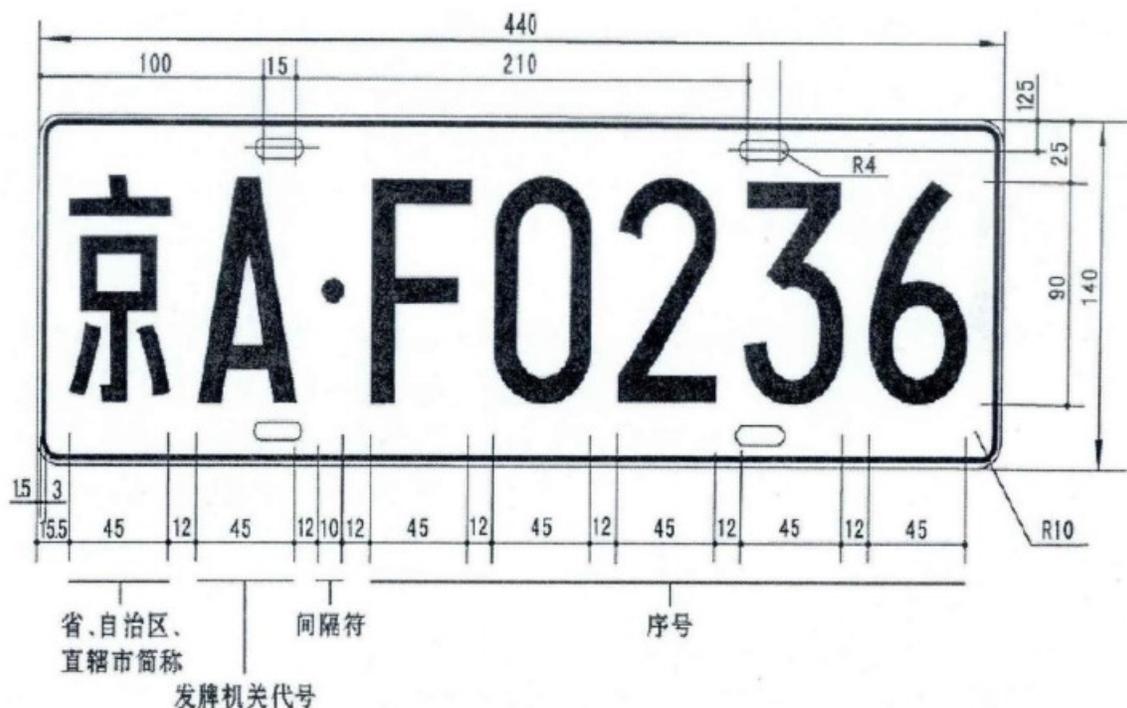


图 4.4 我国常见汽车号牌样式

3.3.2 垂直投影结合车牌字符特点的字符分割本文使用的字符分割方法为垂直投影法，具体方法如下：字符的切分是字符分割最重要的异步，是对前面处理切割后的完整图片根据字符进行切分，切分出一个一个单独的字符。首先要找到字符的边界，根据字符的边界进行切分，这样就能提取出单独

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/605110210113012001>