

# 深圳大学

## 本科毕业论文（设计）

题目： 基于机器视觉的智能辅助驾驶系统

姓名： 陈润城

专业： 自动化

学院： 机电与控制工程学院

学号： 2013110267

指导教师： 徐颖

职称： 副教授

2017 年 5 月 2 日

## 深圳大学本科毕业论文（设计）诚信声明

本人郑重声明：所呈交的毕业论文（设计），题目《基于机器视觉的智能辅助驾驶系统》是本人在指导教师的指导下，独立进行研究工作所取得的成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明。除此之外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。本人完全意识到本声明的法律结果。

毕业论文（设计）作者签名：

日期：                      年      月      日

## 摘 要

随着技术的进步，传统的汽车安全技术已经不能满足人们对车辆安全的需求，人们开始寻求一种能够真正降低交通事故发生率的车辆主动安全技术，车辆智能辅助驾驶系统正是由此诞生。在本文中，作者结合目前主流算法，设计了一种实用的智能汽车辅助驾驶系统，重点研究了车辆识别，车道线检测预警，限速标志和红绿灯路况分析四个部分。

在车辆识别算法中，研究了一种利用机器学习对车辆进行高效准确识别的方法。通过大量高质量的车辆图像正负样本，选取Haar-like特征，经机器学习算法Adaboost进行训练，获得最终的Cascade级联分类器，并对实际道路场景进行测试，表明了该算法在车辆识别中的快速性和准确性。

在车道线检测预警算法中，主要研究了消失点有效提取车道线的方法。在感兴趣区域中对原始图像进行背景减除，过滤大量干扰信息，随后使用Canny算法提取目标边缘，通过Hough直线变换提取所有直线信息，利用车道线的斜率、位置特征排除干扰直线，借助消失点限制条件精确提取出车道线信息。最后，通过车辆的横向偏移以及偏航角计算，对车辆是否偏离车道线进行预警。实验表示，算法具有高效准确，鲁棒性强的优点。

在限速标志算法中，提出了一种Haar+Adaboost分类器和HOG+SVM分类器串联识别的有效方法。首先通过检测时间短且分类效果好的重要特征所组成的Haar强分类器快速且大量地进行非目标区域过滤，然后对过滤后的区域进行面积大小和红色像素点排除，选取出最后的疑似区域进行HOG特征的精确验证。实验表明，HOG特征描述的限速标志边缘特性更值得信任，算法检测速度快，效果准确。

在红绿灯算法中，将红绿灯状态识别转换为红灯状态识别，并使用HOG特征进行红灯验证。首先对原始图像进行HSV颜色空间变换，提取感兴趣区域内的红色像素点，然后获取面积区间[40-70]之间的区域位置信息，在原始图像中进行裁剪并使用SVM分类器进行验证。实验表明，算法能够快速准确地识别红灯状态，满足智能辅助驾驶系统的一般性应用。

最后，利用多线程技术，将以上功能模块有机融合成智能辅助驾驶系统，通过画面预警方式，为车辆的安全行驶保驾护航。

**关键词：**车辆识别，车道线检测，限速标志，SVM，Adaboost

## Abstract

With the progress of technology, the traditional automotive safety technology couldn't satisfy people's demand for vehicle safety. People start to look for a kind of vehicle active safety technology which could actually reduce the incidence of traffic accidents. Thus vehicle intelligent auxiliary driving system was born. In this article, the author combines the current mainstream algorithm, designs a practical vehicle intelligent auxiliary driving system and mainly studies the following four parts: the vehicle identification, the warning of the lane line detection, the speed limit signs and traffic analysis through traffic lights.

In the aspects of vehicle recognition algorithm, the author studied a method of using machine learning to identify vehicles efficiently and accurately. By using many positive and negative samples of high quality vehicle image, selecting features of Haar-like, being trained by the machine learning algorithm of Adaboost, obtaining the final Cascade classifier, as well as testing the actual road scene, quickness and accuracy of the algorithm in the vehicle identification has been showed.

In the lane line detection and early warning algorithm, the method of extracting the driveway line in an effective way has been studied primarily. The author deducts the original image background in areas of interest, filters the interference information, and then uses the Canny algorithm to extract target edge, extracts all information of straight line by Hough linear transformation, eliminates interference of straight line by using the slope and location features of lane line and extract the lane line accurately with the aid of the vanishing point constraints information. Finally, the vehicle's lateral deviation and yaw Angle calculation are used to warn against the vehicle about deviating from the lane line. The experimental results shows that the algorithm has the advantages of high efficiency、 high accuracy and strong robust.

In the speed limit sign algorithm, an effective method for the series identification of Haar + Adaboost classifier and HOG + SVM classifier is proposed. First, filters the non-objected area quickly and largely by detecting Haar strong classifier with important characteristics of short timing and good effect of classification. Then, rules out the filtered area according to size and red pixels and picks out the last of the suspected area for accurate verification of HOG feature. The experiment shows that the edge characteristic of the speed limit marked by HOG-feature are more trustworthy, as well as the algorithm is faster and the results are accurate.

In the traffic light algorithm, the traffic light status recognition is converted to red light status recognition while HOG-feature is used to verify the red light. First of all,proceeds HSV color space transformation to the original image, extracts the interested region of red pixels, then gets area location information ranging between [40-70], crops the original image and verifies by using the SVM classifier. The experiment shows that the algorithm could identify the red light condition quickly and accurately, and could satisfies the general application of intelligent assisted driving system also.

Finally, the author combines the function module of organic integration into intelligent auxiliary driving system through the technology of multithreading, strengthening the security of vehicle's driving by way of the image warnings.

**Key words:** Vehicle detection;Adaboost; Lane detection; Machine vision;SVM

## 目 录

摘 要.....	I
ABSTRACT.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	1
1.2.1 国外研究现状.....	1
1.2.2 国内发展现状.....	3
1.3 本论文的结构安排.....	3
2 车辆识别.....	5
2.1 基于 ADABOOST 算法的 HAAR 分类器.....	5
2.1.1 Haar-like 特征.....	5
2.1.2 积分图.....	6
2.1.3 Adaboost 算法.....	7
2.1.4 级联分类器.....	9
2.2 车辆识别流程.....	10
2.2.1 训练过程.....	10
2.2.2 分类器检测过程.....	11
2.3 本章小结.....	12
3 行人识别.....	13
3.1 基于 ADABOOST 算法的 HAAR 分类器.....	13
3.2 HAAR 分类器训练.....	13
3.3 行人图像识别.....	14
3.4 本章小结.....	15
4 车道线识别预警.....	16
4.1 车道线偏移预警算法流程.....	16
4.2 图像预处理.....	16
4.2.1 图像灰度化.....	17
4.2.2 图像滤波去噪.....	17
4.3 图像背景减除.....	17
4.4 图像边缘提取.....	18

4.5	霍夫变换直线提取 .....	18
4.6	车道线特征提取 .....	20
4.7	消失点精确提取车道线 .....	21
4.8	车道线偏移预警 .....	22
4.8.1	车道线横向位移偏离模型 .....	22
4.8.2	车辆偏航角模型 .....	22
4.9	本章小结 .....	23
5	限速标志识别 .....	24
5.1	限速标志检测算法流程 .....	24
5.2	HAAR 分类器初检测 .....	25
5.3	检测区域再过滤 .....	25
5.4	SVM 分类器验证 .....	25
5.4.1	HOG 特征 .....	26
5.4.2	SVM 分类器原理 .....	27
5.5	限速标志静态图像识别试验 .....	27
5.6	本章小结 .....	28
6	红绿灯识别 .....	29
6.1	红绿灯识别流程 .....	29
6.2	提取 HSV 红色区域 .....	29
6.3	图像形态学处理 .....	30
6.4	SVM 分类器训练 .....	31
6.5	红绿灯静态图像测试 .....	32
6.6	本章小结 .....	33
7	智能辅助驾驶系统设计 .....	34
7.1	系统软硬件环境 .....	34
7.1.1	软件环境 .....	34
7.1.2	硬件环境 .....	34
7.2	系统实现 .....	34
7.3	系统测试 .....	36
7.4	本章小结 .....	37
8	工作总结及展望 .....	38
8.1	总结 .....	38

8.2 展望 .....	38
参考文献 .....	40
致谢 .....	41



## 绪论

### 1.1 研究背景及意义

随着科技的高速发展，越来越多的汽车驶进了人们的生活。根据公安部交管局汽车保有量的相关报告，截至2016年底，我国机动车保有量已达2.9亿辆，年复合增长率约14.8%。其中汽车1.94亿辆，机动车驾驶员3.6亿人，全国总共有49个城市的汽车保有量超过百万辆，18个城市超2百万辆，6个城市超3百万辆，汽车保有量以每年千万数量级的速度增长。伴随着机动车数量的高速增长，交通事故也越来越多。据联合国2015年的统计数据，全世界每年大约有125万人死于交通事故。国内外统计资料表明<sup>[1]</sup>，80%左右的交通事故是由驾驶员因素造成的。文献<sup>[2]</sup>中指出，道路交通安全已是威胁人类健康的五大因素之一。

随着安全意识的提高，传统的只作用于事故发生后的汽车安全技术，例如安全带、安全气囊，已经不足以满足现代社会交通安全的需求。人们开始寻求一种主动的汽车安全技术，能在交通事故发生前为人们保驾护航，真正地降低交通事故的发生率。于是，各种智能辅助驾驶系统开始诞生，如前车碰撞警告（FCW），车道偏离警告（LDW），移动物体防撞预警（MOD），前车接近报警（FPW）。

借助车载摄像头、雷达、超声波等传感器，辅助驾驶系统可以获取前方交通信息，得到前方障碍物的速度和距离，可以提前预防碰撞；得到车道线的位置，可以防止车辆偏离越线。借助辅助系统的语音警告以及画面警告，驾驶员可以提前预防交通事故的发生。

目前，受益于处理器计算速度的提升，图像传感器在辅助系统的实时性得到保证，而机器视觉技术的发展，则为图像传感器的实用性奠定了理论基础。尤其是随着机器学习的浪潮，基于图像信息的分类识别算法层出不穷，图像传感器开始在汽车辅助驾驶领域中大放光彩。

与其他类型的传感器相比较，图像传感器更贴近汽车驾驶员的实际视角，符合汽车驾驶员的思维习惯；同时，它包含了大量的图像信息，配合适当的算法，图像传感器可以为驾驶员提供除空间坐标以外的信息，如交通标志的语义，红绿灯的状态。而且，由于成本较低，图像处理器更容易在民用领域中得到普及。

### 1.2 国内外研究现状

#### 1.2.1 国外研究现状

发达国家从二十世纪七十年代开始进行智能车辆研究。1998年，面对频发的交通事故，美国开展了智能车辆主动IVI(Intelligent Vehicle Initiative)计划<sup>[3]</sup>，该计划意图使驾驶员的注意力集中，以提高道路交通的安全性，同时引导防碰撞系统的普及。1999年，美

国卡耐基梅隆大学制造的无人驾驶汽车 Navlab-V,横穿了美国东西部,行程长达 5000km。而代表着全球最高水准的谷歌,在 2010 年的加州道路上组织实施了 7 辆 Google 无人驾驶汽车车队的试行。至今,谷歌无人车在自动驾驶模式下已经累积行驶达 228.5 万公里,而通过手动受控驾驶已经累计行驶 159.2 万公里。目前,这个数字还在以每周 1.6~2.1 万公里的速度在增长。同样著名的还有特斯拉,遍布 42 个国家超过 107000 辆的特斯拉汽车已累计行驶了近 20 亿英里。

从 2000 年开始,欧盟连续五年资助 PROTECTOR<sup>[4]</sup>以及 SAVE-U<sup>[5]</sup>项目,研制出基于机器视觉的行人检测系统;意大利 Parma<sup>[6]</sup>大学开发了基于单目视觉系统的 GOLD 系统,能够对行人、车辆和车道线实施监测。德国 DaimlerChrysler 公司开发了 Chamfer 系统,借助机器学习的方法,计算行人纹理特征和距离信息,结合行人姿势的模板匹配,准确捕获图像中行人的位置<sup>[7]</sup>。同时,其生产的商用汽车中,均配备了车道偏离预警系统,凭借后视镜上的车载摄像头,可以监测车辆的行驶轨迹,预防在没有打转向灯的情况下驶入相邻车道。宝马公司发布的 MINI 驾驶辅助系统,包括平视显示系统、带城市制动功能的碰撞和行人警告系统,同样基于摄像头的速度和车距控制系统可让车辆自动与前车保持安全车距。限速信息识别系统(Speed limit info.)、交通标志记忆和禁行信息系统则帮助驾驶者了解当前交通信息。

在日本,汽车制造商成立了“高速公路高级巡航系统研究协会”,Infiniti 公司推出了安全屏障系统、丰田公司推出了 STAR 系统<sup>[8]</sup>以及无人驾驶公共汽车,本田公司则推出了基于红外摄像机的行人检测系统。

2017 年 2 月,博世举办的汽车技术创新体验日上,展示了包括自适应巡航、侧视辅助系统、自动紧急制动以及自动泊车等一系列驾驶员辅助系统,让开车变得更轻松,同时这也是日后实现无人驾驶的基础。

ACC 自适应巡航系统,该系统的核心部件是雷达,通过雷达探测自身车辆与前车之间的距离,由电脑进行数据的分析与判断,并自动控制车辆加速或制动,最终实行与前车保持合理、安全车距的目的。

车道保持系统:该系统通过多功能摄像机对道路上的车道线进行检测,当发现车辆偏移行驶路线时,系统会发生警告,在驾驶员不干预的情况下会用很小的力(大约 3 牛)对方向盘自动进行纠正,以保证车辆能在车道内行驶。

自动紧急制动系统:搭配多功能摄像头,对行人、自行车以及动物等体积相对较小,而且形状比较特殊的物体进行检测,从而实现自动紧急制动功能。

侧视辅助系统:该系统能主动监督在车辆后方盲区内的其它车辆,并通过灯光闪烁、发出警示音等方式提醒驾驶者。该系统在大雾、雨雪天气,后视镜视线较差的情况下非常有用,能避免因为盲目变道导致的碰撞事故的发生。

自动泊车辅助系统:不再需要驾驶员控制刹车以及手动切换档位,真正实现全自动

化的泊车操作，驾驶者甚至不需要坐在车内。

### 1.2.2 国内发展现状

中国在智能汽车的开发方面要比国外稍晚，研发主要以高校为主。国防科技大学从二十世纪八十年代开始进行智能车辆研究，并于 1992 年研发出中国首辆真正意义上的无人驾驶汽车。

2003 年，清华大学研制成功 THMR-V (Tsinghua Mobile Robot-V) 型无人驾驶车辆，能在清晰车道线的结构化道路上完成巡线行驶，最高车速超过 100km/h。2007 年，国防科技大学机电工程与自动化学院和中国第一汽车集团公司联合研发的红旗旗舰无人驾驶轿车，其总体技术性能和指标已经达到世界先进水平。2011 年 7 月 14 日，中国自主研发的无人车——由国防科技大学自主研发的红旗 HQ3 无人车，首次完成了从长沙到武汉 286 公里的高速全程无人驾驶实验，创造了中国自主研发的无人车在复杂交通状况下自动驾驶的新纪录，标志着中国无人车在复杂环境识别、智能行为决策和控制等方面实现了新的技术突破，达到世界先进水平。2011 年 7 月，由一汽集团与国防科技大学共同研制的红旗 HQ3 无人驾驶汽车完成了 286 公里的高速全程无人驾驶试验，人工干预的距离仅占总里程的 0.78%。2012 年，军事交通学院的“军交猛狮 III 号”以无人驾驶状态行驶 114 公里，最高时速 105 公里 / 小时。

2015 年 12 月初，百度无人驾驶汽车在北京进行全程自动驾驶测跑，实现多次跟车减速、变道、超车、上下匝道、调头等复杂驾驶动作，完成了进入高速到驶出高速不同道路场景的切换，最高时速达 100 公里 / 小时，是国内无人车领域迄今为止进行的难度最大，最接近真实路况的开放道路测试。

## 1.3 本论文的结构安排

本文的章节结构安排如下：

第一章绪论。简述本课题相关技术的国内外外展现状。

第二章车辆检测。通过 Adaboost 算法训练 Haar 分类器对车辆进行识别。

第三章行人检测。通过 Adaboost 算法训练 Haar 分类器对行人进行识别。

第四章车道线检测。详述了车道线检测算法，通过背景减除和边缘提取，结合车道线特征和消失点计算，对车道线进行精确提取，并横向偏移和偏航角模型进行预警。

第五章限速标志检测。提出并详述了 Haar+Adaboost 和 HOG+SVM 分类器快速检测识别限速标志的有效方法。利用 Haar 分类器的快速性大量排除非目标区域，通过区域面积和颜色进一步过滤区域，最后利用 HOG 特征进行限速标志的精确验证。

第六章红绿灯检测。对原始图像感兴趣区域进行 HSV 颜色空间提取后，进行区域面积过滤，最后使用 SVM 分类器进行红灯状态的验证。

第七章智能辅助驾驶系统设计。设计系统的整体实现流程并进行实际测试。

第八章总结与展望。对智能辅助驾驶系统进行总结，分析系统不足，为进一步研

究提出建议。

## 2 车辆识别

### 2.1 基于 adaboost 算法的 haar 分类器

本文通过机器学习的方法进行车辆的识别。首先，计算车辆的 haar 特征，采用 Viola 提出的 Adaboost 算法训练成检测分类器，对送入分类器的图像进行分类。当该图像满足分类器的所有分类条件时，分类器判断该图像为车辆。

#### 2.1.1 Haar-like 特征

如下图 2.1 所示，将一副图像中黑色区域的像素值之和减去白色区域的像素值之和，所得数值即为该图像的 Haar-like 特征值。由图 2.1 计算得出的 4 个特征值代表 4 种不同类型的 Haar-like 特征。

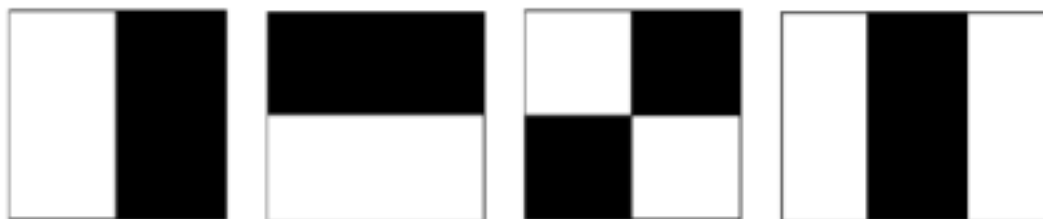


图 2.1 基本矩阵特征

显然，4 种简单的特征不足以唯一代表一个物体的特征，因此，在后来的特征描述中又引入了更多类型的 Haar-like 特征，以更准确地描述目标物体。图 2.2-2.4 列出了典型常用的 Haar-like 特征。

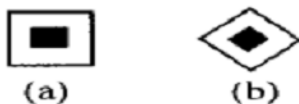


图 2.2 中心特征

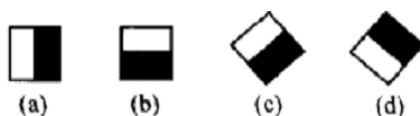


图 2.3 边界特征

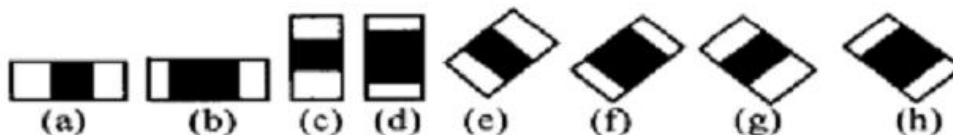


图 2.4 线特征

在图像密集的像素点中，借助 Haar-like 特征，我们可以很好地描述目标物体的轮廓，形状等特征，将目标物体与其他物体区分出来。图 2.5 表示了 Haar-like 特征如何描述物体。



图 2.5 Haar-like 特征描述示意图

在一副图像中，Haar-like 特征可能达到几万甚至几十万个，如果每一个特征都进行所有像素点的像素值加减，计算量将会十分巨大。因此，如何快速计算出所有特征值，将是用好 Haar-like 特征的关键，积分图的出现解决了这一难题。

### 2.1.2 积分图

积分图<sup>[9]</sup>如图 2.6 所示，定义黑色区域的所有像素值之和为  $I(x,y)$ ，如公式(2-1)所示：

$$I(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2-1)$$

式中：

$i(x,y)$  —— 坐标值  $(x,y)$  处的像素值。

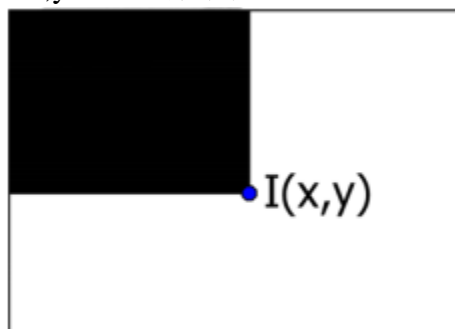


图 2.6 积分图

则图像中任意一处区域的像素值之和可由三次加减计算得到，如图 2.7 所示。

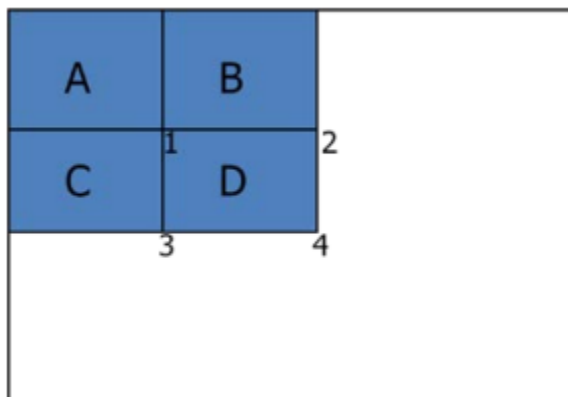


图 2.7 积分图计算示意图

对于图像中 D 区域的像素值之和，可由公式  $I_4(x,y) - (I_2(x,y) + I_3(x,y)) + I_1(x,y)$  快速获得，而  $I_1(x,y)$ 、 $I_2(x,y)$ 、 $I_3(x,y)$ 、 $I_4(x,y)$  的值我们可以在得到图像的第一时刻先行计算，并存放在一张表中，需要时直接查表获得。也就是说，任意一个区域的像素值之和，都

只需要 3 次运算便可以得到，大大简化了原本每个区域都要逐一计算的巨大工程，保证了 Haar-like 特征在实际应用场景检测的实时性。

每一个 Haar-like 特征都可以作为判断图像是否为目标物体的依据，Adaboost 算法提供了如何使用这些 Haar-like 特征进行判断的途径。

### 2.1.3 Adaboost 算法

Adaboost 算法<sup>[10]</sup>的思想是，给定同一个训练集，训练出不同的分类器，然后把若干弱分类器串联起来，构造成一个更强的最终分类器，即强分类器。算法的流程图如图 2.8 所示：

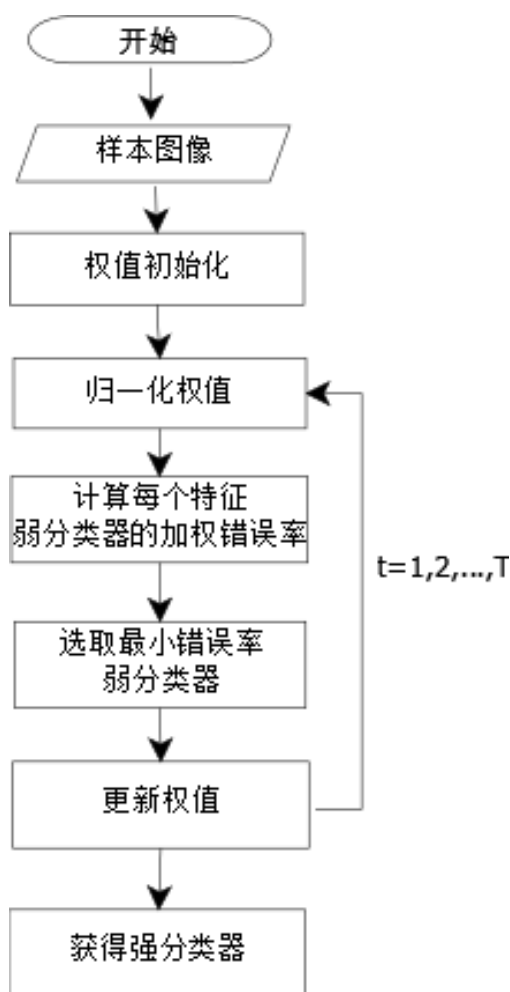


图 2.8 Adaboost 流程图

当我们选择 Haar-like 特征作为弱分类器时，我们需要对每一个特征  $f$  训练一个弱分类器  $h(x,f,p,\theta)$ ：

$$h(x,f,p,\theta) = \begin{cases} 1 & pf(x) < p\theta \\ 0 & \text{其他} \end{cases} \quad (2-2)$$

其中  $f$  为特征， $\theta$  为阈值， $p$  指示不等号方向， $x$  代表一个检测子窗口。对每个特征  $f$ ，训练一个弱分类器  $h(x,f,p,\theta)$ ，就是确定  $f$  的最优阈值，使该弱分类器对所有的训练样本的分类误差最低。

弱分类器训练的流程如下：

1: 对每个特征  $f$ ，计算所有样本的特征值。

2: 将特征值排序。

3: 对排好序的每个元素：

(1). 计算全部正样本的权重和  $T^+$ 。

(2). 计算全部负样本的权重和  $T^-$ 。

(3). 计算该元素前所有正样本的权重和  $S^+$ 。

(4). 计算该元素前所有负样本的权重和  $S^-$ 。

4: 选取当前元素的特征值  $F_{ki}$  和它前一个特征值  $F_{ki-1}$  中间的数为阈值，使得分类误差最小，该阈值的分类误差为：

$$e = \min\{S^+ + (T^- - S^-), S^- + (T^+ - S^+)\} \quad (2-3)$$

在训练好弱分类器后，借助 Adaboost 算法，开始选择最佳弱分类器进行强分类器的训练，具体的算法流程如下：

首先，给定一组训练集即  $N$  张图像： $\{x_1, y_1\}, \{x_2, y_2\}, \dots, \{x_n, y_n\}$ ，其中  $y_i \in \{0, 1\}$ ，0 和 1 分别表示负样本和正样本。然后，初始化正负样本的权值为  $w_{1,i} = 1/n$ 。

对  $t=1, 2, \dots, T$  次循环，执行以下步骤：

1. 归一化权重：

$$q_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}} \quad (2-4)$$

2. 对每一个特征  $f$ ，训练一个确定了最优阈值的弱分类器  $h(x, f)$ ；然后对所有特征计算出对应的弱分类器的加权错误率集  $\{\varepsilon_f\}$ ，加权错误率  $\varepsilon_f$  的公式为：

$$\varepsilon_f = \sum_i q_i |h(x, f) - y_i| \quad (2-5)$$

3. 在所有特征中，即加权错误率集中，选择具有最小错误率的最佳弱分类器  $h_t(x)$ 。

$$h_t(x) = \min \{\varepsilon_f\} = h(x, f_t) \quad (2-6)$$

4. 根据加权分类错误率调整下一次训练中每个样本的权重：

$$W_{t+1,j} = w_{t,j} \beta_t^{1-e_i}, \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t} \quad (2-7)$$

其中样本  $x_i$  被正确分类时， $e_i=0$ ，否则  $e_i=1$ ， $\varepsilon_t$  表示第  $t$  次循环中最佳弱分类器的分类错误率。

5. 根据加权分类错误率计算该最佳弱分类器在强分类器中的权重  $\alpha_t$ ：

$$\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t} \quad (2-8)$$

6. 经过  $T$  次循环后，得到  $T$  个最佳弱分类器，按公式 (3-8) 串联成强分类器：

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{其他} \end{cases} \quad (2-9)$$

利用强分类器检测图像时，组成强分类器的  $T$  个弱分类器会对图像进行投票，当最终的加权投票值大于平均投票值一半时，强分类器认为该图像为目标物体。



### 2.1.4 级联分类器

级联分类器的出现，大幅减少了实时运算时的计算时间。它把若干强分类器串联起来使用。在检测时，先通过计算时间少但分类效果显著的分类器过滤掉大量负样本，把剩余样本留给后面计算时间长的分类器再次验证。当样本满足所有分类器的通过条件时，分类器判断样本为目标物体，其检测流程如图 2.9 所示。

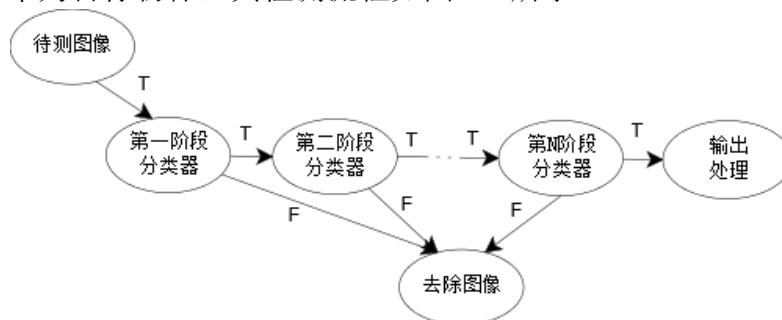


图 2.9 级联分类器检测流程图

假设 cascade 级联分类器中有  $T$  个强分类器， $f$  和  $d$  分别为某层分类器负样本误判率，正样本通过率，则级联分类器负样本误判率  $F = \prod_{i=1}^T f_i$ ，正样本通过率  $D = \prod_{i=1}^T d_i$ 。

训练Cascade级联分类器流程如下：

1. 给定每层强分类器的最小检测率  $d$  和最大误识率  $f$ ，级联分类器误识率  $F_t$ ， $P$  为正样本集， $N$  为负样本集；

2. 初始化  $F_0=1.0$ ， $D_0=1.0$ ， $i=0$ ；

3. For :  $F_i > F_t$

$i=i+1$ ；

$n_i=0$ ；

$F_i=F_{i-1}$ ；

For :  $F_i > f * F_{i-1}$

$n_i = n_i + 1$ ；

使用Adaboost算法在样本集 $P$ 和 $N$ 里训练含有 $n_i$ 个弱分类器的强分类器；

计算当前级联分类器的检测率 $D_i$ 和误识率 $F_i$ ；

For :  $d_i < d * D_{i-1}$ ；

降低第 $i$ 层强分类器的阈值；

计算当前级联分类器的检测率  $D_i$  和误识率  $F_i$ ；

End

$N = \Phi$ ；

对于当前级联分类器检测的非车辆图像，将误判的图像放入 $N$ ；

End

End

级联分类器的级数越高，检测越精确，但检测时间也越多。对于级联分类器，一般具有 10 个左右的强分类器，每个强分类器大约又含有 15 个弱分类器。在训练时，需要综合考虑检测速度和所需要的检测精度，以此选择合适的级联分类器级数。

## 2.2 车辆识别流程

车辆识别中，一般分为两个阶段，即样本训练阶段和检测阶段，其流程图如下图 2.10 所示：

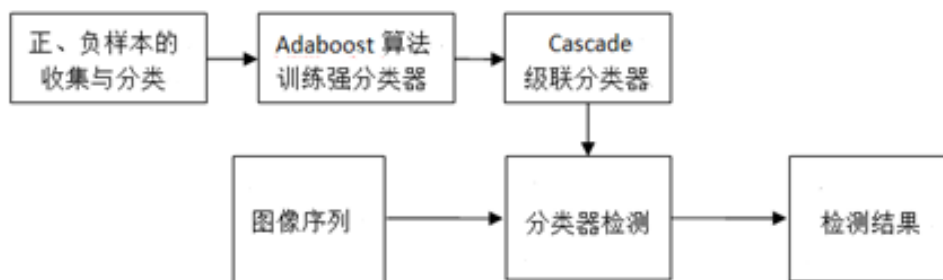


图 2.10 车辆识别流程

### 2.2.1 训练过程

分类器的训练包含四个过程，样本收集，样本预处理，生成样本特征描述文件，样本训练。

#### 1) 样本收集：

训练样本对于最终分类器的检测效果至关重要，正样本为车辆图像，负样本为不包含车辆图像，本文所用样本来源于 MIT-CBCL 项目、手动截取以及网络收集，在收集正样本时着重考虑拍摄车辆的角度，车型，在收集负样本时着重考虑了实际识别场景可能出现的随机情况，最终样本包含正样本 2828 幅，负样本 7708 幅，正负样本示意图如图 2.11、2.12 所示。



图 2.11 正样本示意图



图 2.12 负样本示意图

## 2) 样本预处理:

首先, 对收集到的样本统一缩放为 24X24 大小, 目的在于使每一种 Haar-like 特征在所有正负样本之中都存在, 保证了训练过程的有效性, 同时缩放有利于加快训练过程, 节省大量训练时间; 然后把正负样本存储的相对路径记录于.txt 文件中, 为程序读取图片做好准备。

## 3) 生成正样本特征描述文件:

正样本需要生成正样本描述文件, 负样本不需要。在 Windows 的 CMD 命令中, 运行 `createsamples.exe -info pos/info.txt -vec pos.vec -num 2828 -w 24 -h 24` 即可生成样本描述文件。

其中参数 `pos.vec` 表示正样本的特征描述文件, `-num` 表示正样本数量, `-w` 表示正样本宽度, `-h` 表示正样本高度。

## 4) 样本训练:

OpenCV 提供了训练程序 `haartraining.exe`, 在 CMD 命令中运行如下训练命令即可。

```
haartraining.exe -data cascade -vec pos.vec -bg neg/neg.txt -npos 2828 -nneg 7708 -nstages 18 -mem 1000 -mode ALL -w 24 -h 24 -nonsym
```

haar 分类器训练时间较长, 一般取决于样本数量, 计算机处理速度, 分配内存的大小, 本文所训练的分类器持续时间大约为一周。训练好 haar 分类器后, 可以得到一个 `cascade.xml` 文件, 这就是检测所需要的分类器。

### 2.2.2 分类器检测过程

在 VS2013 中, 使用 OpenCV 提供的分类器 `CvHaarClassifierCascade` 类, 载入训练好的级联分类器 `cascade.xml`, 然后调用下面的 OpenCV 函数进行目标检测:

```
CvSeq* cvHaarDetectObjects(  
    const CvArr* image,  CvHaarClassifierCascade* cascade,  
    CvMemStorage* storage,  double scale_factor=1.1,  
    int min_neighbors=3,  int flags=0,  
    CvSize min_size=cvSize(0, 0));
```

检测结果如下图 2.13 所示:



图 2.13 车辆检测结果

## 2.3 本章小结

本章详细介绍了通过 haar 分类器进行车辆识别的有效方法。首先收集大量检测目标,即车辆的正负样本,归一化成相同大小,然后对图像进行 haar-like 特征的计算,选择检测效果最好的特征进行 Adaboost 算法的强分类器训练,再集成成级联分类器,最后借助该分类器,对实际道路的车辆进行有效识别。

## 3 行人识别

### 3.1 基于 adaboost 算法的 haar 分类器

行人作为非刚性物体，在实际场景中存在各种各样的姿势形态，对于机器学习来说，需要非常多的训练样本，以更好地覆盖实际场景中可能出现的行人姿态。而对于 Adaboost，当样本数量越多时，其检测效果越精确，速度也更快。综合考虑下，本章依然采用 Adaboost 算法训练 Haar 分类器，对行人进行识别。

### 3.2 Haar 分类器训练

行人识别 Haar 分类器训练过程与车辆识别分类器训练过程相似，因为行人的特征与车辆有所区别，在训练时需要注意以下情况：

1) 行人的长宽比一般为 1:2，因此在训练时将检测窗口设置为 24 X 48，以更好地适应实际场景中的行人特征。

2) 实际场景中行人可能不是单独出现，可能伴随着单车，行李，背包等情况，因此，在正样本中需要适量包含此类情况。

3) 行人站立时与电线杆等物体相似。在选择正样本时，需要非常注意行人样本的质量，尽可能选择能够与电线杆等物体区分开来的样本，如双手放开的正样本；同时，将电线杆等竖直物体作为负样本送入 Haar 分类器中，以降低误检率。

在本章训练行人分类器中，使用正样本 4876 个，负样本 9192 个，样本来源于 MIT 数据库，USC 数据库，ETH 数据库，INRIA 数据库，手动截取以及网络收集，全面涵盖了行人各种可能的姿态。

对正样本进行缩放后，开始生成正样本特征描述文件，如图 3.1 所示。

图 3.1 行人样本特征描述文件

得到正样本特征描述文件后，开始训练分类器，为了保证训练过程正样本数量充足，取正样本数量为  $4876 \times 0.85 = 4144$  个，如图 3.2 所示

```
PARAMETERS:
cascadeDirName: xnl
vecFileName: pos.vec
bgFileName: neg\neg.txt
numPos: 4144
numNeg: 9192
numStages: 18
precalcValBufSize[Mb] : 256
precalcIdxBufSize[Mb] : 256
stageType: BOOST
featureType: HAAR
sampleWidth: 24
sampleHeight: 48
boostType: GAB
minHitRate: 0.995
maxFalseAlarmRate: 0.5
weightTrimRate: 0.95
maxDepth: 1
maxWeakCount: 100
mode: BASIC

===== TRAINING 0-stage =====
<BEGIN
POS count : consumed   4144 : 4144
NEG count : acceptanceRatio   9192 : 1
Precalculation time: 10.332
+-----+
| N |   HR |   FA |
+-----+
```

图 3.2 Haar 分类器训练示意

### 3.3 行人图像识别

分类器训练完成后,以实际道路场景对分类器进行试验,试验结果如下图 3.3 所示。

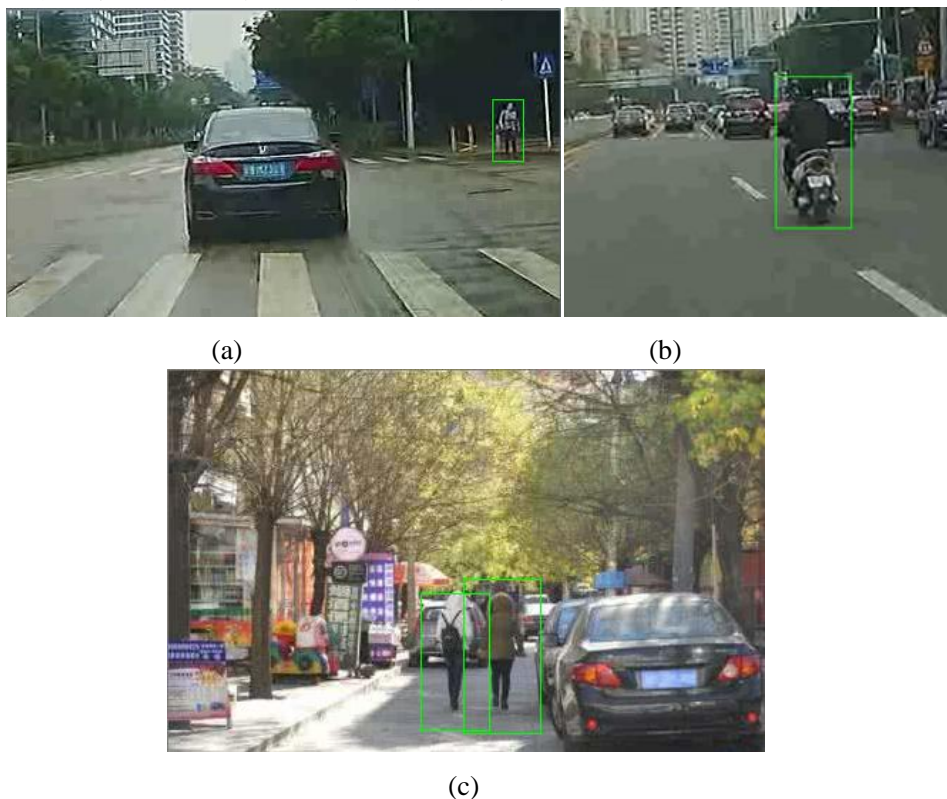


图 3.3 行人识别试验结果

图 3.3(a)中,行人穿越斑马线,是道路场景中常见的情况,识别效果很好。在图 3.3(b)中,摩托车正行走在车道线上,可能发生越道情况,而即使有摩托车的遮挡,Haar 分类器依然能够识别出行人,并提醒车辆驾驶员。图 3.3(c)中,对于多行人情况,Haar 分类器依然能够胜任识别任务。

### 3.4 本章小结

本章通过 Adaboost 算法训练 Haar 行人分类器，通过大量高质量的正样本，涵盖了各种各样的行人姿态，并在负样本中强调了与行人相似的物体，以提高 Haar 分类器识别的准确率。最后，在实际道路场景试验下，表明了 Haar 分类器能够胜任正常行人、有遮挡行人、多行人场景下的识别任务，表现出了良好的鲁棒性。

## 4 车道线识别预警

### 4.1 车道线偏移预警算法流程

车辆行驶时，由于摄像机镜头的透视效果，车道线在图像中是两条相交于远处的直线，并非平行线。在图像中，车道线有以下明显的特征：

- 1) 颜色：颜色为白色，单一且密集度高，在灰度空间中灰度值比较高；
- 2) 边缘：车道线周围路面的灰度值较低，单一且连续，使得二者之间的灰度差值明显且稳定，易于提取车道线的边缘特征，这也是车道线最明显的特征。
- 3) 位置：车道线在图像中的位置相对稳定，一般出现在图像的下方，大多呈左右对称分布。
- 4) 形状：在视频流相同位置截取一小段车道线，其形状近似于平行四边形，有相对固定的宽度和高度。

本文将根据车道线以上特征，借助霍夫变换和消失点原理对车道线进行精确提取，最后对车辆是否偏移车道进行判断。车道线偏移预警算法流程如下图 4.1 所示：

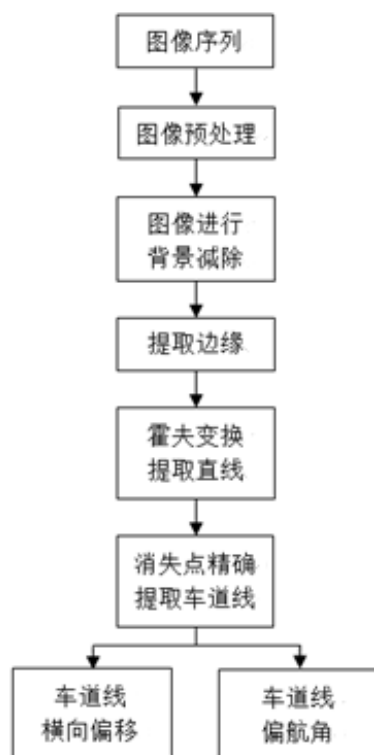


图 4.1 车道线偏移预警算法流程

### 4.2 图像预处理

实际道路场景中，不仅存在我们需要的车道线图像信息，还存在我们不需要甚至严重影响处理结果的干扰信息，例如路面积水，树木阴影等。因此，在对图像进行处理之前，一般都需要先进行某些预处理，一方面可以去除图像中的干扰信息，另一方面可以



提高图像处理效果的速度和准确度。例如，提取车道线时，如果将图像二值化，减少处理的颜色信息，则可以极大提高处理速度；而二值化前先对图像进行滤波处理，则可以提高处理的准确度。

#### 4.2.1 图像灰度化

摄像机拍摄到的彩色画面，是基于 RGB 颜色模型空间的成像画面。RGB 颜色模型通过将红、蓝、绿三种颜色的色光按照不同比例进行叠加，进而可以产生出其他不同的颜色。在计算机中，一般采取 24 位色的模式，即红、蓝、绿三个通道各使用 8 位无符号整数（0 至 255），对颜色进行表示。除此之外，还有一种只由一个 8 位无符号整数表示颜色的灰度模型，它可由 R、G、B 通道计算得到，计算公式（4-1）如下所示：

$$\text{Gray} = 0.299r + 0.587g + 0.114b \quad (4-1)$$

在计算机中，一般采用公式（5-2）计算图像灰度，以减少计算机中的浮点数运算。

$$\text{Gray} = (299r + 587g + 114b)/1000 \quad (4-2)$$

#### 4.2.2 图像滤波去噪

车载摄像头拍摄的图像中，由于光照变化、摄像机抖动、镜头蒙尘等因素，可能会产生模糊，斑点等成像质量较差的情况。这些像素点的存在会对图像的精细化操作产生不良影响，使处理效果无法达到预期。针对这些情况，可通过图像滤波去噪的方法加以改善。

##### 1) 中值滤波<sup>[8]</sup>

如下图 4.2 所示，在原图像中设定一个 M X N 大小的窗口（以 3 X 3 为例），对该窗口内的所有像素点的像素值进行排序，选取这些像素值的中值，替代该窗口中心处的像素值。即，对 {13, 5, 24, 9, 18, 12, 21, 7, 1} 9 个像素值进行排序，取中值 12 替代窗口中心处的 18，同时保持其余 8 个像素值不变。

当该窗口遍历整幅图像后，便完成了整幅图像的中值滤波。

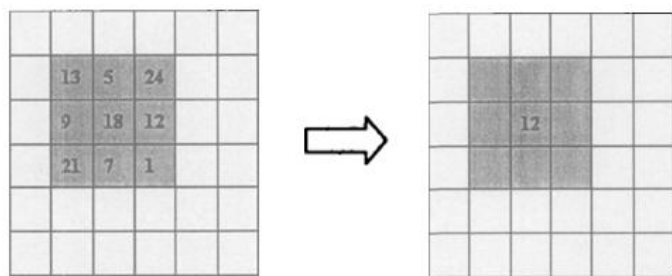


图 4.2 中值滤波

##### 2) 均值滤波<sup>[8]</sup>

均值滤波与中值滤波类似，将窗口内的像素值取平均，将该平均值替代窗口中心处的像素值即可。

### 4.3 图像背景减除

图像背景减除的目的在于去除掉大量不需要的图像信息和噪声干扰,使图像中目标物体更加明显。图像背景减除的具体实现如下所示:

- 1) 在图像底部正中取  $40 \times 50$  大小的样本窗口;
- 2) 计算样本窗口中所有像素点的灰度平均值  $m$  和标准差  $\sigma$ ;
- 3) 若标准差  $\sigma > 6$  时,向左平移 10 个像素点再次提取路面背景,重复步骤 2,若连续 6 次提取失败,意味着路况不好,舍弃此次图像帧,进行下一帧图像处理;若标准差  $\sigma < 6$  时,路面背景提取成功,转到步骤 4.
- 4) 将原图像所有像素值减去步骤 2 所得的背景灰度值  $m$ ,完成图像背景减除。

图像背景减除效果如下图 4.3 所示,原图中路面灰度值高低起伏,而在减除路面背景后,车道线标志十分明显,背景颜色为连续单一的黑色,为接下来的车道线边缘提取做了很好的铺垫。

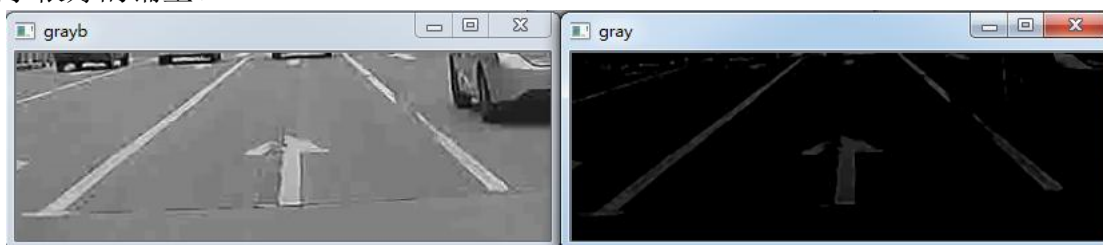


图 4.3 图像背景减除效果图

#### 4.4 图像边缘提取

边缘是分割目标物体和背景的重要特征。在图像中,边缘通常表现为两像素值之间的阶跃突变区域,利用这个特征,可以有效提取出目标物体的边缘。在图像处理中,Canny 算法由于边缘提取效果好,运算速度快,广泛为大家所用。

Canny 算法具体分为四步实现:

- 1) 利用高斯卷积内核,对图像进行高斯平滑,抑制噪声。
- 2) 计算梯度幅值、方向,即获取图像所有像素的边缘强度和边缘方向。
- 3) 去除非局部极大值,设置其余点为可选边缘点。
- 4) 应用双门限方法,以高门限作为边缘点,从种子点出发,根据低门限跟踪图像边缘点。这样的做法可以很好地抑制孤立噪声,而且在边缘处不会出现虚线。

运用 Canny 算法提取边缘的效果如图 4.4 所示:

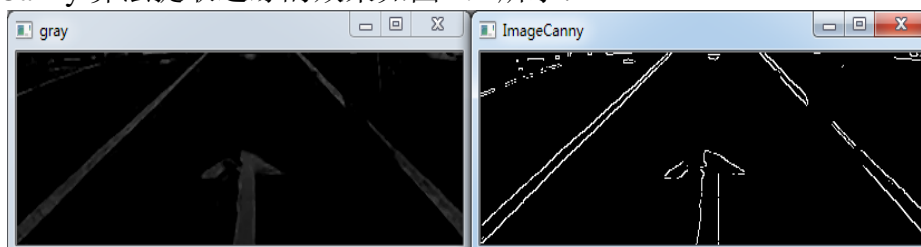


图 4.4 Canny 算法边缘提取

#### 4.5 霍夫变换直线提取

在图像处理中，霍夫变换常用于识别基本几何形状，如直线、圆。在本文中，由于预处理方法得当，对车道线进行边缘提取后，直线特征明显，很适合霍夫变换提取直线。

对于公式(4-3)，在我们熟悉的  $x, y$  参数平面，如果  $x, y$  为未知数， $\cos\theta$ 、 $\sin\theta$ 、 $\rho$  为定值，则该方程表示一条直线；如果  $x, y$  为定值， $\theta, \rho$  为未知数，则该方程同样表示同一条直线；如果将  $x, y$  参数平面转换为  $\theta, \rho$  参数平面，则如图 4.5 所示，有：

1) 对于公式 (4-3)，在  $x, y$  参数平面表现为直线，在  $\theta, \rho$  参数平面表现为一个点。

2) 对于点  $(x, y)$ ，在  $x, y$  参数平面表现为一个点，而在  $\theta, \rho$  参数平面则表现为一条直线。

$$\rho = x * \cos\theta + y * \sin\theta \quad (4-3)$$

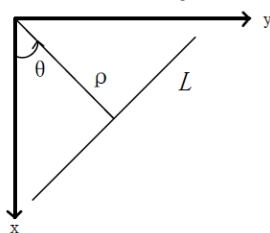


图 4.5 霍夫变换示意图

也就是说，图 4.5 中， $x, y$  参数平面直线  $L$  上的  $N$  个点，对应于  $\theta, \rho$  参数平面上的  $N$  条曲线，并且，这  $N$  条曲线会相交于一点  $(\theta, \rho)$ 。当某一相交点上经过曲线数量越多，该相交点越可能表示  $x, y$  参数平面上的一条直线。

只要统计  $\theta, \rho$  参数平面上相交点所经过曲线的数量，取经过曲线数量最多的点  $(\theta', \rho')$ ，再变换回  $x, y$  参数平面，即可得  $x, y$  参数平面上的直线。

霍夫变换提取直线流程图如下图 4.6 所示：

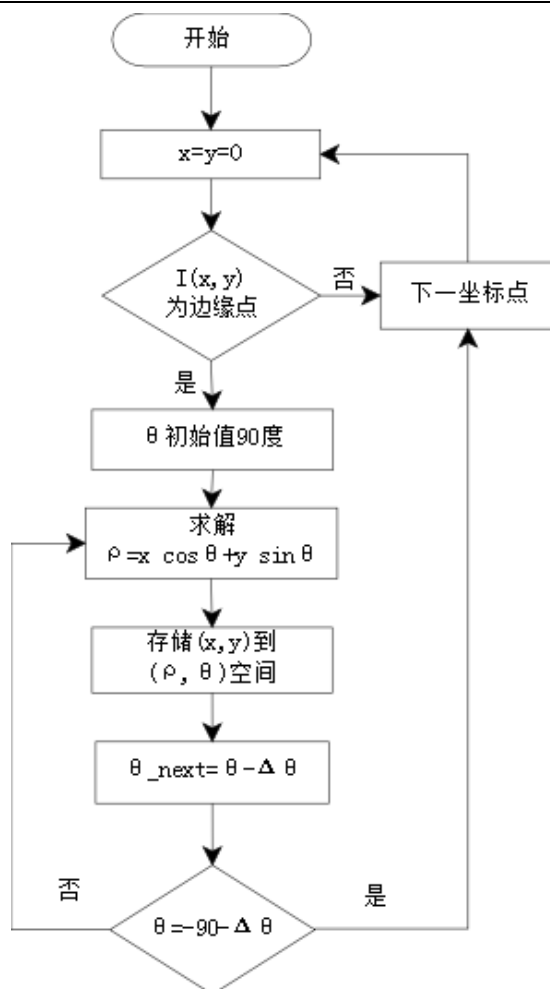


图 4.6 霍夫变换流程图

## 4.6 车道线特征提取

霍夫变换提取的直线，不仅仅包含了我们所需的车道线直线信息，还包括了其他物体的直线信息，如下图 4.7 所示：



图 4.7 霍夫变换直线提取结果

在图中，中间的箭头指标直线，图像两段的细小直线，都不是我们所需要的直线信息，如果不对这些直线进行去除过滤，会对车道线提取产生严重的影响。

1) 直线斜率筛选：由上图可知，车道线斜率在视频流中保持在相对稳定的区间里，而车道线的斜率区间，可以通过实验获取，只要摄像机固定，车道线的斜率也相对固定。因此，只要计算图中所有直线的斜率，把不在车道线斜率区间里的其他直线去除，便可以过滤掉绝大部分的干扰直线。

2) 直线端点位置筛选：对于斜率和车道线斜率相近的干扰直线，还可以通过其端

点位置进行过滤。由上图可知，中间位置的箭头，图像左上方和右上方的干扰直线，都可以通过直线端点的方法进行过滤。

经过以上两个步骤的直线筛选后，即使还有未清除干净的干扰直线，在接下来的消失点限制下，也不会产生多大的不良影响。而对于路面积水，树木阴影，由于其灰度值较低，在背景减除时便已经被过滤清除了。

#### 4.7 消失点精确提取车道线

文献<sup>[11]</sup>详细介绍了消失点的计算方法，步骤如下：

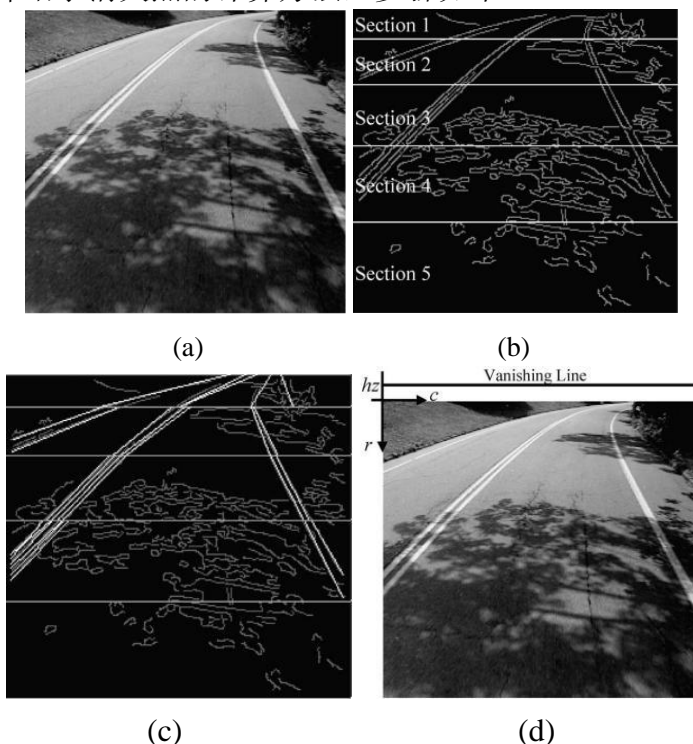


图 4.8 消失线计算

(a) 原始图像 (b) 图像边缘分区 (c) 直线提取结果  
(d) 得到消失线

步骤 1：原始图像如图 4.8 (a) 所示，通过 Canny 算法进行边缘提取，得到边缘图像 4.8 (b)。

步骤 2：在图 4.8 (b) 中，将图像划分成若干个水平区域。由于车道弯线在图像中，位置越高的弯曲越大，因此在分区时，位置越高，水平区域的高度越小，以利于更好地提取直线。

步骤 3：对图 4.8 (b)，进行霍夫变换提取直线，如图 4.8 (c) 所示。需要注意的是，进行霍夫变换时，是在每一个水平区域里单独进行，各不影响。由于区域 5 没有检测到直线，可以忽略不计。

步骤 4：进行消失线检测。在每一个水平区域里，单独进行直线配对，即计算出左右位置任意两条直线的交点，每两条直线只进行一次配对。将每个区域得到的交点，进

行统计，交点重复次数最多的位置，定义为消失点，消失点所在水平线即为消失线，如图 4.8 (d) 所示。

步骤 5：消失点即是左右两条车道线的交点，回溯产生该交点的直线，即可精确找到车道线。

车道线精确提取如下图 4.9 所示：



图 4.9 车道线精确提取效果图

## 4.8 车道线偏移预警

在精确提取出车道线后，便可以根据图像中车道线的位置对车辆偏离车道线的情况作出预警。预警的方式有两种，一种是车辆对车道线的横向偏移；一种是车辆行驶方向对车道线的偏移，即车辆的偏航角，当满足其中任意一个预警条件时，系统进行预警。

### 4.8.1 车道线横向位移偏离模型

由交通规则可以知道，在实际道路中的车道线间距都是固定的，在我国，车道线之间的间距一般为 3.75m，也就是说，对于图像中任意高度的车道线，在水平方向上的距离都是 3.75m，而图像中像素的分布是规律等距的，只要按照比例进行缩放就可以获取车辆与车道线之间的距离，如公式 (4-4) 所示：

$$d_{\text{left}} = \left( \frac{\text{width}_{\text{pic}}}{2} - x_{\text{left}} \right) * 3.75 / (x_{\text{right}} - x_{\text{left}}) \quad (4-4)$$

其中,  $d_{\text{left}}$  表示车辆中心到左车道线之间的距离,  $\text{width}_{\text{pic}}$  表示图像宽度,  $x_{\text{left}}$ 、 $x_{\text{right}}$  分别表示图像中左、右车道线的横坐标, 需要注意的是,  $x_{\text{left}}$ 、 $x_{\text{right}}$  取值需在同一水平线上。在本文中, 为了减少误差, 取图像底部的水平车道线横坐标进行计算。

当  $d_{\text{left}} < 1.35$  时, 认为车辆左偏; 当  $d_{\text{left}} > 2.4$  时, 认为车辆右偏。

### 4.8.2 车辆偏航角模型

在图像中, 参数坐标以图像左上角为原点; 而在实际计算时, 我们需要将图像进行上下反转, 如图 4.10 所示:

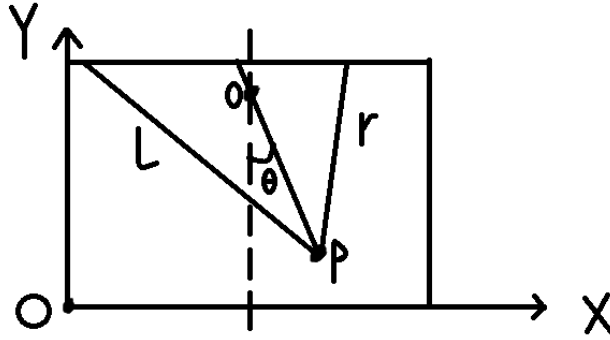


图 4.10 车辆偏航角计算模型

图 5.9 中，矩形为原始图像的上下反转示意图；直线  $l$  和  $r$  分别为原始图像中左、右车道线的上下反转；点  $P$  为两车道线的交点，即消失点；两车道线夹角的角平分线，与图中虚线，即车辆的行驶方向（ $y$  轴负方向），所成夹角  $\theta$  定义为车辆的行驶偏航角。

当车辆平行于车道线行驶时，偏航角  $\theta$  为  $0^\circ$ ；当车辆偏向左车道线行驶时，角平分线斜率为正；当车辆偏向右车道线行驶时，角平分线斜率为负；而偏航角大小又可以由角平分线斜率计算可得。因此，本文系统中直接计算角平分线斜率以判断车辆是否偏离车道线。

偏航角的计算步骤如下：

1: 设角平分线与车辆行驶方向所在虚线的交点为  $O(200, Y_o)$ ，车道线消失点  $P(x_p, y_p)$ ，其中  $y_p$  为消失点纵坐标，为已知值，而  $x_p$  可根据车道线斜率算出。

2: 根据  $O$  点到两车道线的距离相等，可由点到直线的距离公式（4-5）得出  $Y_o$ 。

$$d = \left| \frac{Ax + By + C}{\sqrt{A^2 + B^2}} \right| \quad (4-5)$$

由两点式可计算出角平分线的斜率。

计算出角平分线的斜率后，便可以判断车辆偏离车道线的具体情况，从而对车辆偏离进行预警。

## 4.9 本章小结

在本章中提取车道线的算法中，先对原始图像进行预处理以及背景减除，去除了大量干扰信息；随后对图像进行 Canny 算法边缘提取；然后由霍夫变换提取出图像中所有直线信息；再根据车道线的斜率特征以及位置特征筛选出疑似车道线的直线；最后利用消失点算法精确提取出车道线信息。提取车道线之后，对车道线进行建模，由车道线横向偏移模型以及偏航角模型对车辆是否偏离车道线进行联合判断。

## 5 限速标志识别

### 5.1 限速标志检测算法流程

在实际道路场景中，由于车载摄像头的拍摄角度，限速标志并非理想的圆形形状。在图像中，限速标志有以下明显的特征：

1) 颜色：限速标志颜色外围为明显的红色，表示意义的字体为黑色，在两者之间填充了白色。在图像中，颜色特征是限速标志最明显的特征，目前主流的识别方法中，皆是通过颜色特征进行初定位。

2) 形状：限速标志在图像中呈椭圆形，可利用这一特征对非限速标志物体进行筛选过滤。

3) 位置：限速标志一般位于摄像头视野的正上方或右上方。

由于限速标志与道路场景其他物体区别较大，与其他红色交通标志差别较小，常规的图像处理和分类器识别效果并不好。基于此背景，本文提出了两种分类器串联识别限速标志的算法，先通过 Adaboost 算法训练的 Haar 分类器快速且大量地过滤检测区域，借助图像处理技术选取疑似区域，最后通过基于 HOG 特征的 SVM 分类器进行精确验证，算法流程如下图 5.1 所示：

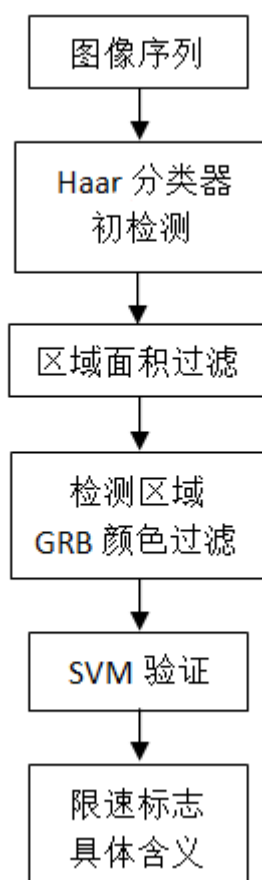


图 5.1 限速标志识别算法流程



## 5.2 Haar 分类器初检测

在本文提出的算法中，Haar 分类器的任务并不是精确识别出限速标志，而是快速且大量地排除非限速标志的局部图像，这是本文算法的关键之处，也是满足实时性的必然要求；也就是说，Haar 分类器的层级要控制在较少的范围之内，以利用检测时间短且分类效果好的重要特征所组成的强分类器进行区域过滤，而难以区分的区域则留给后面的 SVM 分类器进行验证。在本文算法中，之所以利用 SVM 分类器进行最终验证，是因为利用 HOG 特征进行限速标志的边缘表达更值得信任，识别效果更好。

在本节分类器训练中，所用正样本为{20,30,40,50,60,70,80}时速的最高限速标志图像，统一归一化为 32 X 32 分辨率大小，一共有 459 个，如下图 5.2 所示：

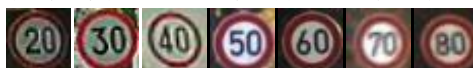


图 5.2 限速标志正样本

负样本选取实际道路中的随机场景，在视频中直接裁剪出 32 X 32 大小的图片，不破坏随机场景的完整性，同时，大量选取交通标志，特别是红色的圆形标志，以快速排除其他红色物体及交通标志。最终的负样本数量为 737 个，如下图 5.3 所示：



图 5.3 限速标志负样本

计算正负样本的 Haar 特征，通过 Adaboost 训练出 8 级级联分类器，对原始图像感兴趣区域进行快速过滤。

## 5.3 检测区域再过滤

在车载摄像头的视频流中，限速标志的大小约为分辨率 20 X 20 到 80 X 80。考虑到检测算法对于限速标志大小的要求，理想的检测环境为分辨率 40 X 40 到 70 X 70，因此，对于检测到的区域，先快速过滤掉此面积区间之外的局部区域。

对于剩余区域，进行 RGB 通道的红色像素点检测。为了提高区域过滤的速度和准确率，在检测红色像素点时，取左上角四分之一图像进行检测。若红色像素点的数量少于 30，则过滤该区域；否则判定为疑似区域。

在二值化图像中得到的原始图像红色区域，不仅包含了限速标志的位置信息，同时包含了其他红色物体的位置信息，例如车辆的红灯，路上穿红色衣服的行人，红绿灯等，还包含了阈值分割引起的误差，如橙色区域，针对这些情况，需要对二值图像进行形态学图像处理。在本节中，针对二值化图像进行中值滤波，腐蚀膨胀，粒子过滤，以去除大量非限速标志物体。

## 5.4 SVM 分类器验证

本节将采用机器学习的方法，对交通标志初定位区域提取出 HOG 特征集，送入 SVM

分类器中进行最终验证。

#### 5.4.1 HOG 特征

HOG<sup>[12]</sup> (histograms of oriented gradient, HOG), 即是图像的梯度方向直方图。作为一种特征描述子, 通过计算图像的梯度或者边缘方向密度来描述图像局部区域的表象和形状。由于 HOG 特征在实际应用中具有的强鲁棒性, 在图像处理中, 常采用 HOG 特征进行目标对象的特征描述。

HOG 特征的思想是: 将待检测图像划分为若干连通域, 作为细胞单元 (cell), 然后计算各个细胞单元中所有像素点的梯度或者边缘方向梯度直方图并进行归类, 再通过将细胞组成更大的块 (block), 使用重叠的局部对比归一化方法, 以取得最终的梯度方向直方图。下图 5.4 表示了细胞单元和分块之间的关系。

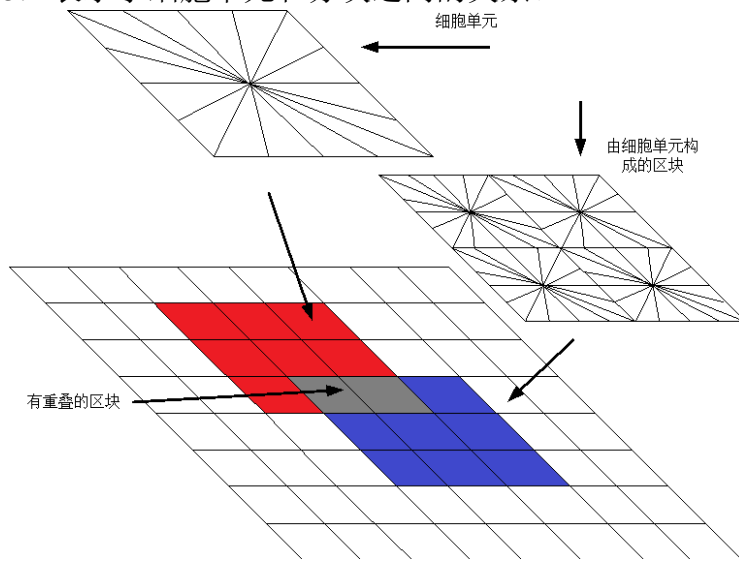


图 5.4 细胞单元与分块的关系

通过在图像局部细胞单元上计算得到的梯度方向直方图, 对于图像几何以及光学形变, 具有良好的不变性有点, 其具体的计算过程有以下步骤:

1. 图像归一化: 通过在较大范围内计算图像的直方图能力, 再将计算结果在细胞单元中进行归一化, 可以增强梯度方向直方图对光照改变的适应性。
2. 计算梯度值: 通过一维的离散型微分掩膜在细胞单元中计算一维梯度方向直方图, 包括水平和垂直两方向。
3. 通过细胞构建相应的梯度方向直方图。其中, 每一个细胞内的像素值要等于通道直方图的加权平均, 权重要等于按高斯加权计算的像素点梯度幅值。
4. 在分块 (block) 中归一化梯度。根据光照因素的影响和梯度变化, 使分类器适应这些变化并根据 L2 的范数归一化因子的需要, 如公式 (5-1) 所示:

$$f = \frac{1}{\sqrt{\|V\|^2 + e^2}} \quad (5-1)$$

提取 HOG 特征之后, 送入 SVM 分类器中进行训练。

### 5.4.2 SVM 分类器原理

支持向量机(Support Vector Machine, SVM)的核心思想为:通过核函数将输入空间映射到高维线性可分空间,然后求取该空间里的最优超平面,使得两类数据能够被分类在该平面的两侧。最优超平面对应的最优分类函数如下所示:

$$f(x) = \text{sign}\{\sum_{i=1}^n \alpha_i y_i K(x_i, x) + b\} \quad (5-2)$$

其中,  $x$  表示输入向量,  $(x, y)$  为支持向量的集合,  $y \in \{+1, -1\}$  是类别符号,  $\alpha$  与  $b$  为拉格朗日系数与分类阈值,  $K(x_i, x)$  是 SVM 核函数, 本文采用 LINEAR 线性核函数,  $K(x_i, x) = x_i^T x$ ;  $\text{sign}\{*\}$  则表示符号函数。

分类器的实质是,通过训练集训练一个分类模型,使我们能够通过输入样本的属性值,获得样本对应的目标值,从而完成对目标的分类。文献<sup>[13]</sup>中详细介绍了台湾大学林教授所开发的 LIBSVM 集成工具箱。

### 5.5 限速标志静态图像识别试验

首先,取一幅包含限速标志的实际道路场景图像,如图 5.5(a)所示,进行 Haar 分类器初检测,检测结果如图(b)、(c)所示。

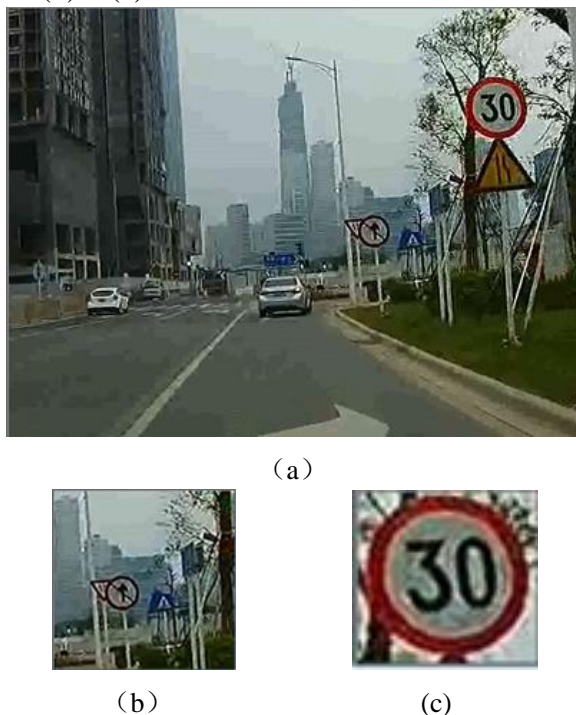


图 5.5 限速标志图像检测示意图

对于图像(b)、(c),进行检测区域面积和红色像素点再过滤,可得到唯一的图 5.5(c)作为疑似限速标志区域进行最终的 SVM 分类器验证,验证结果如下图 5.6 所示:



图 5.6 限速标志识别结果

## 5.6 本章小结

本章通过两种分类器串联进行限速标志的识别。先通过 Adaboost 算法训练的 Haar 分类器快速且大量地过滤检测区域；然后借助图像处理技术进行区域面积和红色像素点过滤，得到疑似限速标志区域；再通过基于 HOG 特征的 SVM 分类器进行精确验证。实验结果表明，该算法的识别效果符合预期，能快速准确地识别限速标志。

## 6 红绿灯识别

### 6.1 红绿灯识别流程

在实际道路场景中，相对于绿灯，驾驶员更关注的是红灯，也就是说，红绿灯识别的关键在于识别红灯，以防止由于驾驶员疏忽而导致的闯红灯情况的发生。因此，本文在对红绿灯特征做了详细研究的基础上，提出了一种快速准确的识别方法，巧妙地将红绿灯识别转换为红灯识别，算法流程如下图 6.1 所示：

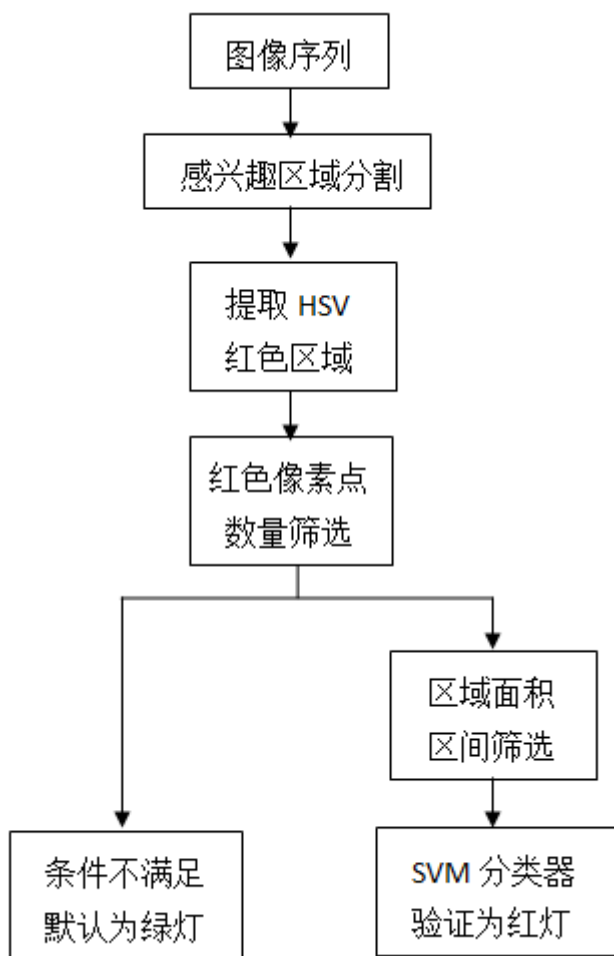


图 6.1 红绿灯识别算法流程

### 6.2 提取 HSV 红色区域

在实际道路场景中，由于光照影响，树木遮挡等因素，限速标志在图像中亮度较低，不能明显区别于其他物体和颜色，因此，本文将采取 HSV 颜色空间检测的方法，对限速标志进行初定位，以减少光照的影响。

HSV 颜色分别由色调、饱和度、亮度三个属性构成。一般以倒置的六棱锥模型进行表示，如图 5.2 所示。其中 H 代指色调，在水平面以角度  $0\sim 360^\circ$  表示，其中  $0^\circ$  为红色



轴，120°与 240°分别表示绿色和蓝色；S 代指饱和度，指到 V 轴的距离，其取值的范围为 0-1，值越大，色彩越鲜艳，当 S = 0 时，只有灰度。V 则是指色彩明亮的程度，范围在 0-1 之间。

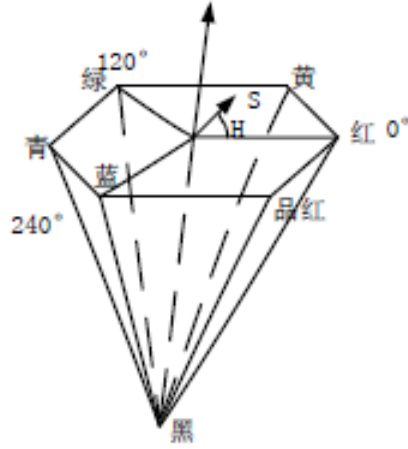


图 6.2 HSV 颜色模型

限速标志在图像中以 RGB 颜色模型表示，需要将 RGB 颜色空间转换为 HSV 空间，转换的公式如（5-1）所示：

$$H = \begin{cases} \frac{60(G-B)}{SV} & V = R \\ 60[2 + \frac{B-R}{SV}] & V = R \\ 60[4 + \frac{R-B}{SV}] & V = R \\ 0 & V = R \\ H + 360 & V = R \end{cases} \quad (5-1)$$

在实际试验中，当色调值为：0≤H≤15 以及 156≤H≤180，以及饱和度 S>159 时，对于原始图像中红色区域提取效果较好。HSV 红色区域的提取结果（位置信息）在二值化图像中进行表示。

经实际测试，一副图像中一个红绿灯占据的红色像素点约为 60 个单位。一般情况下，在感兴趣区域图像中，如果包含了红绿灯局部图像，在考虑了限速标志等红色物体的干扰，以及驾驶员紧急制动所需反应时间的情况下，合理的红色像素点上下阈值应为 40 和 400。因此，在提取 HSV 红色区域的同时，计算原始图像中红色像素点的数量。若红色像素点过少，则直接排除当前红绿灯为红灯的情况，默认情况为绿灯；若红色像素点数量过多，则严格缩小提取区域，重新计算像素点数量，若仍然过多时，认为当前图像背景（红绿灯周围或者天空）为红色，检测环境恶劣，不进行检测。之所以不进行检测的原因是，当检测背景为红色时，检测难度较大，而且，即使能检测出红灯，检测时间之长也不能够满足实时性的要求。

在提取了 HSV 红色区域后，对提取的二值图像进行形态学处理，以找出疑似红绿灯的区域。

### 6.3 图像形态学处理

连通域<sup>[14]</sup>的含义是，图像中含有相同像素值且位置相连的前景像素点围成的区域。连通域分析则是指找出并标记图像中所有连通域。

本文介绍一种简单又实用的连通域分析方法：Seed-Filling 种子填充法，其算法步骤如下所示：

步骤一：扫描当前图像，直到扫描像素点  $P(x,y) == 1$ ：

a：将像素点位置  $P(x,y)$  作为种子，赋予其一个 label，然后将该种子所有相邻的前景像素点都压入栈中；

b：弹出栈顶像素点，赋予其相同 label，然后将与该栈顶像素点相邻的所有前景像素点都压入栈中；

c：重复 b 步骤，直到栈空；

此时，完成步骤一即找到图像中一个连通区域，且该区域内所有像素值都被标记为相同 label；

步骤二：重复步骤一，直到所有像素点扫描结束。

在扫描结束后，便可以得到图像中的所有连通域，如图 5.5 所示。

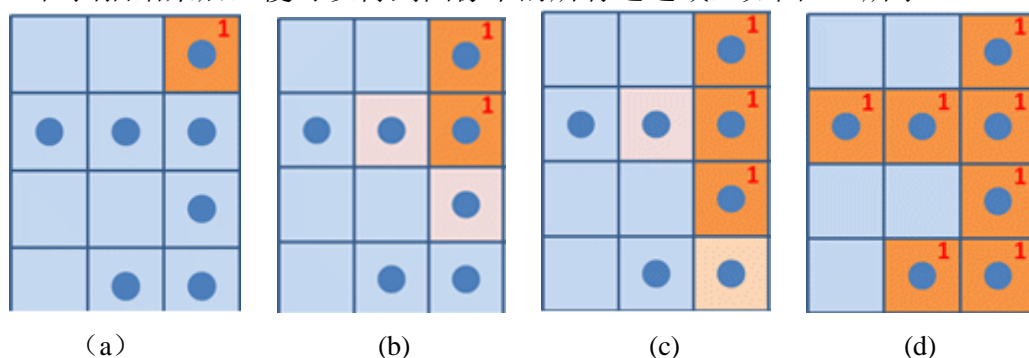


图 6.3 种子填充法示意图

首先扫描到第一行前景像素点，将其作为种子，接着将其相邻正下方的前景像素点压入栈中，如图 5.5 (a) 所示；随后进行步骤 b，将该像素点弹出，并标记其为 label1，再将该新标记的像素点相邻的两个像素点压入栈中，如图 5.5 (b) 所示；随后先弹出其中一个像素点进行标记，再压入新的相邻的像素点，如图 5.5 (c) 所示，需要注意的是，此时栈内（底）还有一个图 (b) 中压入的像素点没有弹出过；重复以上步骤，标记所有相邻像素点，如图 5.5 (d) 所示。

利用连通域分析，对提取的 HSV 二值区域进行面积筛选。在图像中，红绿灯的红色像素点面积变化范围在区间[40,70]之间。因此，过滤掉所有面积不在[40,70]区间内的红色区域；对剩余疑似红绿灯区域，上下各扩大 4 个像素单位，左右各扩大 15 个单位，获取将扩大后的区域，在原始图像中进行裁剪。对于裁剪后的局部图像，进行最终的 SVM 分类器验证。

## 6.4 SVM 分类器训练

首先从实际道路场景中收集适量的高质量正负样本，以准确表达出分类器的识别环境，为分类器优秀的分类效果做铺垫。

在本文中，为了使分类器有较高的识别率，所用样本皆从实际道路场景视频中进行截取。其中，正样本 60 张，为红绿灯的红灯状态，其分辨率大小为  $18 \times 9$ ，如图 6.4 所示。负样本则截取红绿灯周围场景，实际道路中可能出现的所有橙红色物体，例如交通标志，红色汽车，落叶，以及实际道路中的随机场景，一共有 520 张，如图 6.5 所示：

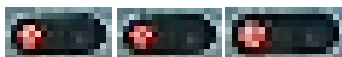


图 6.4 红灯正样本



图 6.5 红灯负样本

计算所有正负样本的 HOG 特征，送入 SVM 分类器中进行训练，训练的结果如下图 6.6 所示：



图 6.6 红绿灯分类器训练结果

## 6.5 红绿灯静态图像测试

训练好分类器之后，对面积区间在[40-70]之间的所有区域进行 SVM 分类器验证，验证结果如下图 6.7 所示：

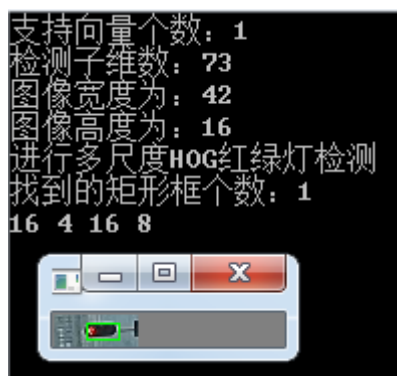


图 6.7 红绿灯

从图中绿色小框可以看到，本文算法在找到了红色区域的基础上，利用 SVM 分类器对红色区域进行了精确验证，使得红绿灯识别结果的精确率大大提升，满足实际道路场景的一般性使用。



## 6.6 本章小结

本章将红绿灯状态识别转换为红灯状态识别。首先对原始图像进行 HSV 颜色空间变换，对感兴趣区域内的红色像素点进行提取；然后过滤掉面积区间不在[40-70]之间的区域；对于剩余区域，扩大其上下边界各 4 个像素单位，左右边界各 15 个像素单位；然后获取其位置信息，在原始图像中进行裁剪；对于裁剪图像，进行最终的 SVM 分类器验证。在对实际图像进行测试时，能够快速准确地识别红灯状态，满足智能辅助驾驶系统的一般性应用。

## 7 智能辅助驾驶系统设计

### 7.1 系统软硬件环境

#### 7.1.1 软件环境

Windows7 操作系统：应用程序开发的基础环境，智能辅助驾驶系统的所有实现和应用皆在 Windows7 系统上完成。

Visual Studio 2013：通过 C++ 编程实现前文所有算法，并通过 MFC 实现智能辅助驾驶系统的用户界面。

OpenCV2.4.9 视觉处理库：在机器视觉领域，OpenCV 凭借其开源特性被大家广泛认可并且使用。它提供了绝大部分图像处理技术的实现函数，常用于视频图像的输入输出，统计分类以及模式识别，通过 C/C++ 和 Java 的函数接口，可以很方便地移植于其他平台之中。在近些年来，OpenCV 发展很迅速，扩展了很多新应用，也开始适用于 Android、iOS 操作系统。

LIBSVM 工具箱：Opencv 提供的函数并不支持 SVM 多分类问题，而 LIBSVM 可以 HOG 特征数据集进行训练，对目标实现多分类。

#### 7.1.2 硬件环境

测试硬件使用笔记本 Lenovo Win7 PC，其 CPU 为双核 Intel(R) Core(TM) i5-3120M，主频 2.5GHz，视频帧率 29fps。为了便于程序调试，在本文中使用先采集好的实际道路场景视频流作为图像输入。

### 7.2 系统实现

系统整体的算法流程如下图 7.1 所示：

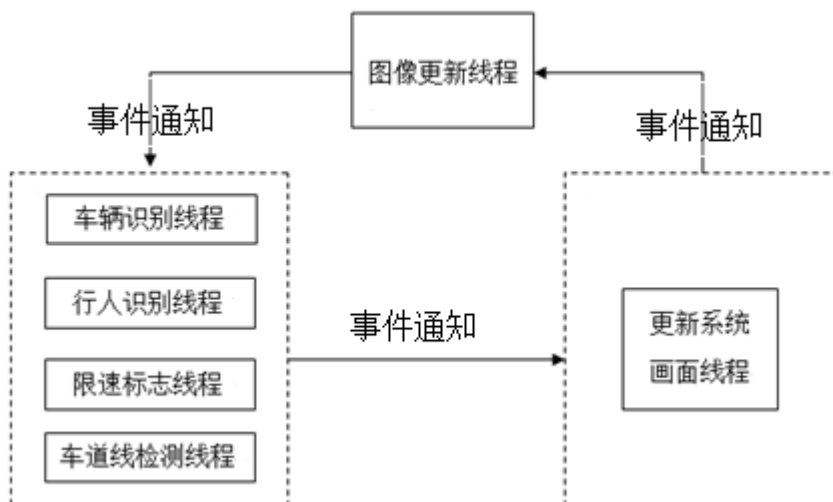


图 7.1 系统流程

具体的系统流程如下所述：

第一步：图像更新线程更新（取得）视频流的第一帧图像，然后通知车辆识别等四个线程，同时自身挂起（此时挂起的线程还有系统画面更新线程）。

第二步：这四个线程接收到事件通知后，开始同时独立对原始图像进行检测识别处理，互不影响。在处理完一帧原始图像后，开始通知系统画面更新线程，同时自身挂起（此时挂起的线程还有图像更新线程）。

第三步：系统画面更新线程接收到事件通知后，开始对系统画面的识别结果和显示结果进行更新。更新好画面之后，通知图像更新线程，同时自身挂起，并跳转到第一步（此时挂起的线程还有四个图像处理线程）。

各图像处理线程如图 7.2-7.4 所示：

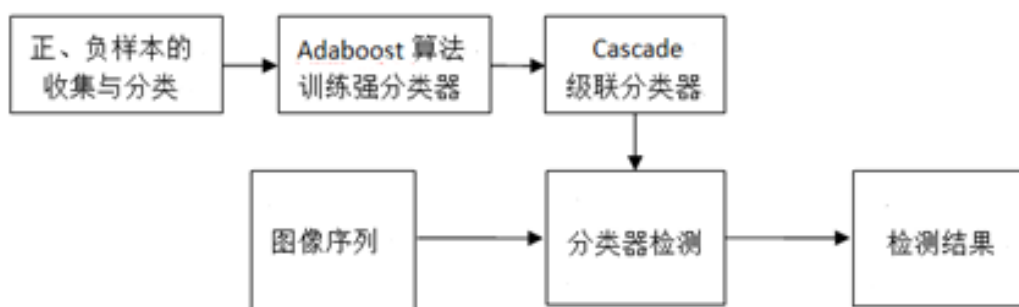


图 7.2 车辆识别流程

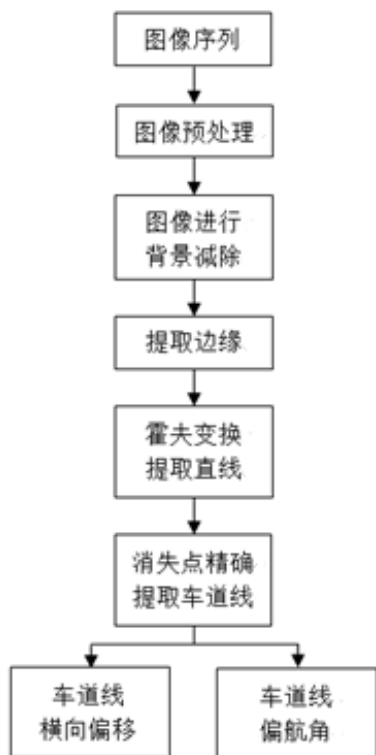


图 7.3 车道线偏移预警算法流程图

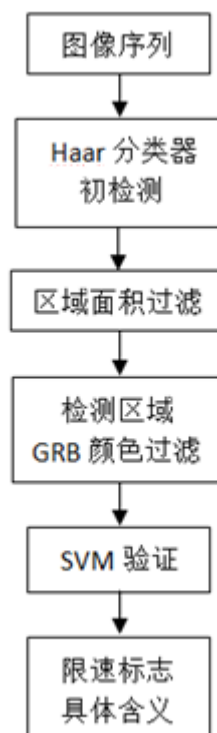


图 7.4 限速标志识别算法流程

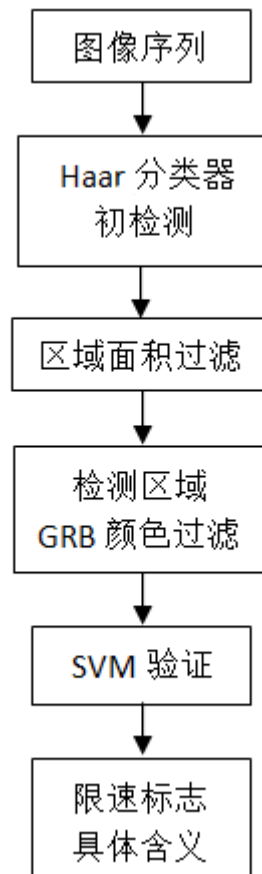


图 7.5 红绿灯识别算法流程

### 7.3 系统测试

在 Visual Studio 2013 开发平台上，通过 C++ 语言实现上文算法，并利用 MFC 设计智能辅助驾驶系统的用户界面，最终的系统框架如下图 7.6 所示：

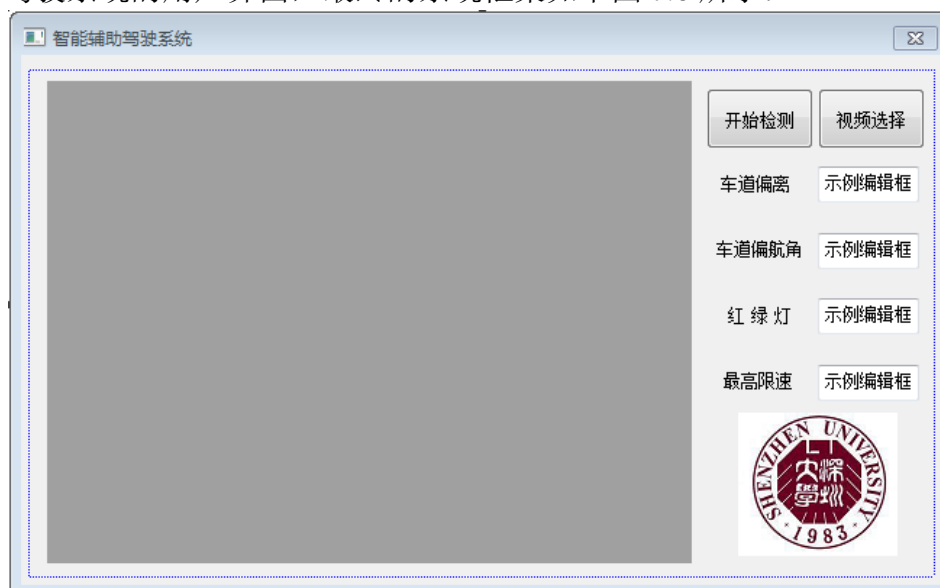


图 7.6 系统框架

用户界面包括两个按钮控件，“开始检测”按钮和“视频选择”按钮，分别用于启动智能辅助驾驶系统应用程序以及选择所检测的视频文件。下方有四个示例编辑框，用于实时显示视频流的检测识别结果。而视频预警画面则显示在左侧的灰色区域。

选择预先采集到的视频文件作为视频流输入，点击“开始检测”按钮进行实验，结果如下图 7.7 所示：

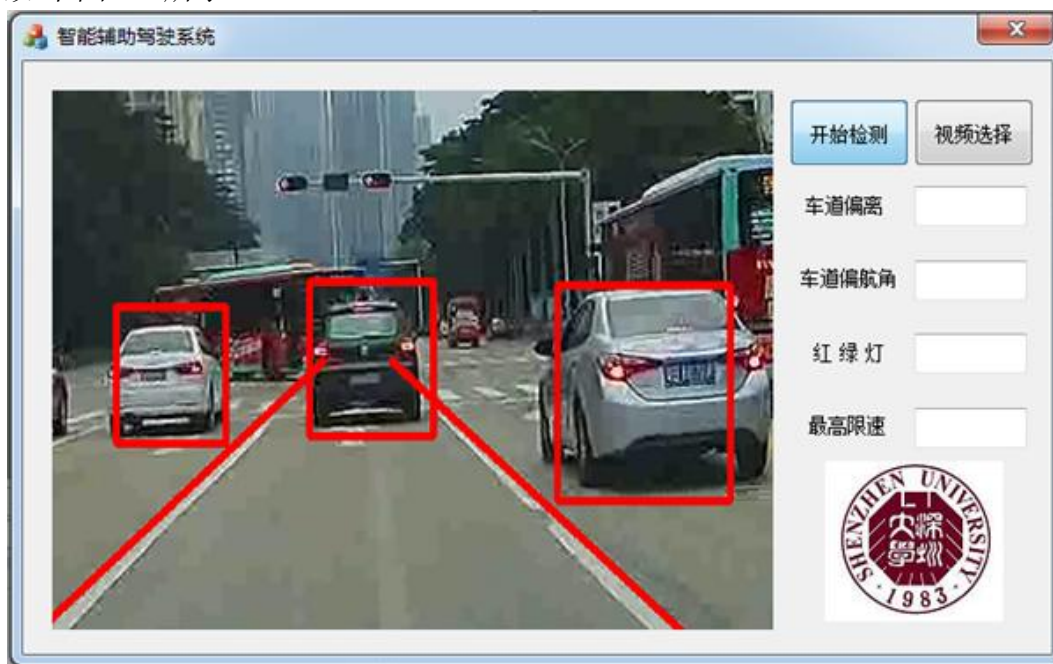


图 7.7 系统测试

从图 7.7 可以看到，本课题设计的智能辅助驾驶系统可以对复杂的实际道路场景进行检测，能够有效地检测出多目标多物体，并通过画面预警的方式通知驾驶员。

#### 7.4 本章小结

本章介绍了智能辅助驾驶系统的整体算法流程，并在 Visual Studio 2013 开发平台上，通过 C++ 语言实现上文算法，然后利用 MFC 设计实现了系统的用户界面。经过实际测试，系统的鲁棒性和实时性较好，满足一般性道路场景的使用需求。

## 8 工作总结及展望

### 8.1 总结

本文对基于机器视觉的车辆主动安全技术进行了详细研究，通过国内外各种先进算法的对比，设计了一种实用性强，鲁棒性好的车辆智能辅助驾驶系统，主要围绕车辆识别、车道线检测预警、限速标志和红绿灯路况分析四个部分展开。

在车辆识别部分：详细阐述了Haar-like特征的优点和Adaboost的算法实现。在本文中，通过2828幅车辆正样本图像以及7708副负样本图像，经Adaboost算法训练成18级Cascade级联分类器，对实际道路上的车辆进行有效的实时识别。

在车道线部分：主要研究了消失点有效提取车道线的方法。在感兴趣区域中对原始图像进行背景减除，过滤大量干扰信息，随后使用Canny算法提取目标边缘，通过Hough直线变换提取所有直线信息，利用车道线的斜率、位置特征排除干扰直线，借助消失点限制条件精确提取出车道线信息。最后，通过车辆的横向偏移以及偏航角计算，对车辆是否偏离车道线进行预警。实验表明，算法具有高效准确，鲁棒性强的优点。

在限速标志部分：提出了一种Haar+Adaboost分类器和HOG+SVM分类器串联识别的有效方法。首先通过459个限速标志正样本图像和737个负样本图像，训练成8级Cascade级联分类器，快速对原始图像进行非目标区域过滤；然后对过滤后的区域进行面积大小和红色像素点排除，选取出最后的候选区域；最后，通过134个限速标志正样本和737个负样本的HOG特征，训练成SVM分类器，对候选区域进行精确验证。实验表明，算法符合预期效果，能够快速排除非目标区域，又能有效识别候选区域，鲁棒性和实时性都较好。

在红绿灯部分：将红绿灯状态识别转换为红灯状态识别，并使用HOG特征进行红灯验证。首先对原始图像进行HSV颜色空间变换，提取感兴趣区域内的红色像素点，然后获取面积区间[40-70]之间的区域位置信息，在原始图像中进行裁剪并使用SVM分类器进行验证。实验表明，算法能够快速准确地识别红灯状态，满足智能辅助驾驶系统的一般性应用。

最后，利用多线程技术，将以上功能模块有机融合成智能辅助驾驶系统。通过预先收集好的视频文件进行测试，表明了系统作为车辆主动安全技术的可靠性，能够通过画面预警方式，为车辆的安全行驶保驾护航。

### 8.2 展望

1.在实际测试中，系统的视频处理速度约为14fps，距离标准的19fps还有一定距离，算法性能有待提升。在今后的工作中，可以在图像的像素点访问以及分类器样本数量上做提升。

2.本课题设计的辅助驾驶系统仅适用于白天路况，虽然通过HSV颜色空间变换和背景减除等图像预处理，对于光照变化具有较强的适应性，但是无法针对晚间的道路环境进行预警。在这方面，可以考虑通过红外线摄像头进行道路环境的检测。

3.本文算法的鲁棒性还有待提升。对于结构化标准化的道路环境，算法能够很好胜任检测识别任务；对于检测环境恶劣的情况，检测识别效果不够理想，不能满足主动安全的实际需求，还有待进一步优化。

## 参考文献

- [1] 巴布可夫. 道路条件与交通安全[M]. 景天然, 译. 上海:同济大学出版社, 1990.
- [2] 王占强. 道路交通事故 “达摩克利斯之剑”[J]. 世界汽车, 2009(9):32-33.
- [3] NOOKALAM, ESTOCHEN B. Minnesota, USA Intelligent Vehicle Initiative [C] // The Proceedings of IEEE Intelligent Vehicles Symposium.2002: 22-27.
- [4] Tons M, Doerfler R, Meinecke M M, Obojski M A. Radar sensors and sensor platform used for pedestrian protection in the EC-funded project SAVE-U. In: Proceedings of IEEE Intelligent Vehicles Symposium. Parma, Italy. IEEE, 2004.813-818.
- [5] Gavrilă D M, Giebel J, Munder S. Vision-based pedestrian detection: the protector system. In: Proceedings of IEEE Intelligent Vehicles Symposium. Parma, Italy. IEEE, 2004.13-18.
- [6] Broggi A, Bertozzi M, Fascioli A, Sechi M. Shape-based pedestrian detection. In: Proceedings of IEEE Intelligent Vehicles Symposium. Dearborn, USA, IEEE, 2000.215-220.
- [7] GAVRILA D M, GEIBEL J. Shape-Based Pedestrian Detection and Tracking [C] // The proceedings of IEEE Intelligent Vehicles Symposium. 2002: 8-14.
- [8] 孙即祥.图像处理[M].北京: 科学出版社, 2009, 11
- [9] P. Viola, M. J. Jones. Robust real-time face detection[J]. International journal of computer vision, 2004, 57(2):137-154.
- [10] Y. Freund, R. E. Schapire. Experiments with a new boosting algorithm[C]. ICML. 1996, 96:148-156.
- [11] Y. Wang, E. K. Teoh, D. Shen. Lane detection and tracking using B-Snake[J]. Image and Vision computing, 2004, 22(4):269-280.
- [12] N.Dalal, B. Triggs. Histograms of oriented gradients for human detection[C]. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1:886-893.
- [13] C. W. Hsu, C. C. Chang, C. J. Lin. A practical guide to support vector classification[J]. 2003.
- [14] 索爱黎. 形态学图像分析原理与应用[M].北京: 清华大学出版社, 2008, 6



## 致谢

在论文完成之际，我要衷心地感谢我的论文指导老师徐颖老师。在我的论文写作过程中，徐老师给予我耐心的指导和帮助，从论文选题、构思、撰写开题报告到正式写完整篇论文，徐老师都给出了宝贵的意见和建议，在此我向老师表示真诚的感谢。

同时，我要感谢参加论文评审和答辩的各位老师，谢谢他们不辞辛劳、无私奉献、提出宝贵意见。

最后，我要感谢我的母校深圳大学的培育和熏陶，在这里四年的学习和生活让我获益良多，我将永远难忘在母校的时光。