

# 基于机器视觉的驾驶员危险驾驶检测系统的设计与实验研究

张贝宁, 郜健铭, 王靖涵, 张 敬, 顿媛雅, 许留洋

(周口师范学院, 河南 周口)

**摘 要:**本文基于机器视觉技术设计了驾驶员危险驾驶检测系统,该系统的硬件部分包括 DSP 芯片、摄像头和音频芯片等;软件部分包括图像采集和预处理模块,目标检测模块和疲劳驾驶检测模块。实验结果表明,该系统的人脸检测准确率较高,疲劳驾驶检测的准确度达到 86%,较好地完成了驾驶员危险驾驶行为的检测任务。

**关键词:**机器视觉;驾驶员危险驾驶检测系统;人脸检测;图像预处理

中图分类号:TP391.41;U463.6

文献标识码:A

文章编号:2096-4390(2024)03-0048-04

疲劳驾驶是诱发交通事故的重要因素,如果能够及时发现驾驶员出现疲劳驾驶时准确检测并进行语音提醒,可以帮助驾驶员集中注意力,从而降低交通事故的发生率。基于机器视觉的驾驶员危险驾驶检测系统,利用车内摄像头实时采集驾驶员的头部图像信息,并通过人脸检测算法对驾驶员的眼睛状态进行识别,通过算法计算出是否处于疲劳状态。当判定为疲劳驾驶时,通过语音提示驾驶员及时休息,切实保障了行车安全。

## 1 驾驶员危险驾驶检测系统的硬件设计

### 1.1 系统的硬件结构

该系统对图像处理速度要求较高,为满足系统的实时性要求采用了 TMS320C6748 型 DSP 芯片,该芯片能减少卷积运算和信号滤波的时间,显著提高了数字信号的运算速率,满足了实时检测的需求。从组成

架构上来看,驾驶员危险驾驶检测系统主要由图像采集、图像存储和图像处理 3 个核心模块构成,除此之外还有配套的电源模块、语音报警模块等,硬件的整体架构见图 1。

图像采集模块的核心设备是前端摄像头,用于收集汽车驾驶员的图像信息,将收集信息进行 A/D 转换后利用 VPIF 接口输入到 DSP 芯片中。CPU 对转换后的信息进行处理,利用 EDMA 控制器将处理后的结果传输至存储器中。根据分析结果,如果系统判定驾驶员存在危险驾驶行为,则利用 McASP 音频输出接口控制报警器发出警报,提醒驾驶员。

### 1.2 图像采集模块设计

该模块的主要作用是采集驾驶员的实时图像,因此对摄像头的像素、灵敏度、动态范围等参数有较高的要求。本系统选用带图像传感器的 MT9V034 型摄像头,输出图像的像素为  $760 \times 500$ ,最大时钟频率 30 MHz,在光线较暗的环境下也能清晰成像。在确定摄像头的型号与参数后,还要寻找合适的安装位置。为了拍摄到完整的驾驶员头部图像,将摄像头放置在驾驶员前方 35~40 cm 处,仰角为 60~70°,摄像头焦距在 15~18 mm 之间。

### 1.3 处理器的选择

处理器的性能决定了驾驶员危险驾驶检测系统

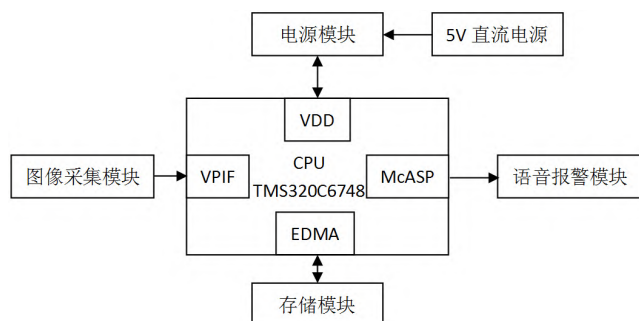


图 1 系统硬件结构图

**基金项目:**周口师范学院大学生创新创业训练计划项目(202310478009)资助。

**作者简介:**张贝宁(2004-),女,本科,研究方向:电子设计。

的应用效果,本系统的处理器(CPU)选择 TI 公司生产的 TMS320C6478 芯片,该芯片拥有 C674X 内核结构,内置 64 个 32 位的通用寄存器,支持 C674x 系列指令集和 C++ 语言,降低了系统的开发难度。提供 VPIF 视频接口和 McSAP 语音接口,提高了图像、语音、视频等多种类型信息的传输能力。配备的 EMDA 增强型访问控制器,对提高系统的整体运行效能有积极帮助<sup>[1]</sup>。

### 1.4 语音报警模块设计

在系统检测到驾驶员处于疲劳驾驶状态时,播报警示语音提醒驾驶员集中注意力或尽快停车休息。该模块的核心硬件是 TLV320AIC3106 音频芯片,共有 3 个 I/O 口。其中 2 个为音频输入口,分别是音频模拟输入口、麦克风输入口;1 个音频模拟输出口。在系统启动运行后,把报警语音文件通过音频模拟输入口传输到芯片中,并利用内置的 A/D 转换模块将音频信号变成电信号。当系统检测到驾驶员处于疲劳驾驶状态后,触发语音报警功能。此时音频芯片控制音频模拟输出口工作,再重新将电信号转变为身心信号,利用外置扬声器播报语音。

## 2 驾驶员危险驾驶检测系统的软件设计

### 2.1 系统的软件结构

本文设计的驾驶员危险驾驶检测系统,软件部分主要由图像采集、图像预处理、人脸检测和危险驾驶判定等核心模块构成,软件结构见图 2。

在硬件初始化后,软件部分同步运行,图像采集模块收集驾驶员的图像,并执行一个判断程序“是否成功获取一帧图像?”如果判定结果为“否”,则返回初始化程序,重新采集图像信息;在判定结果为“是”后,进入图像预处理环节。预处理后的图像有两种用途,一种是进行人脸检测,通过人眼定位、提取眼睛闭合度、计算 Perclos 值等一系列处理后,完成危险驾驶的判定;另一种是用于分心驾驶判定。如果有疲劳驾驶和分心驾驶行为,则进行语音告警。

### 2.2 基于视觉特征的疲劳驾驶检测

#### 2.2.1 图像预处理

在汽车行驶过程中,由于外界光线的变化以及车辆或摄像头的抖动等原因,采集到的图像可能会出现噪声、失真等情况,导致图像的清晰度变差,影响检测结果的准确性。为了避免此类问题,在设计驾驶员危险驾驶检测系统的软件时,加入了图像预处理模块,具体方法有两种:

其一是图像滤波法,已知图像的能量主要分布于幅度谱的中低频段,噪声主要分布于高频段,通过图像滤波有助于降低高频成分幅度,达到去除噪声的效果。同时,图像滤波还能尽可能地保留图像的细节特征,不影响图像本身的精度。其二是光照补偿法,主要是解决因为光照过强或过暗引起的图像失真问题<sup>[2]</sup>。目前常用的补偿算法是直方图均衡法,计算待处理图像的灰度分布直方图,然后通过非线性变换让图像中的像素值重新排列,让像素从集中分布变为均匀分布,达到光照补偿效果。

#### 2.2.2 人脸检测

对于预处理完毕后的图像,使用 AdaBoost 算法做进一步分析,提取特征并检测图像中的人脸。人脸检测的实现大体可分为两个步骤,即分类器训练和图像

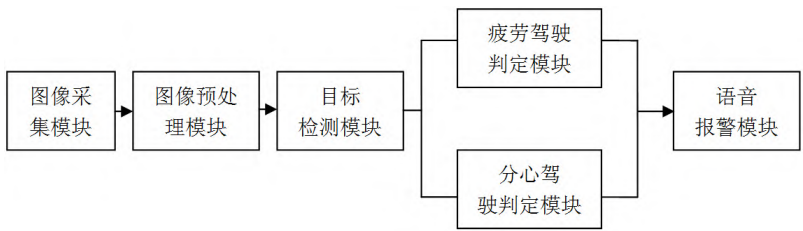


图 2 系统软件结构图

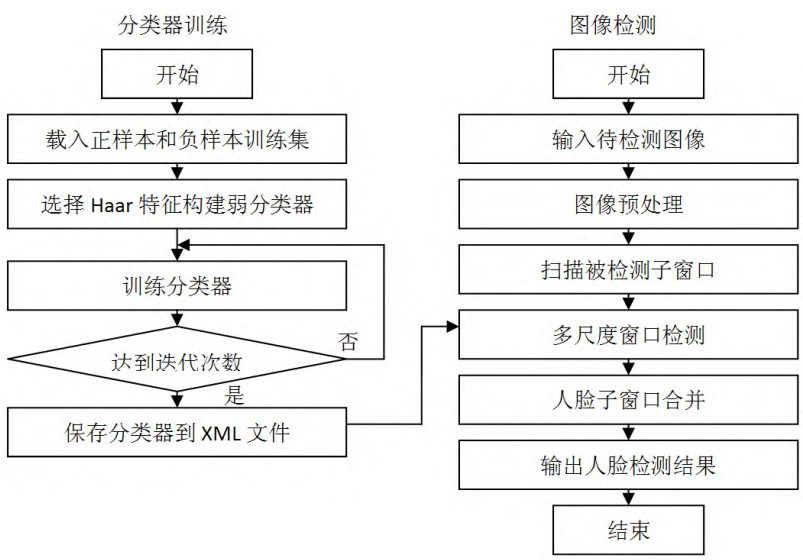


图 3 人脸检测流程图

检测,流程见图 3。

样本质量在一定程度上决定了分类器对人脸的识别准确度,为此必须提供足够多的样本图像数据进行分类器的训练<sup>[3]</sup>。在选择样本时,收集 10 000 张人脸图像作为正样本集,同时收集 6 000 张非人脸图像(如动物、汽车、风景等)作为负样本集。利用正负样本集训练分类器。利用 Haar-like 特征构建弱分类器,其数学模型为:

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

上式中, $x$ 表示子窗口图像, $f(x)$ 表示图像的 Haar-like 特征值, $p$ 表示不等号的方向, $\theta$ 表示阈值。如果 Haar-like 特征值在阈值内,则输出结果为“1”,表示图像为人脸;反之,如果 Haar-like 特征值不再阈值内,则输出结果为“0”,表示图像为非人脸,由此实现人脸检测。按照上述方法对分类器进行 500 次迭代训练。执行一个判断程序“是否达到迭代次数?”,在迭代次数达到 500 次后完成训练,将结果保存为 XML 文件,并利用训练好的分类器进行人脸检测。

### 2.2.3 疲劳驾驶判定

当驾驶员长期驾驶并出现驾驶疲劳时,会出现眨眼频率升高、眼睛闭合速度变慢、闭眼时间过长等生理特征<sup>[4]</sup>。基于机器视觉的疲劳驾驶检测就是在识别人脸的前提下,对人眼的状态进行识别,从而利用相关参数判断驾驶员是否处于疲劳状态。本文采用了以统计像素值为基础的计算眼睛张开度的方法,计算公式如下:

$$f(i) = \begin{cases} 0, & H(i) \leq H_m \times 1/2 \\ 1, & H(i) > H_m \times 1/2 \end{cases}$$

上式中, $H(i)$ 表示第  $i$  帧图像中眼睛的睁开度, $H_m \times 1/2$ 表示判断眼睛睁闭的阈值。如果  $f(i)$  为 0,表示眼睛闭合;如果  $f(i)$  为 1,表示眼睛睁开。

在识别眼睛状态的基础上,还要提取疲劳参数、进行疲劳判定。现阶段常用的疲劳参数提取方法有两种类型,分别是基于脉搏、心率的生理参数法和基于眨眼频率、眼睛闭合时间的视觉特征参数法,本文选择后一种方法,并将单位时间内眼睛闭合时间百分比(Perclos)作为判定指标<sup>[5]</sup>。Perclos 与疲劳程度的映射关系如图 4 所示。

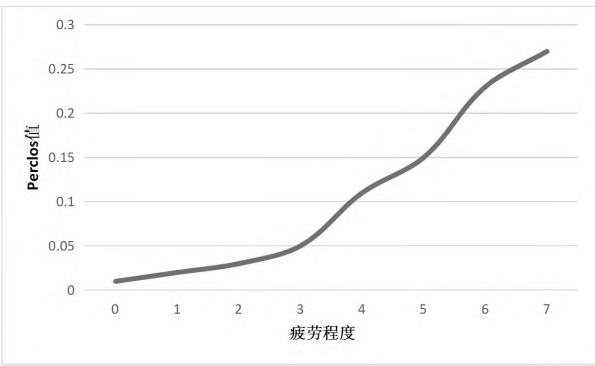


图 4 Perclos 与疲劳程度的关系曲线

由图 4 可知,Perclos 值越高,说明疲劳程度越高,该值可通过以下公式求得:

$$Perclos = \frac{M}{m} \times 100\%$$

上式中, $M$ 表示 60 s 内眼睛闭合图像的帧数, $m$ 表示 60 s 内系统采集到的图像总帧数。

## 3 驾驶员危险驾驶检测系统的实验探究

### 3.1 系统开发平台

本系统的硬件部分包括 TMS320C6748 芯片、MT9V034 摄像头、LCD 显示屏,以及 1 台 GYX1060 计算机;软件部分使用 Matlab2018 进行系统开发,使用 OpenCV 数据库,运行系统为 DSP/BIOS。收集 1 000 张(700 张为人脸、300 张为非人脸)图像和 50 段视频进行检测。

### 3.2 系统实验结果

#### 3.2.1 人脸检测结果

将 1 000 张图像输入到驾驶员危险驾驶检测系统中,测试人脸检测准确率,结果见表 1。

表 1 人脸检测分类器测试结果

图像类别	图像数量	检测准确率
非正面人脸	113	94.2%
多人脸	206	91.8%
有遮挡	121	96.1%
光线过强或过暗	260	84.4%

由表 1 数据可知,对于非正面人脸、多人脸和有遮挡的图像,分类器的检测准确率较高,基本维持在 90%以上;而光照过强或过暗的图像,检测准确率相对较低,这表明光线强弱是影响人脸检测准确率的核心因素。

#### 3.2.2 疲劳驾驶判定结果



将 50 段驾驶员的驾车视频（时长 40~100 s）作为检测样本,利用该系统进行疲劳判定,结果见表 2。

表 2 疲劳驾驶判定结果

视频序号类别	Perclos 值	眨眼频率(次/min)	疲劳判定结果
1 正常驾驶	0.061	13	正常驾驶
5 疲劳驾驶	0.206	15	疲劳驾驶
9 正常疲劳	0.086	14	轻度疲劳
23 轻度疲劳	0.056	26	正常驾驶
26 疲劳驾驶	0.377	11	疲劳驾驶
38 正常驾驶	0.113	16	正常驾驶

参与检测的 50 段视频中,有 43 段视频的疲劳判定结果准确,准确率为 86%;其他 7 段视频中,有 5 组属于正常驾驶,系统判定结果为轻度疲劳,这意味着在系统设计时需要适当提高疲劳驾驶的判定阈值。

### 结束语

基于机器视觉的驾驶员危险驾驶检测系统以前端摄像头提供的图像作为分析对象,检测到人脸后提取眼部参数,然后求解 Perclos 值,进而推测出驾驶员的驾驶状态。如果判定为疲劳驾驶则进行语音提示,使驾驶员集中注意力。该系统判定结果的准确率可以

达到 86%,应用效果符合设计预期。

### 参考文献

- [1]黄志超,赵红梅,陈奕迅,等.基于面部识别的驾驶员疲劳危险驾驶检测系统[J].机电工程技术,2021(12):40-42.
- [2]郭小玉,杨蕾,鹿德源.基于 JetsonXavierNx 的智能驾驶员行为检测系统设计[J].现代信息科技,2022(9):114-115.
- [3]李孟成,罗甜,张琰,等.基于 YOLOv5 的疲劳与危险驾驶行为检测系统研究[J].内蒙古科技与经济,2023(6):107-110.
- [4]林静怡,史晓颖.双轮车危险驾驶行为智能管控系统[J].计算机时代,2022(10):100-102.
- [5]许旻,罗紫琳.基于深度学习的危险驾驶状态检测预警系统设计[J].无线互联科技,2022(12):50-52.

# Design and Experimental Research of Dangerous Driving Detection System for Drivers Based on Machine Vision

Zhang Beining, Gao Jianming, Wang Jinghan, Zhang Jing, Dun Yuanya, Xu Liuyang  
(ZhouKou Normal University, Zhoukou, China)

**Abstract:** This paper designs a dangerous driving detection system for drivers based on machine vision technology. The hardware part of the system includes DSP chip, camera and audio chip. The software includes image acquisition and preprocessing module, target detection module and fatigue driving detection module. The experimental results show that the system has high face detection accuracy, and the accuracy of fatigue driving detection reaches 86%, which can well complete the task of dangerous driving behavior detection.

**Key words:** machine vision; dangerous driving detection system for drivers; face detection; image preprocessing