



北京石油化工学院
BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY

北京石油化工学院

人工智能研究院

课程设计任务书

——驾驶员疲劳检测系统设计

课程名称：计算机视觉课程设计

开设院系：人工智能研究院

一、题目

驾驶员疲劳检测系统设计

题目概述：

疲劳驾驶是道路交通安全的重大隐患。据统计，约有 20%-30% 的交通事故与驾驶员疲劳有关。本课程设计要求学生设计并实现一个基于计算机视觉的驾驶员疲劳检测系统，通过实时分析驾驶员的面部特征（眼睛闭合程度、打哈欠频率等），判定疲劳状态并及时预警，从而有效预防交通事故的发生。

二、已知技术参数和设计要求

（一）技术参数

1. 编程语言：Python 3.8+
2. 核心库：OpenCV、Dlib、NumPy
3. 开发环境：推荐使用 PyCharm、VS Code 或 Jupyter Notebook
4. 版本控制：建议使用 Git 进行代码管理
5. 模型文件：shape_predictor_68_face_landmarks.dat

（二）系统功能要求

1. 人脸检测：从视频流中准确定位驾驶员人脸区域
2. 关键点提取：检测 68 个面部关键点，重点提取眼睛和嘴部特征
3. 特征计算：计算眼睛纵横比 (EAR)、嘴部纵横比 (MAR)
4. 疲劳判定：根据 PERCLOS 准则判定疲劳状态（清醒/轻度疲劳/重度疲劳）
5. 实时告警：检测到重度疲劳时发出视觉和听觉告警
6. 结果展示：可视化展示检测状态、特征值和疲劳程度

（三）设计要求

基本要求（60–75 分）

- 能完成基本的人脸检测和关键点提取
- 能实现简单的 EAR 计算
- 能根据固定阈值判定疲劳状态
- 代码能运行，有基本输出

良好要求（76–85 分）

- 人脸检测准确率达到 90% 以上
- 实现 PERCLOS 疲劳判定算法
- 实现打哈欠检测功能
- 代码结构清晰，有详细注释
- 有基本的可视化界面

优秀要求（86–95 分）

- 整体检测准确率 90% 以上
- 处理复杂场景（光照变化、角度偏转）
- 实现自适应阈值调整
- 有完整的 GUI 界面或 Web 界面
- 实时视频处理流畅（FPS ≥ 15 ）

卓越要求（96–100 分）

- 算法有创新改进（如融合深度学习方法）
- 完整的测试报告和性能分析
- 支持多人脸检测或头部姿态估计
- 可部署到嵌入式设备（如树莓派）

三、工作量

（一）代码实现部分

模块名称	工作量说明
人脸检测模块	使用 Dlib HOG+SVM 检测人脸，处理多人脸情况（约 100 行代码）
关键点提取模块	使用 68 点回归树模型提取面部关键点（约 80 行代码）
特征计算模块	实现 EAR、MAR 等特征计算功能（约 150 行代码）
疲劳判定模块	实现 PERCLOS 准则和状态跟踪器（约 200 行代码）
可视化模块	实现检测结果的可视化展示（约 120 行代码）
主程序	整合各模块，实现完整检测流程（约 150 行代码）

预计代码总量：约 800-1000 行（不含注释和空行）

（二）文档撰写部分

文档名称	内容要求	篇幅
课程设计报告	系统设计、算法实现、PERCLOS 原理、测试结果、问题分析	8000-10000 字
使用说明书	环境配置、模型下载、功能说明、使用示例	2000-3000 字
测试报告	测试数据、性能指标（准确率/召回率/F1）、案例分析	3000-5000 字

（三）测试数据准备

- 自行录制测试视频至少 5 段（包含清醒、疲劳状态）
- 邀请 2-3 位不同人员协助录制测试数据
- 准备不同光照条件的测试场景
- 记录真实的疲劳状态标签

四、工作计划

（一）整体安排

阶段	时间	主要任务
第一阶段	第 1-3 天	环境配置、Dlib 安装、人脸检测与关键点提取实现
第二阶段	第 4-6 天	EAR/MAR 特征计算算法实现与调试
第三阶段	第 7-8 天	PERCLOS 疲劳判定算法实现
第四阶段	第 9-10 天	疲劳状态跟踪器实现与参数优化
第五阶段	第 11 天	可视化界面开发、系统集成
第六阶段	第 12-13 天	测试数据准备、系统测试与性能评估
第七阶段	第 14 天	文档撰写、答辩准备

（二）详细计划

第一阶段：环境配置与基础模块（第 1-3 天）

天数	任务	具体内容
第 1 天	环境配置	安装 Python、OpenCV、Dlib，配置开发环境，下载模型文件
第 2 天	人脸检测	学习 Dlib 人脸检测器，实现多人脸检测，选择最大人脸
第 3 天	关键点提取	实现 68 点关键点检测，绘制面部关键点可视化

第二阶段：特征计算（第 4-6 天）

天数	任务	具体内容
第 4 天	EAR 计算	实现眼睛纵横比 (EAR) 计算，提取左右眼关键点
第 5 天	MAR 计算	实现嘴部纵横比 (MAR) 计算，检测打哈欠行为
第 6 天	参数调试	调整 EAR/MAR 阈值，测试不同人员特征值范围

第三阶段：疲劳判定（第 7–8 天）

天数	任务	具体内容
第 7 天	PERCLOS 实现	实现 PERCLOS 准则计算，设计时间滑动窗口
第 8 天	状态机设计	实现清醒/轻度疲劳/重度疲劳三态判定逻辑

第四阶段：状态跟踪与优化（第 9–10 天）

天数	任务	具体内容
第 9 天	状态跟踪器	实现疲劳状态跟踪类，记录眨眼次数、打哈欠次数
第 10 天	参数优化	调整检测阈值，优化误报率和漏报率

第五阶段：可视化与集成（第 11 天）

任务	具体内容
可视化开发	实时显示面部关键点、EAR/MAR 值、疲劳状态、PERCLOS 曲线
告警功能	重度疲劳时发出告警（红色边框、文字提示、可选声音）
系统集成	整合各模块，实现完整的实时检测流程

第六阶段：测试与评估（第 12–13 天）

天数	任务	具体内容
第 12 天	数据准备	录制测试视频，标注真实状态（清醒/疲劳）
第 13 天	性能评估	计算准确率、精确率、召回率、F1 分数，记录 FPS

第七阶段：文档撰写（第 14 天）

任务	具体内容
课程设计报告	撰写系统概述、算法原理、实现细节、测试结果、问题分析
使用说明书	编写环境配置指南、功能说明、使用示例
测试报告	整理测试数据、性能指标、典型测试案例
答辩准备	准备演示 PPT、演示视频

五、提交要求

（一）代码部分

- 完整的源代码（含详细注释）
- requirements.txt 依赖文件
- README.md 项目说明文档
- config.py 配置文件
- models/目录（不含模型文件，注明下载地址）

（二）文档部分

- 课程设计报告（docx/pdf 格式）
- 使用说明书
- 测试报告（含性能指标数据）

（三）演示部分

- 测试视频集（至少 5 段，含清醒/疲劳状态）
- 检测结果截图
- 演示视频（3-5 分钟，展示完整功能）
- 答辩 PPT

六、参考资源

（一）技术文档

1. OpenCV 官方文档: <https://docs.opencv.org/>
2. Dlib 官方文档: <http://dlib.net/>
3. NumPy 官方文档: <https://numpy.org/doc/>
4. 模型下载地址: http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2

（二）核心参考文献

1. Soukupová T, Čech J. Eye blink detection using facial landmarks[J]. 2016.
2. King D E. Dlib-ml: A Machine Learning Toolkit[J]. Journal of Machine Learning Research, 2009.
3. Kazemi V, Sullivan J. One millisecond face alignment with an ensemble of regression trees[C]//CVPR. 2014.

（三）参考书籍

1. Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media.
2. Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer.
3. Gonzalez, R. C., & Woods, R. E. (2018). *Digital image processing*. Pearson.

（四）开源项目参考

项目名称	语言	特点
dlib	C++/Python	人脸检测与 68 点关键点提取
OpenFace	Python/Torch	面部行为分析工具包
MediaPipe Face Mesh	Python/JS	谷歌轻量级人脸网格检测
MTCNN	Python/PyTorch	多任务级联人脸检测

七、指导教师签字

指导教师签字：

日期：_____

教学系（教研室）主任签字：

日期：_____

注：本任务书一式两份，学生与教学系各执一份