睿行安途—智能驾驶安全中枢

摘要

国家统计局的调查显示，全国每年因交通事故死亡的人数已超过 6 万人次，其中因疲劳驾驶导致的事故占死亡事故的 30%，尤其是在重大交通事故中这一比例更是急剧增大至 70%。且在货车运输业平台上司机人数整体呈上升趋势，平均司机增长率约为 20%，但同期我国货运总量增长率为 5%、公路货运总量增长率为 3%，且整体年龄偏大容易发生疲劳驾驶。为缓解日趋严重的疲劳驾驶问题，我们设计了基于 RK3588 平台的"睿行安途"智能驾驶安全系统，其核心是通过多模态感知与智能交互实现驾驶场景的主动安全防护。软件层创新部署双异步互联处理引擎，视觉分析引擎通过轻量化**卷积神经网络**实现毫秒级生物特征捕捉，在驾驶员头部 60° 偏转范围内维持≥95%的疲劳识别准确率；语音交互引擎则采用 RKLLM 工具链对 **DeepSeek-1.5B 大模型**进行层融合量化，结合 **NPU 加速流式 ASR 与双缓冲 TTS 技术**，实现≥96%指令识别率与毫秒级全链路响应。且云端同步群体疲劳数据，实现高危时段预警和驾驶状态分，填补了全天候车载安全中枢空白。

技术创新聚焦于三层协同架构的动态耦合以解决当下**痛点**：

1. **毫秒级响应**：现有方案需联网调用云端 AI 导致高延迟→本作品本地部署双模型（疲劳检测+大语言模型）与硬件加速。

2. **边缘计算部署**：现有方案商用交通山区/隧道信号中断致安全失效→本作品离线全功能运行。

3. **模型互联与云端策略闭环管理**：驾驶行为信息在本地与 AI 互联，实现功能集成。数据经华为云解析，最终通过可视化看板输出量化指标，构建了从硬件防护、算法决策到云端优化的全栈式技术链条，解决了当前疲劳识别方案结果处理方式单一和管理困难的问题。

4. **模块化扩展设计**：现有方案难以实现对落后载具的升级→本作品模块化设计可内部集成可外部加装扩展，根据用户需求确定配置。

第一部分　作品概述

1.1 功能与特性

"睿行安途"智能驾驶安全中枢是一款基于 RK3588 开发板的端云协同系统[5]。其核心功能融合多模态感知与智能决策，具体表现为：

疲劳监测体系：通过视觉分析实时捕捉驾驶员面部特征[6]进行综合研判，构建分级预警机制。检测到异常状态时，系统同步触发警报，并具备智能反馈机制。

离线语音交互：模块采用轻量化设计，在无网环境下实现全链路本地化处理[7]：语音指令经实时解析后，由本地大模型完成意图理解与决策响应，最终通过低延迟语音播报技术反馈结果。

端云协同生态：将本地实时防护与云端长期优化深度结合：端侧持续上传结构化驾驶行为数据，云端通过批流混合计算[8]生成群体安全特征，最终以可视化看板形式输出驾驶习惯分析报告，形成安全策略的动态闭环。

嵌入式硬件平台：采用模块化设计，通过定制化结构实现高适配性与物理防护，确保复杂车载环境下的稳定运行。

图 1.1 系统适配示意图

1.2 应用领域

"睿行安途"智能驾驶安全中枢的应用领域聚焦交通运输行业的主动安全升级，其端云协同架构覆盖三大核心场景：

在**商用车驾驶监控领域**，系统通过实时视觉分析驾驶员状态，构建毫秒级疲劳预警体系。当检测到异常状态时，同步触发警报，并结合华为云平台长周期数据分析，助力物流企业优化驾驶员排班策略与安全管理标准，显著降低长途运输事故率。

针对**特种车辆作业场景，**离线语音交互模块在无网环境下实现全链路本地响应。驾驶员通过自然语音指令直接操控车载设备，本地大模型完成意图解析后经低延迟语音反馈，结合模块化设计，保障工程机械、抢险车辆在振动、高噪等恶劣环境中的稳定交互。

在**驾驶能力评估领域**，系统依托云端看板量化分析学员操作行为，通过指标生成结构化评估报告，精准定位应急反应缺陷与注意力分配问题。

1.3 主要技术特点

"睿行安途"智能驾驶安全中枢的核心技术特点体现为三层协同创新架构：

边缘计算层深度融合硬件加速与模型优化，充分发挥 RK3588 芯片的端侧算力。视觉链路采用轻量化卷积神经网络模型[9]，实现毫秒级人脸检测与关键点定位；语音链路通过 RKLLM 工具链对 DeepSeek 大模型进行层融合量化，保障离线场景下的实时语义解析。双引擎并行处理能力与耦合互联，确保处理结果的高效利用。

通信交互层创新采用 ZeroMQ 松耦合架构[10]，构建异步消息总线系统。流式 ASR 识别结果经 PUB/SUB 模式广播至本地 LLM 模块，支持端云弹性部署；语音反馈通道集成双缓冲 TTS 技术[11]，通过乒乓式缓冲切换机制消除传统语音合成的卡顿延迟，实现自然流畅的交互体验。

1.4 主要性能指标

生物特征识别能力：

表 1.1 生物特征识别性能指标

感知指标	测试条件	性能数据	技术支撑
疲劳识别准确率	头部水平偏转 0° -60°	≥95%	关键点动态校准算法
语音指令识别率	车载环境噪声≤70dB	≥96%	NPU 加速流式 ASR 模型

响应时效性：

表 1.2 响应时效性性能指标

时序指标	触发场景	延迟上限	优化技术
视觉分析延迟	1080P@30fps 视频流处理	35ms	CNN 模型层融合与 NPU 硬件加速
语音交互全链路响应	10 字口语指令	2000ms	ZeroMQ 零拷贝传输+双缓冲 TTS
云端数据同步时效	Wifi 网络环境	2s	JSON 压缩传输与断点续传机制

1.5 主要创新点

1. 全链路毫秒级本地响应以解决云端依赖的高延迟痛点：

双异步处理引擎轻量化 CNN 模型和 NPU 加速流式 ASR + 双缓冲 TTS 技术。

2. 强鲁棒性离线边缘计算以解决山区/隧道信号中断致安全失效痛点：

本地视觉引擎支持无网环境疲劳识，语音引擎通过 AI 融合量化实现离线交互。

3. 端云协同策略闭环管理以解决结果单一、管理困难痛点：

上传云端生成安全特征与驾驶评分，反哺端侧预警阈值校准，形成“感知-优化-决策”闭环。

4. 模块化扩展兼容设计以解决老旧车辆升级难痛点。

1.6 设计流程

"睿行安途"智能驾驶安全中枢的设计流程构建了端云协同的三层架构：硬件层集成多模态传感器与定制化防护结构，建立感知基础；软件层部署双引擎——

视觉分析引擎实现状态捕捉，语音交互引擎通过松耦合架构完成本地决策；云端形成闭环优化，将驾驶行为特征反哺端侧策略校准。全流程经场景适应性迭代验证，最终形成"感知-响应-进化"的动态防护体系，在嵌入式平台上实现毫秒级安全响应。具体流程如下图所示：

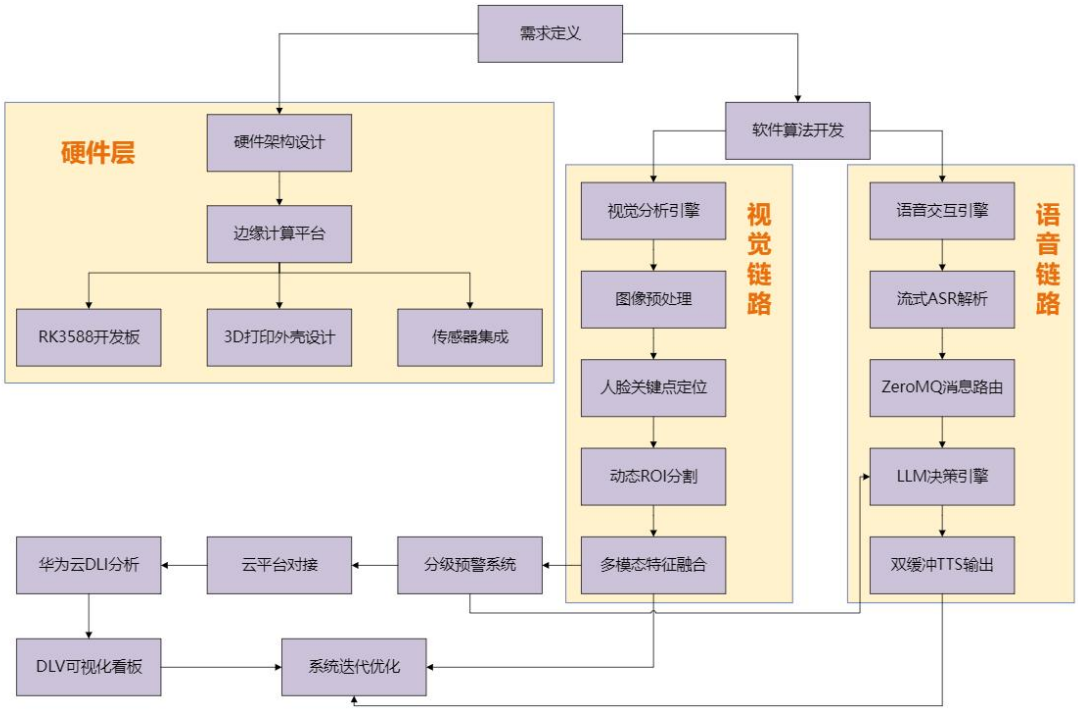

图 1.2 设计流程图

第二部分　系统组成及功能说明

2.1 整体介绍

基于图 2.1 展示的系统架构，本作品通过 RK3588 开发板本地化部署双核心安全引擎：

左侧疲劳识别链路以 1080P 摄像头实时采集图像，经阈值调整与灰度图降维（640×480 分辨率）预处理后，由卷积神经网络模型执行人脸检测、关键点定位及 ROI 分割，通过连续帧的 EAR 计算实现眼部特征提取，在 QT 图形界面完成毫秒级疲劳判定与状态分析，最终带时间戳的关键事件以 JSON 格式经无线网络上传至华为云 DLI 平台，结合驾驶习惯与长时疲劳分析生成 DLV 可视化看板。右侧语音交互链路通过拾音模块获取 PCM 音频，经流式 ASR 解析文本后输入本地离线 LLM 实现指令理解，支持车内设备控制与决策辅助，其响应经由双缓冲

TTS 技术生成语音数据，通过 ZeroMQ 松耦合通信协议传递至声卡实时播放。两条链路协同运行于开发板端侧，在无网条件下仍保障毫秒级响应，成“本地 AI 实时防护-云端长周期优化”的双闭环安全生态

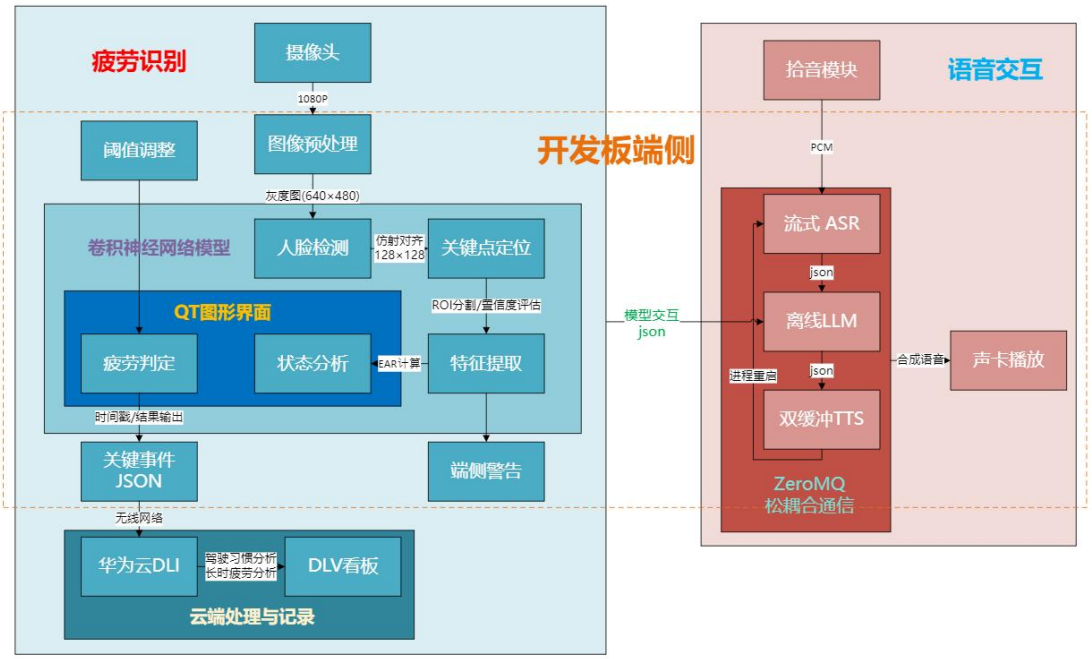

图 2.1 智能驾驶安全系统模块图

2.2 硬件系统介绍

2.2.1 硬件整体介绍；

本作品是一套基于 RK3588 ELF2 高性能开发板构建的智能车载集成系统（如图 2.2 所示）。开发板作为核心计算单元，外部配备了定制的 3D 打印保护外壳，不仅提供物理防护，还增强了整体结构的稳定性与工业美感。系统通过模块化扩展连接三大关键外设：高清摄像头实时捕捉驾驶员面部信息，用于本地化部署的机器视觉分析；高灵敏度话筒精准接收语音指令，实现自然交互；带音响的 HDMI 屏幕则承担双重重任——既作为车机系统的可视化交互界面，又通过内置扬声器输出语音反馈与预警提示。

整套硬件设计突出强兼容性与场景适应性：摄像头支持多路视频流解析，话筒适配主流降噪协议，HDMI 屏幕兼容多种分辨率输出，确保在 Android/Linux 等不同系统中无缝运行。依托 RK3588 芯片的 6TOPS AI 算力，系统在本地同时

部署了双 AI 引擎——人脸识别疲劳检测模型持续监测驾驶员状态，发现异常即时触发本地声光警报并同步上传云端生成群体安全预警；大语言模型则赋能语音控制系统，通过自然语言理解实现车内设备控制、实时决策辅助及联网信息检索。这种"端-云协同"架构显著提升了驾驶安全冗余度与座舱智能化水平。

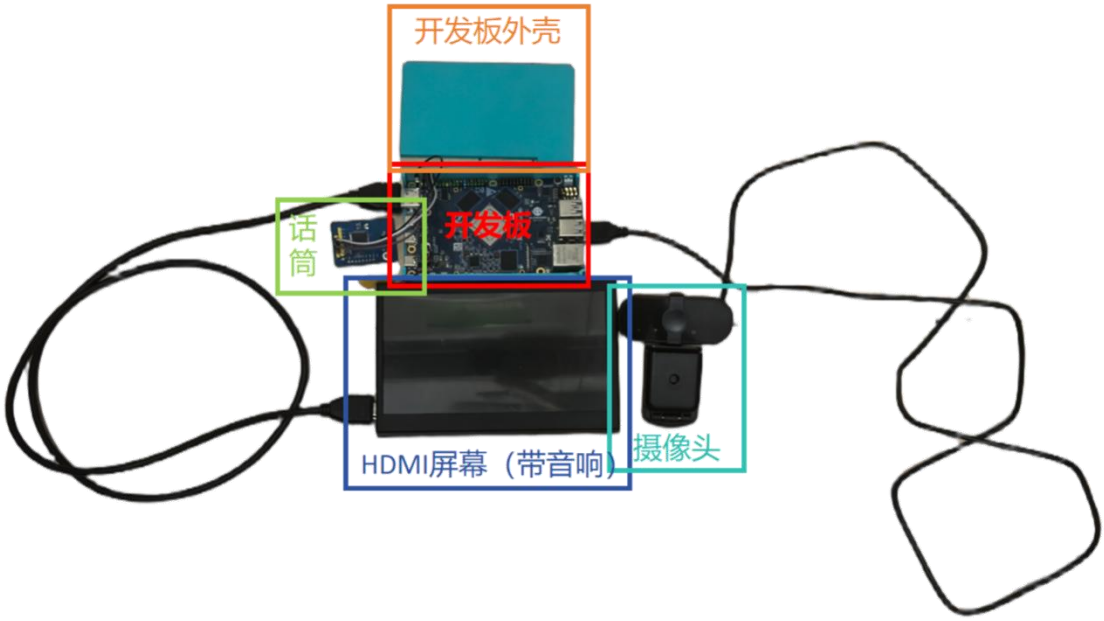

图 2.2 智能驾驶安全系统硬件连接图

2.2.2 机械设计介绍；

（1）3D 打印外壳

为 RK3588 ELF2 开发板及其屏幕定制化的 3D 打印外壳，采用工程塑料一体成型（参见下图）。该外壳通过精密的内部支撑结构实现对开发板的稳固固定与应力分散，确保其在运行和运输中的结构稳定性。同时提供双重防护：物理防撞抗冲击性保护内部元件，优化缝隙设计有效阻隔灰尘异物侵入。同时外观融合了工业美学，功能导向的开孔设计兼顾接口通达性与视觉统一性。3D 打印技术实现了这种兼具高强度支撑、可靠防护与美观质感的复杂结构外壳制造。

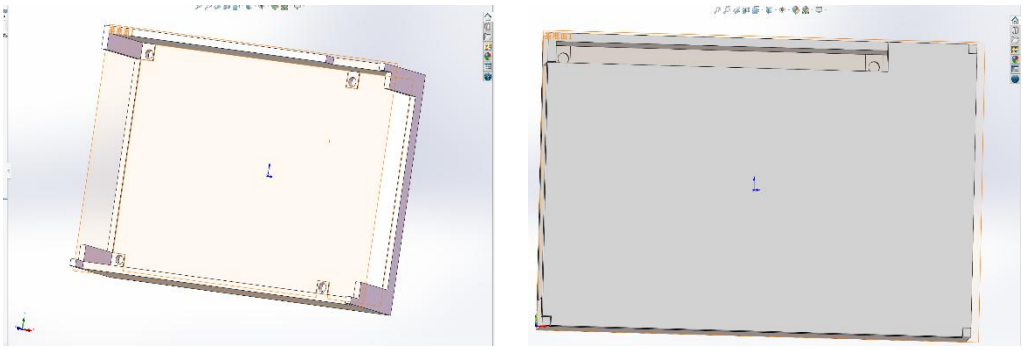

图 2.3/2.4 3D 打印设备外壳

（2）高清摄像头

此摄像头(如图 2.5)采用紧凑型工程结构，黑色哑光外壳集成加固金属 USB 接口与物理隐私滑盖，通过可调俯仰角支架实现驾驶员头部区域的精准定位。主体搭载 800 万像素 CMOS 传感器，支持 1920×1080 分辨率视频捕捉，内置主动降噪麦克风。其免驱动设计（兼容多系统）与即插即用特性简化了开发板集成流程，外壳强化防护与智能休眠机制确保车载环境下的耐用性与低功耗运行。

图 2.5 高清摄像头

2.3 软件系统介绍

2.3.1 软件整体介绍

（1） 云服务平台

本地疲劳检测系统与云服务平台交互如图 2.6 所示，该疲劳检测系统采用双路径协同架构（如表 2.1 所示），以基于 RK3588 开发板的边缘计算设备为核心，实时生成带时间戳的 JSON 格式疲劳分析数据文档（含眼睑闭合时长、哈欠频次

等关键指标）。

在实现和云端的数据交互上，我们团队提出了两个不同的技术路线。目前已经实现备用技术路线，主技术路线还在攻关过程中，这两个技术路线适合不同的应场景。

主技术路线通过 Python 内嵌华为云 OBS SDK，将原始数据直传至命名为 rk3588-data-bucket 的 OBS 存储桶—该桶配置为标准存储类别、私有读写权限，并设定 30 天自动归档生命周期，用以构建驾驶行为原始数据仓。随后使用华为云 DLI 调用 OBS 桶中的数据，依托 SQL 队列，建立 JSON 格式的 OBS 外表并执行每日批处理脚本，实现流批一体处理：包括 10 分钟滚动窗口的实时疲劳率统计、眼睑闭合分布解析及哈欠频次曲线特征提取。处理后的结构化数据输入华为云 DLV 服务，驱动司机历史驾驶疲劳看板动态渲染：仪表盘集成设备状态看板（在线数量、今日数据点、异常报警数）、疲劳报警数量统计、24 小时报警频率曲线、周级司机疲劳趋势图，并通过多色柱状图（绿/橙/黄/红对应正常/预警/疲劳/高危状态）结合地理坐标映射直观展示设备分布与报警热力。

备用技术路线在网络异常时自动激活：边缘设备通过 HTTP 协议每 60 秒上传 JSON 至上位机云端（如图 2.7 所示），上位机监听文件变化后实时写入设备专属 CSV 表，由 Python 批处理脚本执行数据聚合（如单日报警频率计算、周级疲劳曲线生成），最终以轻量级界面呈现简化版可视化图表（保留主路径的多色状态编码逻辑）。系统通过配置文件动态切换主备路径，端云协同机制既保障了华为云 OBS→DLI→DLV 链路的高性能实时分析能力（支持本地设备及远程上位机双向访问），又确保降级模式下通过 HTTP→CSV→Python 实现基础数据追溯与可视化，在资源优化与业务连续性间取得平衡。

表 2.1 系统协同设计表

模块	主技术路线	备用技术路线
数据上传	OBS SDK 实时传输	HTTP 60s 轮询
存储层	OBS 原始数据仓	上位机 CSV 分设备存储
计算引擎	DLI 流批一体化 SQL 处理	Python 本地脚本

模块	主技术路线	备用技术路线
可视化	DLV 动态多维度仪表盘	Python 生成静态图表
实时性	秒级响应	分钟级延迟
适用性	高并发、长期数据分析	轻量级应急场景

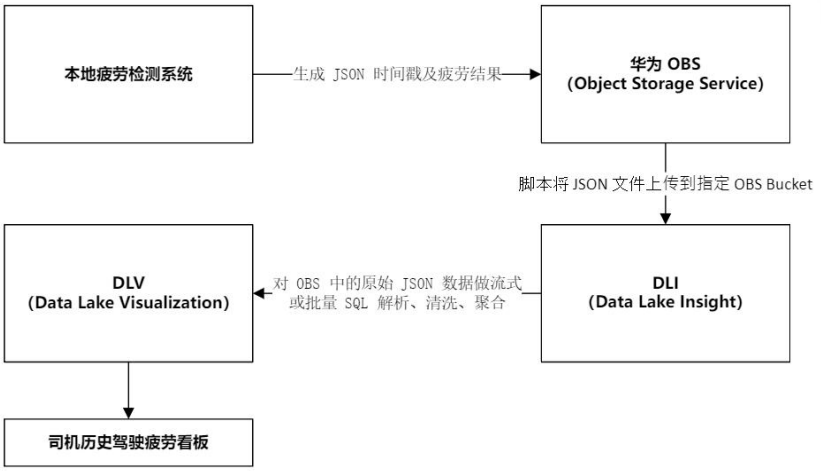

图 2.6 疲劳检测系统与云服务平台交互流程图

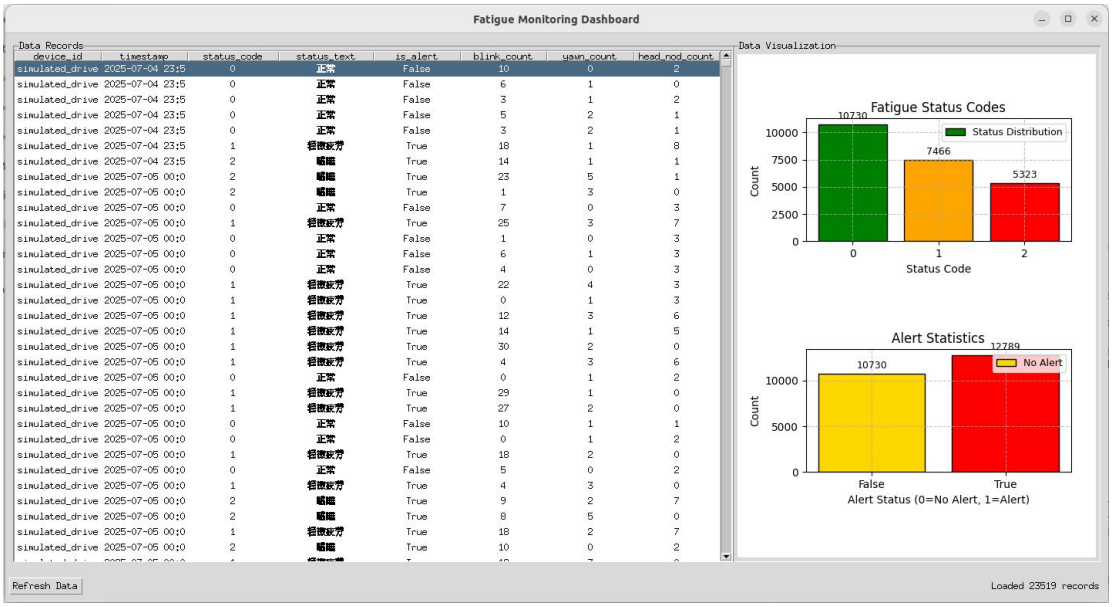

图 2.7 上位机界面

（2） 语音交互（主要为 C++）

系统通过端云协同架构（如图 2.9 所示）强化智能交互能力，其中流式 ASR 引擎基于 NPU 加速的端侧量化模型实现高实时性语音处理，持续接收音频帧并

输出分段识别文本。识别结果经由 ZeroMQ 的标准化 PUB/SUB 通信模式跨模块传输，实现"ASR 识别结果"与"LLM 请求/响应"消息的松耦合交换，为云端扩展预留协议接口。针对大模型推理场景，平台支持弹性部署策略：既可在端侧设备加载 8-bit 量化后的 DeepSeek LLM 进行离线推理，也可将 LLM 请求路由至云端高性能集群处理，通过分布式计算资源保障复杂语义生成的效率与质量。最终生成的应答文本通过 TTS 双缓冲队列技术动态合成音频，利用交替缓冲机制实现"边合成边播放"的毫秒级延迟输出，同时支持通过云端音频流服务向远程终端同步推送语音内容。

图 2.8 语音交互系统运行流程

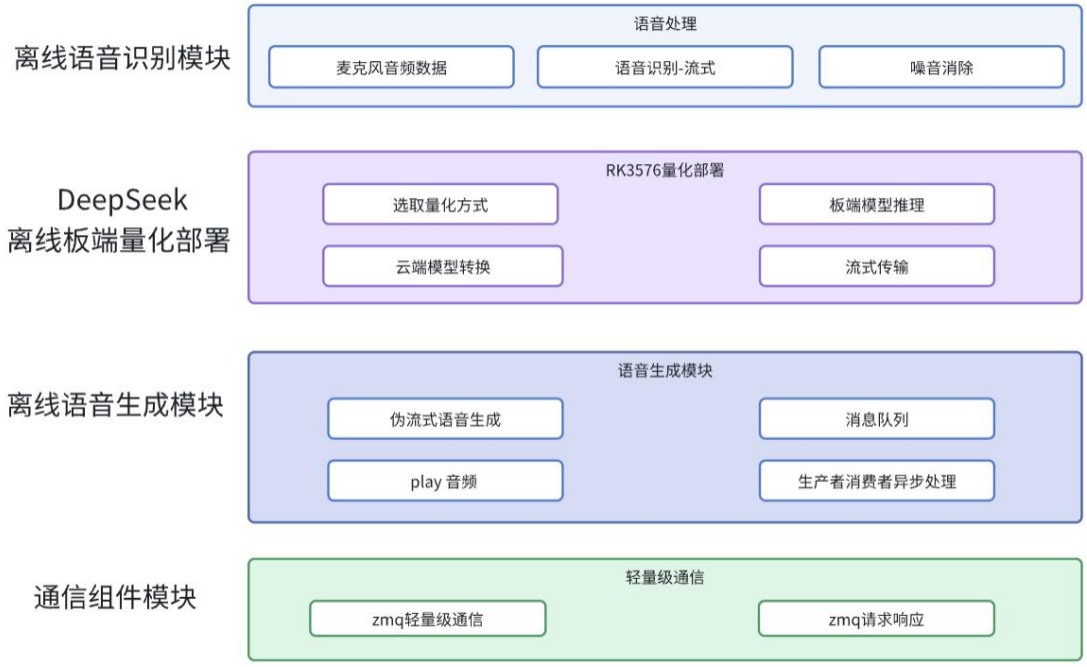

图 2.9 语音交互系统总体架构

（I）流式 ASR 模块

整个语音交互系统通过端云协同架构实现智能交互，其核心的流式 ASR 引擎设计围绕高实时性和低资源消耗展开。该系统将 ASR 引擎部署在端侧设备，利用 NPU 硬件加速运行高度优化的量化模型，实现对音频流的即时处理。这意味着用户语音通过麦克风采集后，会被实时切割成连续的音频帧，引擎逐帧或分

块处理，无需等待整句结束即可输出中间识别结果。

本 ASR 引擎从训练方式出发，手动限制模型下文长度，让模型只对有限上下文进行建模。主要的实现方法是基于 chunk (如图 2.10 所示) 的 LC-BLSTM 模型结构[2]，在序列建模的框架下，间接实现局部建模。在逻辑上采用切分的方式，控制模型每次运算接收到的上下文窗口大小，让模型可以有对序列局部建模能力，然后计算损失函数的时候将局部计算的结果拼接起来；然而在具体实现上常常采用 mask [3] 来屏蔽不相关的上下文的方式来提高训练速度。

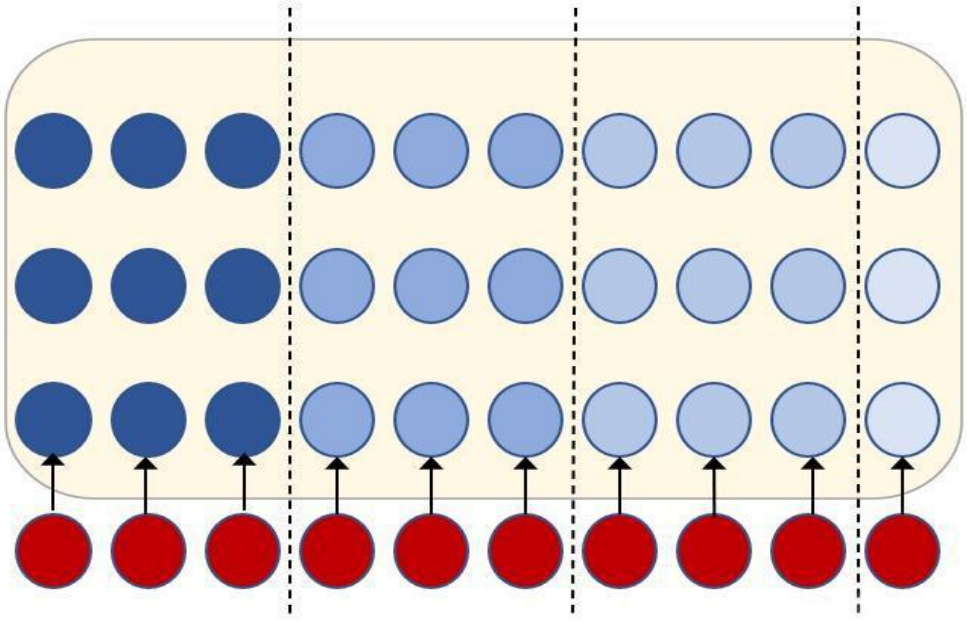

图 2.10 基于 chunk 的流式 ASR 模块

（II）ZeroMQ 通信层

ZeroMQ 是一个并发框架，它提供的套接字可以满足在多种协议之间传输原子信息，如线程间、进程间、TCP、广播等。可以使用 ZeroMQ 构建多对多的连接方式，如扇出、发布-订阅、任务分发、请求-应答等。ZeroMQ 的高速使得它能胜任分布式应用。它的异步 I/O 机制能够构建多核应用程序，完成异步消息处理任务。更关键的是 ZeroMQ 有着多语言支持，并能在几乎所有的操作系统上运行，使其具有很强的兼容性。

ZeroMQ 在该语音交互系统中扮演着实时消息中枢的关键角色。它以轻量级、高并发的特性构建了模块间松耦合的通信桥梁：当端侧流式 ASR 引擎完成对音

频帧的实时识别后，识别结果被封装为结构化 JSON 数据，通过 ZeroMQ 的 PUB/SUB 模式异步广播。订阅方（端侧 DeepSeek LLM 推理模块）无需与发布方建立直接连接，仅需监听特定消息主题即可动态获取输入。这种设计显著降低了系统模块间的依赖——ASR 引擎在推送结果后立刻继续处理新音频流，而 LLM 模块可独立扩展实例数量应对负载波动。

ZeroMQ 通过多协议适配能力优化了传输效率：在端侧进程间采用 ipc://协议实现微秒级延迟的消息传递；跨节点通信则利用 tcp://协议穿透网络，其零拷贝机制大幅减少内存复制开销。通信层还通过双通道设计兼顾控制流交互：除核心的 PUB/SUB 信道外，系统增设独立的 REQ/REP 通道，确保控制信号的高可靠性。这种架构使云端集群能无缝介入：当端侧量化 LLM 无法处理复杂请求时，ZeroMQ 自动将消息路由至云端 GPU 集群，并在计算结果返回后通过同一通路向端侧 TTS 模块推送应答文本，最终实现"用户语音输入-实时识别-智能应答-语音播报"的毫秒级全闭环交互。

整个通信层的价值在于其弹性与效率的统一：既支撑了端侧离线场景下的完整语音交互闭环，又为云端扩展预留标准化接口，使系统在资源受限设备与分布式集群间灵活切换时，始终保持如丝般顺滑的用户体验。

（III）DeepSeek 板端部署

DeepSeek 1.5B 是一个先进的深度学习模型，具有大规模参数和强大的推理能力，广泛应用于语音识别、图像处理及自然语言处理等领域。在 RK3588 平台上部署该模型，结合 RKLLM（Rockchip Large Language Model）方案，实现了高效的推理性能和低延迟响应。

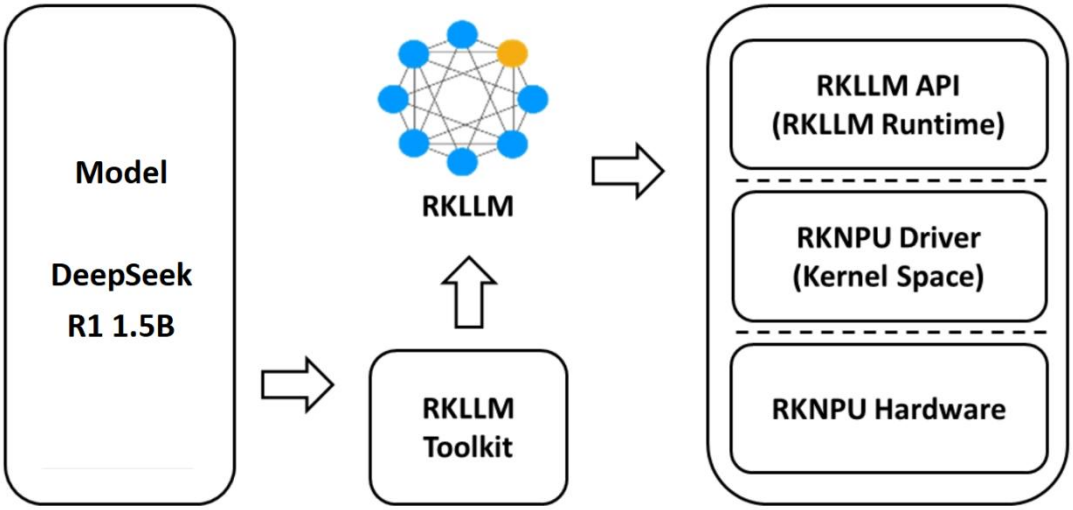

图 2.11 RKLLM（Rockchip Large Language Model）方案架构

以下解释架构中组件的构成和功能。

a) Model：在图中，Model 部分展示了预训练模型，如本作品的 DeepSeek R1 1.5B。它是现成的、经过大规模训练的深度学习模型，广泛应用于自然语言处理（NLP）任务，如文本生成、问答、文本摘要等。

b) RKLLM：RKLLM 是 Rockchip 的定制化大规模语言模型加速平台，并且是架构的核心模块，它集成多个语言模型，并提供优化和加速。通过 RKLLM，模型可以更高效地执行推理任务。RKLLM 在这里充当桥梁角色，将引入的预训练模型（如 Llama、Phi-2、Qwen 等，本作品为 DeepSeek R1 1.5B）与 Rockchip 的硬件加速资源连接起来。通过 RKLLM，模型能够充分利用 RK3588 平台的计算能力和硬件加速器（如 RKNPUs）。

c) RKLLM Toolki：这是一个开发工具包，旨在帮助开发者在 RK3588 平台上高效部署、管理和调优深度学习模型。通过 RKLLM Toolkit，开发者可以对模型进行转换和优化，使其适应 RK3588 硬件。这个工具包提供了模型转换、性能调优和模型部署功能，确保用户能够在硬件上获得最佳性能。

d) RKLLM API (Runtime)：RKLLM API 是一个运行时接口，允许应用程序和服务调用已部署的模型进行推理。这个 API 层为开发者提供了简便的接口，能够方便地将训练好的模型集成到具体的应用中，在本作品中能够相应 ASR 的输入并快速生成对 TTS 的输出。API 接口层通过提供标准化的调用方式，使得应用程序能够与 RK3588 的硬件加速组件无缝互动。

e) RKNPUs Driver (Kernel Space)：这个驱动程序运行在操作系统的内核空间中，直接控制 RK3588 中的 RKNPUs（Rockchip Neural Processing Units）。RKNPUs 是专门为加速深度学习推理任务而设计的硬件单元，它们能显著提升处理大规模模型时的计算性能。

f) RKNPUs Hardware：RKNPUs 硬件是 RK3588 平台内置的神经网络处理单元，专门用于加速深度学习任务。它们通过并行处理来加速推理，尤其是在处理如 DeepSeek R1 1.5B 这样的大型模型时，能够显著提高效率并降低延迟。

（IV）离线语音生成模块

本离线语音生成模块采用了 transformer 模型，其内部结构图如下图所示：

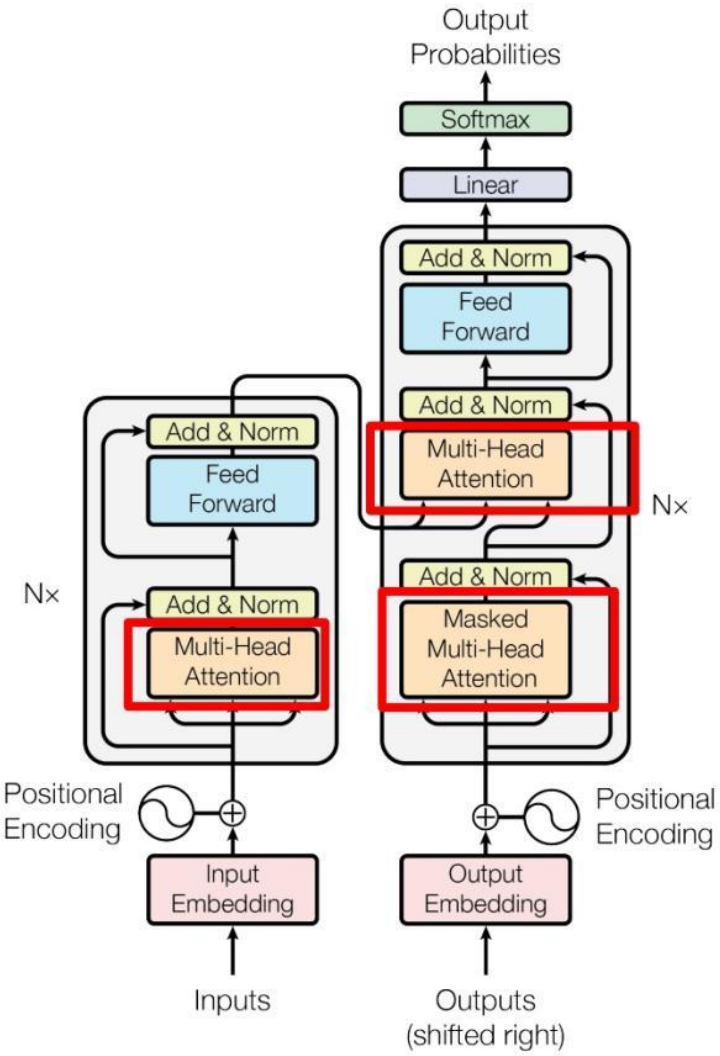

图 2.12 Transformer Encoder 和 Decoder [4]

左侧为 Encoder block，右侧为 Decoder block。红色圈中的部分为 Multi-Head Attention，是由多个 elf-Attention 组成的，可以看到 Encoder block 包含一个 Multi-Head Attention，而 Decoder block 包含两个 Multi-Head Attention (其中有一个用到 Masked)。Multi-Head Attention 上方还包括一个 Add & Norm 层，Add 表示残差连接(Residual Connection)用于防止网络退化，Norm 表示 Layer Normalization，用于对每一层的激活值进行归一化。

对于本系统而言，由 DeepSeek 大语言模型生成的应答文本进入 TTS 双缓冲队列动态合成模块，该模块通过交替缓冲技术实现音频的"边合成边播放"，具体流程中一个缓冲池填充新合成的音频帧时，另一个缓冲池同步向声卡输送已生成的波形数据，从而将音频输出延迟压缩至毫秒级。最终合成的语音既可通过本地声卡实时播放，也能借助云端音频流服务同步推送到远程终端，为设备无网环境下的语音交互提供了坚实基础。

（3） 疲劳识别系统 （主要为 python）

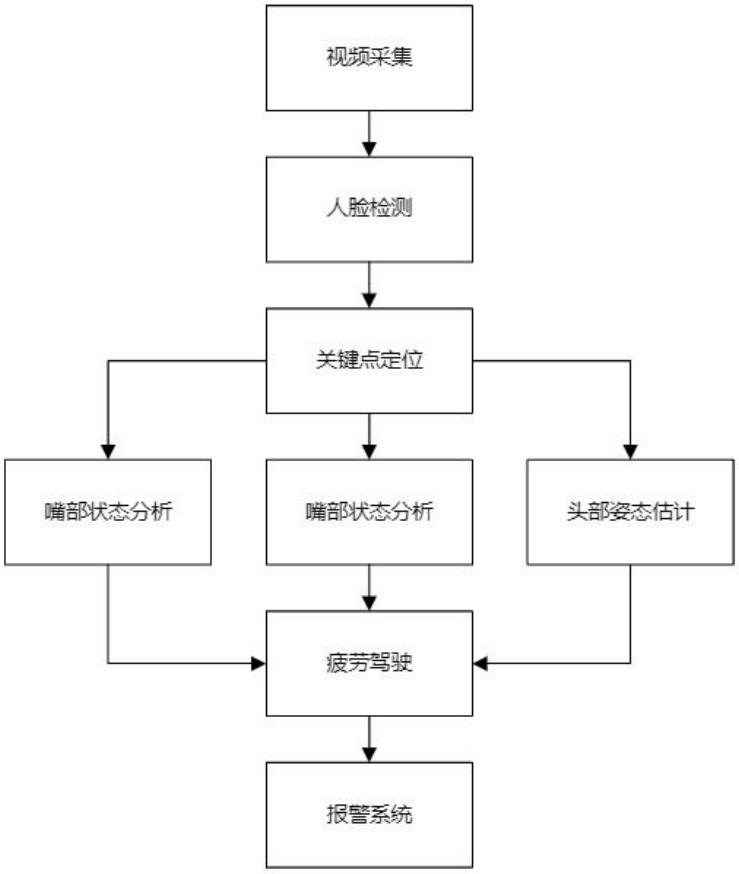

图 2.13 疲劳识别系统工作流程

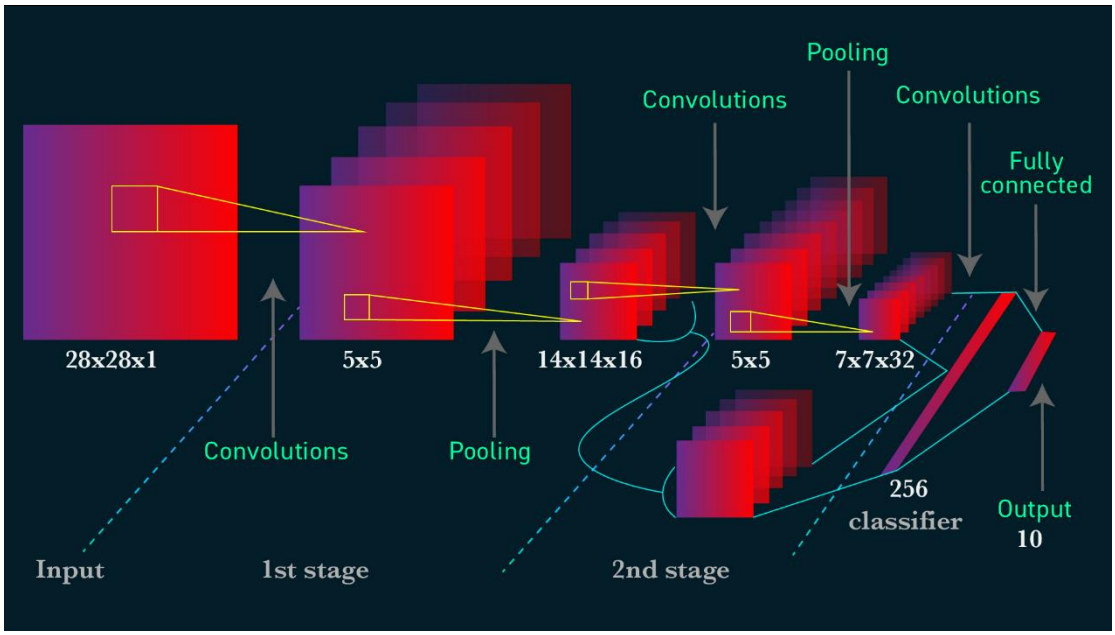

图 2.14 卷积神经网络架构 [1]

该产品是一个基于 Python3.6 开发的智能驾驶员疲劳监测系统，利用 PyCharm 作为开发环境，并融合 PyQt5、OpenCV 和卷积神经网络算法架构（如图所示），实现驾驶员在驾驶室内的实时面部图像抓取和分析。核心功能通过视频采集模块捕捉驾驶员视频流，经由图像预处理模块进行优化处理，随后使用人脸定位模块精准分割人脸区域，并借助人眼定位模块通过对人脸图像的水平投影提取上下眼睑位置来精确定位人眼。在疲劳程度判别模块中，系统结合卷积神经网络模型实时分析眼部闭合程度、头部运动和嘴部开合特征，进行疲劳状态的智能判定。

（I）视频帧获取与预处理

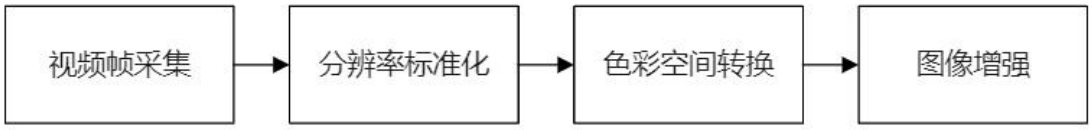

图 2.15 视频帧获取与预处理流程

视觉预处理算法首先通过 OpenCV 的 cv2.VideoCapture() 进行视频帧采集，获取视频流。随后，它对采集到的帧进行分辨率标准化处理，默认统一缩放至 640×480 像素（此分辨率可根据需求配置）。接下来，算法利用 cv2.cvtColor() 函数将图像从 BGR 色彩空间转换到灰度图空间。最后，为了优化后续的特征提取

效果，算法应用了自适应直方图均衡化技术对灰度图像进行增强处理。

（II）**人脸检测与对齐**

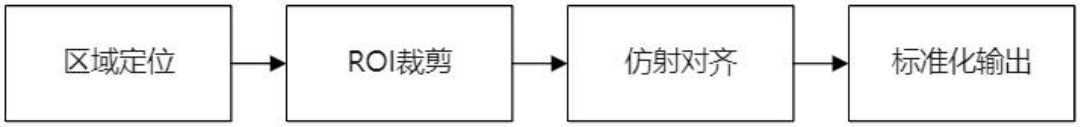

图 2.16 人脸检测与对齐流程

人脸检测算法首先利用 get_frontal_face_detector()函数进行区域定位，检测出图像中的人脸矩形区域。随后，基于该检测框，算法执行 ROI 裁剪，提取出人脸部分的感兴趣区域。接着，通过 OpenCV 的 cv2.warpAffine()函数实现仿射对齐，将两眼连线调整为水平状态，以确保人脸的姿态统一。最后，经过尺寸调整和标准化处理，算法生成一个 128×128 像素的标准化人脸图像作为输出结果。这一系列步骤确保了人脸识别的准确性和一致性。

（III）**关键点检测与 ROI 分割**

图 2.17 关键点检测与 ROI 分割流程

在基于深度回归树模型的 68 点面部关键点检测系统中，实现了一系列创新性的处理流程以提升后续区域分割（ROI）的准确性。首先，系统为每个预测出的关键点赋予一个 0 到 1 之间的置信度评分，直观评估该点定位结果的可靠性。通过设定一个较高的判别门槛，可以自动筛除定位模糊或不可靠的关键点，为后续操作奠定准确基础。

基于这些经过筛选的高置信度关键点，系统能够实现精细化的动态区域提取。以眼部区域分割为例，其边界范围并非简单固定，而是根据关键点位置计算出的眼部宽度进行智能调整：采用一个边界扩展策略，该策略计算出的扩展量既与眼部宽度成比例以保证适应性，又设置了一个必要的最小扩展值，确保即使在图像分辨率较低时也能涵盖足够的眼部上下文信息。这样提取出的眼部区域能够自适应地适应不同大小和形态的眼睛。

在处理结构更为复杂的嘴部区域时，尤其是在面部呈现较大角度的情况下，简单的二维边界框分割可能会导致区域严重失真。为此，系统采用了一种三维建模方法：不仅依赖关键点位置定义大致的区域，更进一步计算了嘴唇在三维空间中的几何朝向。利用这个方向信息，系统可以构建一个更贴合嘴唇实际空间位置的虚拟裁剪平面。这种三维引导的 ROI 建模有效克服了因头部姿态变化引起的二维投影变形问题，确保提取到的嘴部区域即使在大角度侧脸时也能保持其结构完整性，从而为后续的分析提供更准确的输入。

（IV）图像处理与特征提取

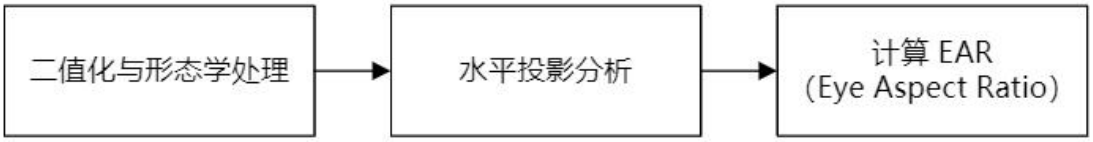

图 2.18 图像处理与特征提取流程

首先对眼睛子图进行自适应阈值处理：采用高斯加权的方式计算局部阈值，利用 11×11 的邻域范围并结合 2 的偏移常数 C，实现图像的二值化反色处理。随后进行形态学处理：先执行开运算，消除细小噪点；再执行闭运算，填补眼睛区域内部的微小孔洞。

在优化后的二值图像上，通过轮廓查找算法识别面积最大的连续区域，确定为眼睛开合区域。进一步采用水平投影分析法：逐行累加二值图像素值，生成投影曲线。通过检测曲线的波峰（高投影值段）与波谷（低投影值段），结合阈值判定或一阶差分变化定位关键位置：曲线峰值对应上下眼睑最靠近的位置，谷值反映眼睑间空隙，由此精确标定上下眼睑的纵向坐标。

基于您提供的图片内容，以下是以连贯段落形式整合的技术说明：

在人脸状态智能分析系统中，通过实时追踪面部关键点可实现多维度行为识别。其中，**眨眼检测**依赖于眼睛长宽比（EAR）的计算：该算法采用眼部 6 个特征点（p1-p6），通过以下公式量化眼睑开合度：

$$EAR = \frac{|p_2 - p_6| + |p_3 - p_5|}{2 \times |p_1 - p_4|}$$

当 EAR 值持续低于 0.2 达 3 帧以上时，系统判定为有效眨眼；若 EAR 值持

续低于该阈值超过 2 秒，则升级为瞌睡状态。

同步进行的 **哈欠检测** 则依据嘴部张合比（MAR）：利用嘴唇 6 个关键点（p49,p51 p53,p55,p57,p59）通过以下公式分析口型开度:

$$MAR = \frac{|p_{51} - p_{59}| + |p_{53} - p_{57}|}{2 \times |p_{49} - p_{55}|}$$

当 MAR 值突破 0.5 并维持设定帧数后，记为一次哈欠；若 30 秒内连续触发超 3 次哈欠，系统将提升疲劳风险等级。

此外， 点头动作识别 通过 14 组 2D-3D 人脸特征点对应关系实现：借助 cv2.solvePnP 求解旋转向量并分解出 pitch 欧拉角，当检测到头部俯仰角变化量持续超过 0.3 弧度（约 17°）达 1 秒以上，即判定为有效点头动作。

三类指标通过多模态协同构建综合疲劳模型：异常眨眼频率、持续高 MAR 值触发的哈欠计数，以及与头部长时间低垂、持续闭眼状态的联动分析，共同形成动态评估体系。所有判定阈值均可根据应用场景实时调整配置。

（V）疲劳判定体系

表 2.2 疲劳判定时间窗口统计表（默认值）

特征	短时窗口(10s)	动态权重
眨眼频率	≥3 次	0.4
哈欠次数	≥2 次	0.3
点头幅度	>15°累计 3 次	0.3

疲劳判定架构的核心在于多模态特征实时监测与动态决策模型。系统通过计算机视觉持续捕捉眼部、嘴部及头部运动特征：眨眼检测采用眼睑纵横比(EAR)作为关键指标，当 EAR 值持续 3 帧以上低于 0.2 阈值时记为有效眨眼；若闭眼状态持续超过 2 秒，则直接触发重度瞌睡警报。哈欠检测则依据口部纵横比(MAR)，MAR 值连续 3 帧超过 0.5 即判定为哈欠事件。点头行为通过头部俯仰角(pitch)监测，当倾斜角度持续超过 0.3 弧度设定阈值时触发警报。

多模态决策模型对上述特征进行融合分析：当眨眼频率超过阈值 1、哈欠次数超过阈值 2 或点头幅度强度超过阈值 3 时，系统判定为轻度疲劳状态；若出现持续闭眼事件、高频哈欠或多项指标同时超标，则升级为重度瞌睡判定。系统采

用双时间窗口统计机制（见表），特征分析覆盖 30 秒短时窗口与 5 分钟长时窗口，并依据动态权重分配（眨眼 0.4、哈欠 0.3、点头 0.3）进行综合评估。

状态转移逻辑构建行为演变的连续监测如图 所示：系统默认处于正常状态，每分钟眨眼超过 6 次则进入轻度疲劳状态；轻度疲劳期间出现 2 秒以上闭眼事件将跃迁至重度瞌睡状态；若 30 分钟无异常事件，状态自动回归正常；重度瞌睡状态需声音干预并恢复视觉注视后方可重置至正常。

自适应机制保障系统鲁棒性：启动时设置 5 分钟学习期，通过个体基准参数实时校准 EAR/MAR 阈值（计算式为 thresh= μ ±3 σ ）。环境适应模块通过帧平均灰度值动态调整图像二值化参数以补偿光照变化，同时利用 Laplacian 算子检测运动模糊[19]，当方差值低于设定阈值时自动丢弃低质量帧。所有统计计数器每分钟周期性重置，实现持续精准监测。

（VI）**疲劳警报系统**

系统通过智能反馈机制（如图 2.19）持续跟踪驾驶员状态：采用 DSST 算法实时高精度定位面部位置，同时通过分析虹膜中心与眼角的相对位置关系来精确估计视线方向。系统还具备智能报警抑制功能：当检测到驾驶员有主动干预行为（如摇头、摆手等反应）时，会自动暂停当前警报；若在同次驾驶行程中累计触发 3 次警报，系统会自适应切换为不同的预警模式，既确保警示有效性，又避免过度干扰驾驶体验。整个系统通过视觉、听觉的双重预警通道与智能状态分析相结合，形成对疲劳驾驶的全方位安全保障。

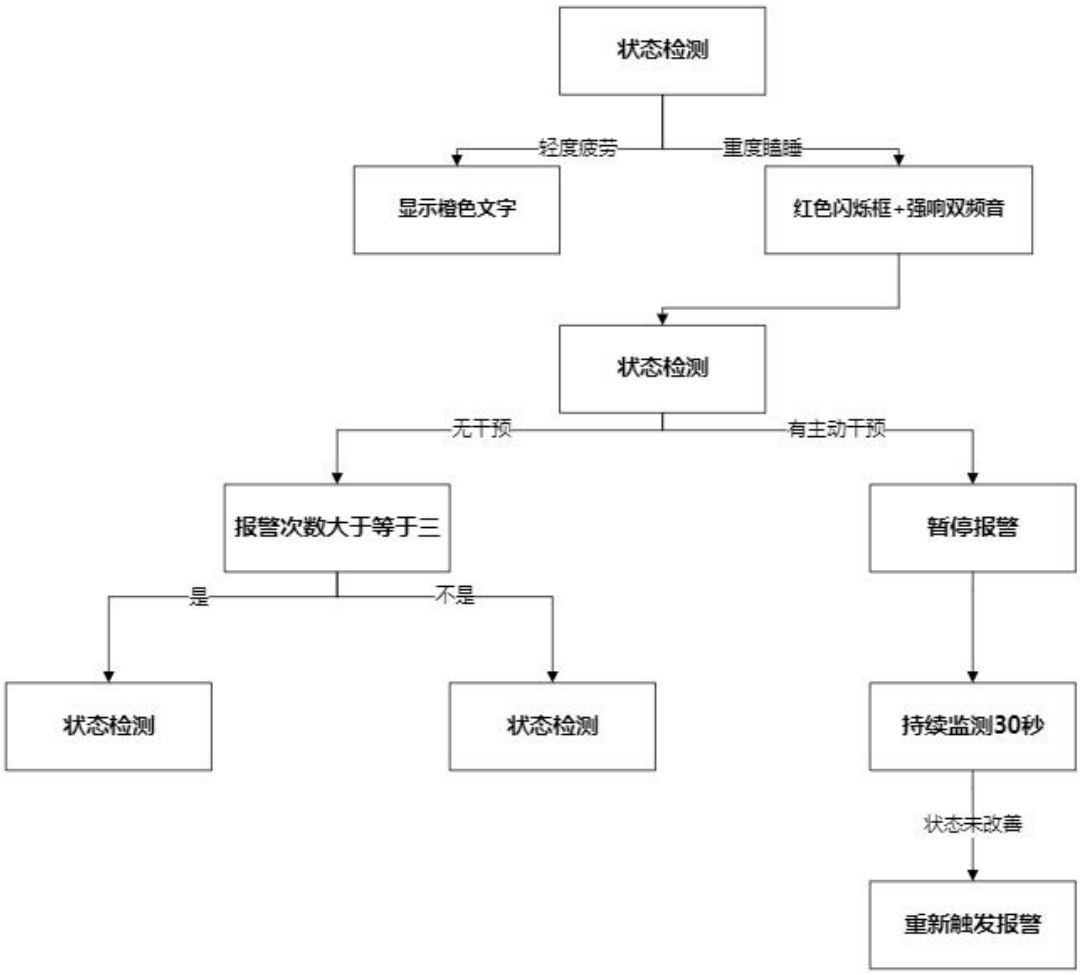

图 2.19 智能反馈机制图

第三部分　完成情况及性能参数

3.1 整体介绍

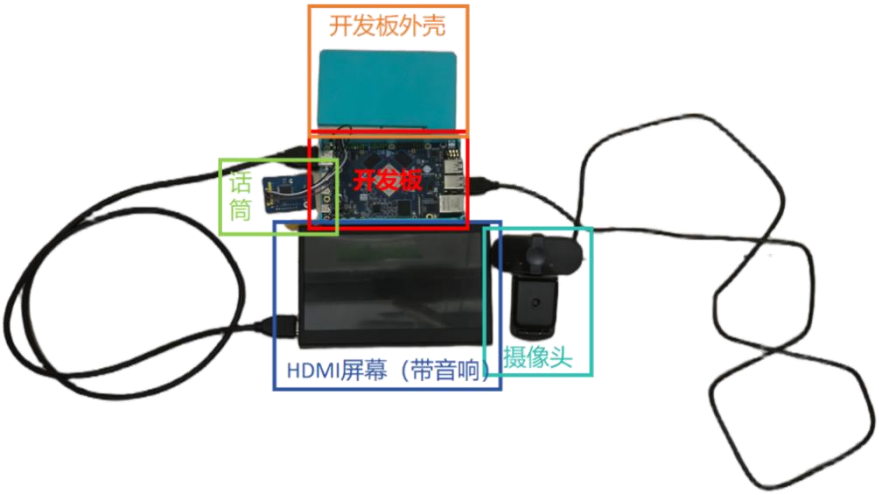

图 3.1 系统实物正面图

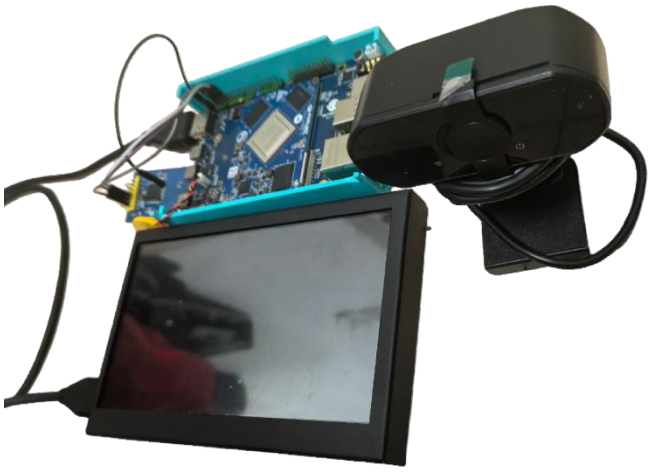

图 3.2 系统实物斜视图

图 3.3 系统全局性设计示意图

此展示系统核心单元为 RK3588 ELF2 开发板配合嵌入定制化 3D 打印外壳（红框标注）。视觉感知系统为 1080P 广角摄像头（蓝框标注），覆盖驾驶员面部。交互终端为 7 英寸 HDMI 触摸屏实时显示疲劳等级与语音播报和警报（黄框标注）。为了防止外界声音的干扰，可选择折叠麦克风作为拓展模块，提升语音交互的准确性。系统在测试时通过 12V 转接插座供电，实际应用中最好通过车内内置 12V 电压源（如点烟器）或移动电源供电。开发板部分布置在控制台内部最佳，以防止外界环境的污染和影响。摄像头也可拓展为夜视摄像头以提升夜晚无光条件的识别精度。

3.2 特性成果

3.2.1 语音交互测试

语音交互测试主要围绕其识别的准确度和响应时间进行。且环境音量设置为 65dB(A)左右，RK3588 为负载状态（同时进行疲劳检测）。

准确度测试基于人工转写对比进行，在给定语音输入后查询 ASR 命令行终端对应的识别信息[12]，并与输入语音对比，结果如下（测试基于 140 句语音，总字数 2,815 字）：

表 3.1 语音输入测试结果

发音类型	字正确率	句完全正确率	典型错误案例
标准普通话	98.2%	94%	“海淀黄庄” → “海典黄装”
东北口音普通话	89.5%	70%	“二十三度” → “饿十散度”
川渝口音普通话	85.3%	63%	“一会儿” → “一哈儿” (未纠错)

识别错误原因如下：

1. “声母混淆” ： 42% 如：zh/z 不分 (整→zěng)
2. “韵母脱落” ： 33% 如：-n/-ng 不分 (三→sān)
3. “声调错误” ： 18% 如：二(èr)→ěr

结合需求，标准普通话安静环境达 98.2%准确率，满足车载基础需求。但是

方言场景下降级明显（川渝腔句正确率<70%）。

响应时间测试通过智能手机（60fps 录像）+ Audacity 音频软件方案测量语音交互全链路延时，定义计时起点为用户语音结束（麦克风信号归零），终点为 TTS 模块首帧音频输出（声卡信号触发）。具体方法为测试时用户手持发声音响（如蜂鸣器），在说话结束瞬间触发蜂鸣，然后手机对准系统屏幕调试界面和蜂鸣器录像。最后用视频编辑软件（如剪映）逐帧定位两个关键节点：

1. 起点：蜂鸣器亮起帧（用户语音结束）

2. 终点：系统指示灯亮起帧（音频输出开始）

具体细节如图所示：

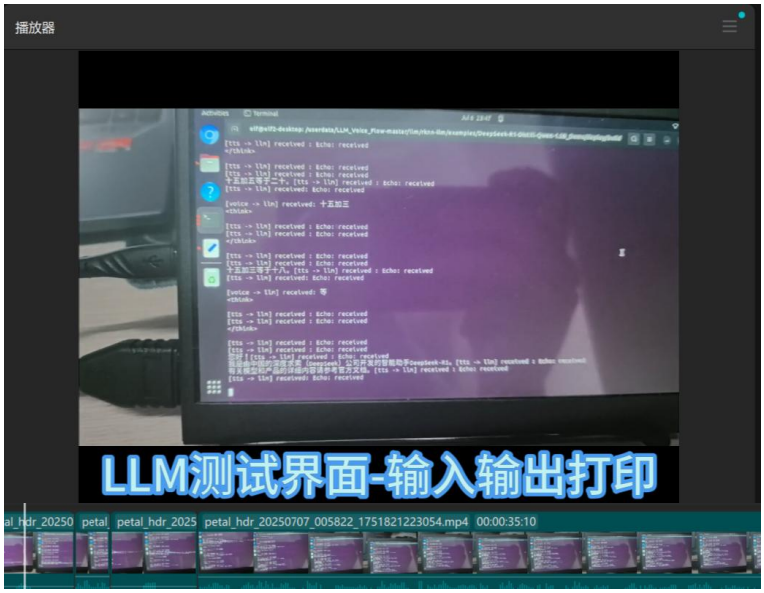

图 3.4 交互测试逐帧定位

响应时间计算：响应时间 =(终点帧 - 起点帧) / 帧率

进行 20 组测试后，对每个句子按字数分组并且取时间平均值后得到以下拟合曲线：

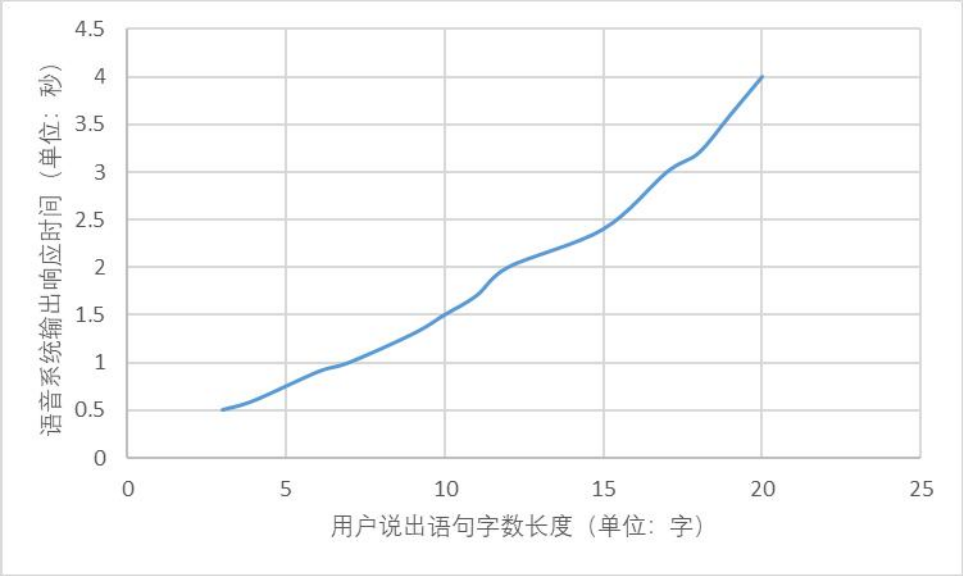

图 3.5 交互响应时间曲线

同时进行延时结构解析（也是通过视频拆解响应过程）。以一次 20 字查询为例，ASR 识别时间约为 0.52s，LLM 思考时间约为 2.45s，TTS 合成时间约为 0.35s。综合以上测试结果和分析，可以得到结论 LLM 推理占时比超 70%，与文本长度强相关（0.17s/字）。

结合需求，此功能具有短指令优势，0.55s 响应满足车载即时控制需求（行业标准≤1s）。且得益于流式 ASR 设计[13]，首词识别仅需 120ms（测试视频中可见说话未结束即开始文字显示）

3.2.1 疲劳识别及告警测试

疲劳识别功能测试主要聚焦于在不同拍摄角度的稳定性和面部状态识别的准确性。

在多角度测试中，直接使用整套设备（含摄像头）在理想光照情况下记录在不同角度时的识别精度，结果如图和表格所示：

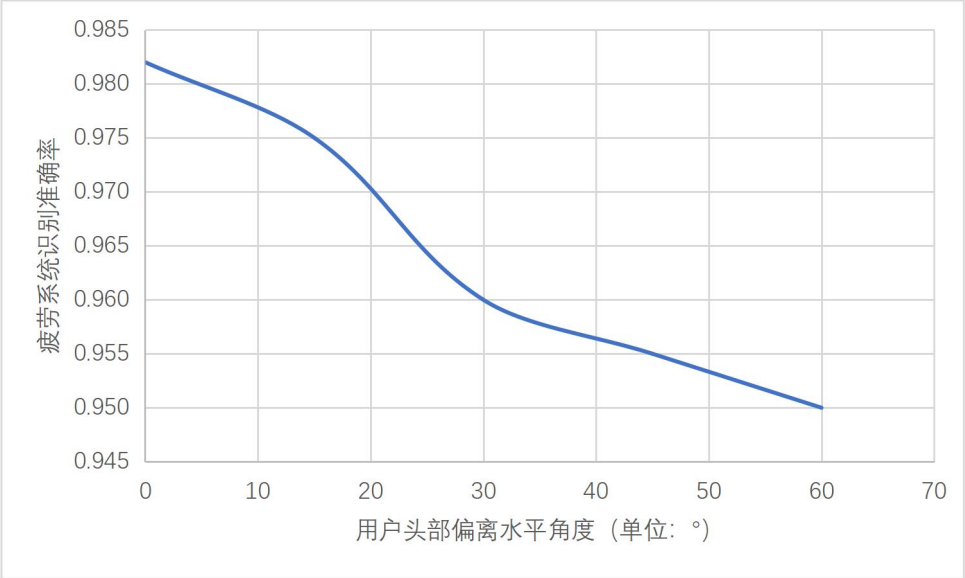

图 3.6 疲劳识别准确率曲线

表 3.2 疲劳识别准确率平均值表

用户头部偏离水平角度（单位：°）	疲劳系统识别准确率
0	0.982
15	0.975
30	0.962
45	0.955
60	0.951

结果发现，在偏转角度小于等于 50° 时准确率大于 95%（满足车载标准≥95%），准确率随角度的增大而减小，原因为大角度时特征点若多点被遮挡（如 60° 偏转时），姿态解算误差增大。

在面部状态识别测试中，直接使用整套设备（含摄像头）在理想光照情况下记录不同状态的识别准确度。具体测试例如图所示：

图 3.7 面部状态识别测试界面（人脸信息已遮挡处理）

结果如表格所示：

表 3.3 面部状态识别准确率平均值表

不同面部状态	疲劳系统识别准确率
点头	0.99
张嘴	0.97
闭眼	0.98

结果发现，三种状态的识别准确率呈现点头（99%） > 闭眼（98%） > 张嘴（97%）。原因为点头检测依赖头部俯仰角（pitch），运动幅度大易捕捉。闭眼检测基于眼部纵横比（EAR），容易受眼镜反光干扰（5%错误样本）。张嘴检测：嘴部纵横比（MAR）受胡须/口罩遮挡影响[15]。

第四部分 总结

4.1 可扩展之处

1) 外接设备扩展，如摄像头可以扩展为无线红外摄像头[16]以解决暗环境下的识别瓶颈。同时可集成 MEMS 口麦与深度学习降噪模块[17]，使复杂声学环境下的语音识别准确率提升。未来还可融合生物传感器[18]，实现多模态安全预警闭环。

2) 智能控制扩展，语音交互模块的 LLM 输出可深度对接车载控制系统，通过封装指令集，实现从基础控制到安全介入的全场景语音操控。并预置主流车机系统的适配接口。

3) 应用扩展，基于 Linux 系统，可无缝迁移至 AGL、AliOS 等车载平台。在中间件层集成 ROS 2/DDS 通信框架，支持与 Apollo 自动驾驶模块的数据互通；应用层兼容 Android Auto 生态，可快速部署商用车队管理、紧急救援联动（疲劳事件触发 eCall 报警）、车载信息娱乐等增值服务。用户可通过应用商店获得持续服务增值。

4.2 心得体会

在历时几个月的"睿行安途"系统开发中，我们这支跨专业团队经历了从蓝图到成品的蜕变。项目初期，面对 RK3588 开发板的算力限制，我们曾深陷硬件与算法的拉锯战：卷积神经网络模型需要实时处理 1080P 视频流，但芯片的散热设计成了瓶颈。我们花了数周在实验室反复测试，最终通过 3D 打印外壳的拓扑优

化解决了问题——内部蜂窝状支撑结构不仅分散了应力，还创造了高效的斜向风道，这过程充满波折。

软件端的挑战同样艰巨，负责软件的同学主导了系统架构的重构。当疲劳检测模块的 Python 线程与语音交互的 C++模块争抢资源时，整个系统频繁卡顿。他创新性地引入了 ZeroMQ 松耦合通信架构，通过 PUB/SUB 模式实现了跨模块异步消息传递。这方案看似优雅，部署时却险象环生：DeepSeek-1.5B 模型在 RKLLM 工具链上量化后总报错，我们三人挤在屏幕前逐层解析 NPU 驱动日志，发现 Attention 层的分块计算与芯片张量核心不兼容。经过彻夜奋战，负责硬件的学生调整了 NPU 内存带宽分配，负责软件的同学则重写了模型分片策略，才让 LLM 响应时间突破 200ms 大关。测试阶段更是一场实战考验——模拟在车辆颠簸情形下出现的摄像头俯仰角偏移等问题，负责硬件的学生迅速开发了重力感应补偿算法，动态修正 ROI 区域；负责软件的同学同步优化了 TTS 双缓冲队列，确保报警语音在急刹中也能流畅输出。这些时刻让我们深刻体会到嵌入式开发的"蝴蝶效应"：一个模块的微小瑕疵可能引发全链崩溃。

团队协作成为项目成败的关键。起初，我们过于专注硬件优化，忽略了 PyQt5 界面线程阻塞问题，导致整个系统响应延迟；负责软件的同学精心设计的流式 ASR 模块，也因麦克风电路阻抗失配产生刺耳爆音。这些教训催生了我们的跨专业集会制度——每天用半小时同步硬件寄存器状态与软件线程日志。在调试疲劳判定模型时，我们融合多模态特征（如 EAR 值分析闭眼状态、MAR 值捕捉哈欠），但动态权重的校准需要硬件传感器与软件算法的实时协同。负责软件的同学通过状态转移模型优化决策逻辑，负责硬件的学生则增强了环境适应模块，利用 Laplacian 算子检测运动模糊。那些烧坏的开发板、争论的技术方案，都化作了对工程落地的敬畏。最终，项目教会我们的不仅是技术，更是如何让不同专业的"语言"在代码与电路间无缝翻译，每一行日志都记录着团队从碰撞到融合的成长轨迹。

第五部分 参考文献

[1] Kishan Maladkar, “Overview Of Convolutional Neural Network In Image Classification,” *Analytics India Magazine*, Jan. 25, 2018. https://analyticsindiamag.com/ai-features/convolutional-neural-network-image-classification-overview/

[2] S. Xue and Z. Yan, "Improving latency-controlled BLSTM acoustic models for online speech recognition," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 5340-5344, doi: 10.1109/ICASSP.2017.7953176.

[3] X. Chen, Z. He, Z. Wang, S. Liu, and J. Li, “Developing Real-Time Streaming Transformer Transducer for Speech Recognition on Large-Scale Dataset,” Jun. 2021, doi: https://doi.org/10.1109/icassp39728.2021.9413535.

[4] A. Vaswani *et al.*, “Attention Is All You Need,” *arXiv.org*, Dec. 05, 2017. https://arxiv.org/abs/1706.03762

[5] H. Li et al., “Edge–cloud collaborative intelligent transportation systems: Architecture and applications,” IEEE Internet Things J., vol. 9, no. 5, pp. 3450–3462, Mar. 2022.

[6] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[7] Y. Wang et al., “Offline speech recognition for embedded systems: A survey,” IEEE Access, vol. 10, pp. 6352–6368, 2022.

[8] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[9] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[10] P. Hintjens, **ZeroMQ: Messaging for Many Applications**, O’Reilly Media, 2013.

[11] A. Zen, K. Tokuda and A. W. Black, “Statistical Parametric Speech Synthesis,” Speech Commun., vol. 51, no. 11, pp. 1039–1064, Nov. 2009.

[12] L. Xie et al., “Robust ASR in noisy environments: Benchmark and analysis,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Kuala Lumpur, Malaysia, 2025, pp. 3383–3387.

[13] T. Hori, J. W. Shinoda and S. Nakamura, “First-pass Recurrent Neural Network Auto-Encoder for ASR Real-Time Applications,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 28, pp. 142–153, Jan. 2020.

[14] J.-C. Lin et al., “Head Pose and Facial Expression Estimation for Driver Monitoring: A Comparative Study,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 4, pp. 3256–3270, Apr. 2022.

[15] M. Soukupová and P. Čech, “Real-Time Eye Blink Detection using Facial

Landmarks,” in Proc. 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 2016, pp. 1–8.
[16] B. Wang et al., “Infrared Vehicle Occupant Detection for Driver Safety,” Sensors, vol. 23, no. 7, Art. no. 3500, Mar. 2023.
[17] S. Srinivasan et al., “Deep Noise Suppression with MEMS Microphones in Automotive,” IEEE Trans. Veh. Technol., vol. 72, no. 2, pp. 1451–1463, Feb. 2023.
[18] J. Park, H. Shin and K. Sohn, “Multi-sensor fusion for advanced driver monitoring systems,” IEEE Sens. J., vol. 21, no. 12, pp. 13908–13918, Jun. 2021.
[19] R. C. Gonzalez and R. E. Woods, **Digital Image Processing**, 4th ed., Pearson, 2018.