

# 基于 RK3588 的多模态货运车辆安全监测终端

安途小分队

## 摘要

随着国家“交通强国”战略的深入实施<sup>[1][2][3]</sup>，交通运输行业正加快向智能化方向发展<sup>[4][5][6]</sup>。在如今的长途货运行业中，货运车辆运行时间长、司机疲劳驾驶和燃油被盗等安全问题频发<sup>[7][8]</sup>。针对这一现状，本文基于国产 RK3588 高性能边缘计算平台，设计并实现了一套运行于边缘端，融合视觉检测、生理参数监测与语音交互的多模态货运车辆安全监测系统<sup>[9][10]</sup>。本文所完成的主要工作如下：

硬件方面，板端采用搭载 RK3588 的飞凌 ELF2 开发板，连接两路摄像头（MIPI/USB），分别布置于驾驶室和燃油箱侧，用于检测司机疲劳状态和分心行为，以及油箱附近人员可疑行为的监控。配备了自主设计的生理参数检测模块，能够检测司机的心率，血氧等生理参数，并通过 4G 模块发往服务器端。连接了一个 I2S 的扬声器，在监测到上述行为时进行语音动态预警。

软件方面，使用 PYQT6 设计了一个运行于板端的上位机软件，用于司机实时掌握自身以及油箱的状态。通过自建分心驾驶行为数据集，采用 YOLOv8 识别玩手机、抽烟、喝水等分心动作；基于 MediaPipe 提取人脸关键点并结合眼睑闭合度等指标，实现对司机瞌睡、闭眼等疲劳状态的精准监测；针对燃油箱场景，使用 MediaPipe 提取靠近人员的人体关键点，构建可疑行为状态机，实现“弯腰低头”、“重复挥打”等可疑行为的精准识别。

为了实现板端回传数据的可视化，在服务器端设计了上位机软件。当板端完成多模态数据的融合分析后，通过 4G 网络将图像帧、生理参数及行为判定上传至服务器端。服务器端呈现上传的图像、安全监测次数动态变化，绘制生理参数曲线；并支持阈值配置、日志导出与报告生成功能，极大提升了运维效率与决策能力。

最后经系统测试，疲劳检测准确率达 98.5%，燃油防盗监测准确率达 95.3%；

端云协同处理延迟控制在 200ms 以内。相比传统方案，本系统具有三大优势：1）多模态数据融合分析能力；2）边缘-云端协同计算架构；3）高精度行为识别算法。实际应用表明，本系统有效解决了货运安全监测实时性差、手段单一等行业痛点，为智能货运发展提供了可靠的技术支撑。

## 第一部分 作品概述

### 1.1 功能与特性

（1）多模态安全监测：实时监测驾驶员疲劳与分心行为，支持“玩手机”，“抽烟”，“喝水”等多种分心操作和闭眼、瞌睡等疲劳状态的动态识别<sup>[11]</sup>；实时监测驾驶员生理参数，结合驾驶员疲劳状态，综合分析驾驶员驾驶状态。板端内置语音播报引擎，可根据检测结果发出语音报警提示，满足了多维度的安全监测需求。

（2）高性能边缘计算：基于 RK3588 高性能边缘计算平台，利用其 NPU 与 CPU 协同加速的优势<sup>[12]</sup>，板端部署 YOLOv8 轻量级目标检测算法与 MediaPipe 关键点识别模型，实现全时段、低延时的多模态数据融合与分析，无需依赖云端算力，满足了高实时性的安全预警需求。

（3）端-云协同安全预警：边缘端通过 4G 网络接入远程监测中心。边缘端界面侧重让司机掌握自己的驾驶状态及货运车辆的安全状态，及时收到预警安全状态，具有直观性、及时性的特点。服务器端上位机侧重数据集中展示，能够汇集来自多辆货运车辆的回传数据，提供驾驶员生理参数曲线、图像回放、日志导出，方便对多司机进行管理。

### 1.2 应用领域

（1）长途货运。长途货运车辆往往横跨多个省份，行驶里程长、驾驶环境复杂，司机疲劳与分心行为极易引发交通事故。同时，车辆在途停靠时燃油被盗风险较高，对运输企业造成直接经济损失。部署本系统后，车载终端在驾驶室与燃油箱侧分别安装摄像头，并配合心率、血氧等生理检测模块，实现驾驶员状态与油箱异常行为的全方位监控。

(2) 城市公共交通。在公交、地铁安检区或企业班车场景中，乘客人流密集、突发事件风险高，既要保障司机的状态安全，也需对车内或站台的异常行为进行预警。系统可在车厢内部与安检通道部署多摄像头，结合生理参数监测，形成乘客行为与司机状态的多模态联动安全网。

(3) 建筑与矿山工程车辆。建筑工地与矿山作业环境常伴随粉尘大、震动强、高空落物等危险，重型卡车司机操作不当或疲劳都可能导致高成本事故与设备损坏。本系统可以在车辆驾驶舱和卸料装置处部署工业级摄像头与生理传感器，减少长时间作业引发的安全隐患。

### 1.3 主要技术特点

(1) 自主设计独立的生理参数检测模块。由于传感器需要与司机保持长时间的接触，因此设计了独立的生理参数检测模块。由 STM32 为主控，使用 MAX30102 传感器采集生理参数，并通过 ESP8266 通过 TCP 与终端设备通信。同时模块还配备了 3.7 V 可充电锂电池，保证模块不间断采集司机生理参数。

(2) 实时多模态数据融合。在本系统中，终端采集的数据包括 YOLOv8 识别的驾驶员分心行为图像、MediaPipe 提取的人脸关键点、生理参数检测模块提供的心率与血氧等生理参数，以及 MediaPipe 识别到的油箱区域人员的可疑动作。系统在边缘端完成同步处理后，实时触发本地语音播报，并通过 4G 网络将图像帧、行为标签和生理参数上传至服务器。

(3) 目标检测模型的嵌入式部署。通过自建数据集，使用 YOLOv8 进行分心行为模型训练。在模型训练结束后先将其转为 ONNX 格式，再使用 RKNN-Toolkit2 将其转化为 RKNN 格式，并在转换过程中启用量化，有效压缩模型体积、降低计算复杂度。最终模型部署于 RK3588 的 NPU 上运行，充分利用其高达 6 TOPS 的神经网络加速能力，实现目标检测模型在边缘端的低延时、高帧率推理，满足全天候实时监测需求。

### 1.4 主要性能指标

为全面体现本系统在多模态安全预警、边缘计算等方面的能力，本文从模型指标、生理参数检测模块指标、系统稳定性等角度提炼出关键性能指标，包括疲

劳识别准确率、分心识别准确率、可疑行为识别准确率、模型大小、内存占用、传输延迟、上报率、续航、误报率、端-云通信延迟，如下表 1-1 所示：

表 1-1 系统主要性能指标

模型指标					生理检测模块指标			系统稳定性	
疲劳识别准确率	分心识别准确率	可疑行为识别准确率	模型大小	内存占用	传输延迟	上报率	续航	准确率	端-云通信延迟
98.5%	96.2%	95.3%	Yolov8: 8298KB MediaPipe: 无	Yolov8: 43% MediaPipe: 21%	25ms	100%	6h	98%	200ms

## 1.5 主要创新点

### （1）基于 RK3588 的高性能边缘部署架构

利用 RK3588 平台的 NPU 与 CPU 协同加速能力，将轻量级模型部署于边缘端，结合模型量化实现毫秒级低延时推理，摆脱云端算力依赖，满足全天候实时预警需求。

### （2）多模态融合的安全监测机制

充分整合视觉感知与生理信号，实现对驾驶员状态与车辆周边安全的高精度识别与实时预警。所有数据在边缘端完成同步融合与实时预警，并通过自然语音播报。

### （3）驾驶安全与资产防盗的双重预警

集成驾驶员状态检测与燃油防盗功能，提供了一个多维度、综合性的安全预警解决方案。

### （4）端 - 云协同的高效安全预警架构

基于 4G 网络的端 - 云协同架构，边缘端实现本地推理与实时预警，云端负责多车数据集中管理与可视化展示，适用于大规模车队的统一监控与运维。

## 1.6 设计流程

为了实现系统的整体功能，设计时采用了模块化与分层架构原则，将各部分逻辑实现进行封装，只留出接口便于调用。首先，基于 RK3588 芯片构建硬件平台，采用两路摄像头进行状态检测，设计生理参数检测模块，I2S 接口的扬声器

等。然后自建数据集，使用 YOLOv8 训练模型并转化为 RKNN 格式便于部署到边缘端。最后搭建上位机，采用多线程的方式来实现摄像头采集和界面更新互不影响，将各个实现逻辑进行模块化封装，便于调试和调用方便。最后对所完成的系统进行实际模拟测试，测试是否达到预期需求。

## 第二部分 系统组成及功能说明

### 2.1 整体介绍

本系统整体架构如图所示，主要由运行于边缘侧的安全监测终端和运行于云端的远程监控中心两大部分构成。系统细节如图 2-1 所示。

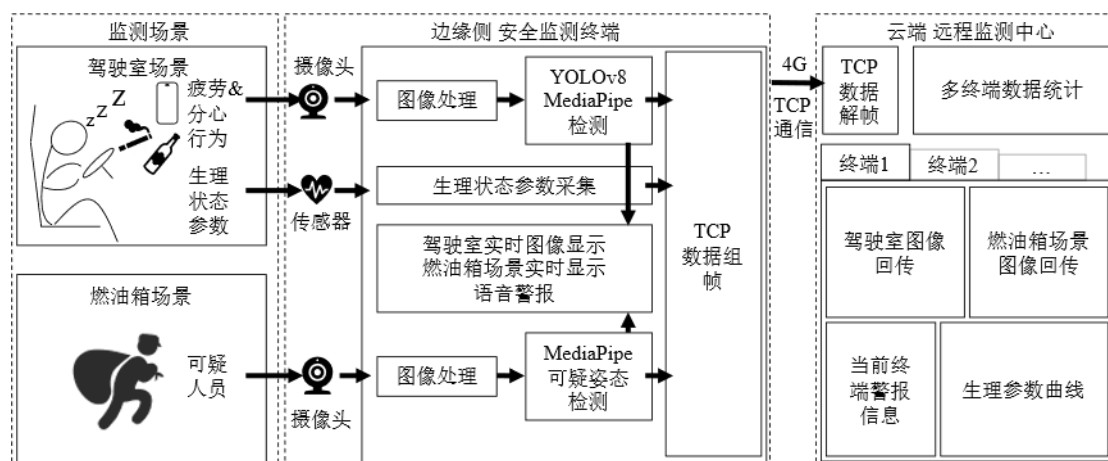


图 2-1 系统整体框图

边缘侧的安全监测终端基于 RK3588 边缘计算平台，集成 YOLOv8 与 MediaPipe 模型，通过摄像头识别驾驶员疲劳状态和分心行为，以及油箱区域可疑人员动作；生理参数模块通过独立传感器采集心率、脉搏与血氧数据。边缘侧上位机显示两大场景实时图像并综合判断当前安全状态，动态触发自然语音播报。TCP 组帧模块将上述数据以及判断结果组合成 TCP 数据包，通过 4G 网络上传至远程监测中心。

云端的远程监测中心基于 Qt 6 设计，分为 TCP 解帧模块、多终端数据统计区域以及各终端的独立显示 Tab 页。Tab 页中显示驾驶室回传图像、油箱场景回传图像、警报信息、生理参数曲线。同时远程监测中心同样可以实现动态警报，

以便管理人员实时掌握车队情况。

系统实现了端 - 云协同的安全预警与多车数据集中管理。该架构具备模块化、实时性强、可扩展性高等特点,适用于多种复杂运输场景下的智能安全监测需求。

## 2.2 硬件系统介绍

### 2.2.1 硬件整体介绍

考虑到生理状态参数的获取需要依赖接触式传感器,且需要长时间佩戴,本文在设计生理参数检测模块时采用模块独立的设计,模块与终端采用 TCP 包进行通信,并配备一块 3.7 V 容量为 500 mAh 的可充电式聚合物锂电池。

模块主要结构包括电源模块电路、STM32 最小系统、通信模块电路以及传感器接口电路构成,如下图 2-2 所示。

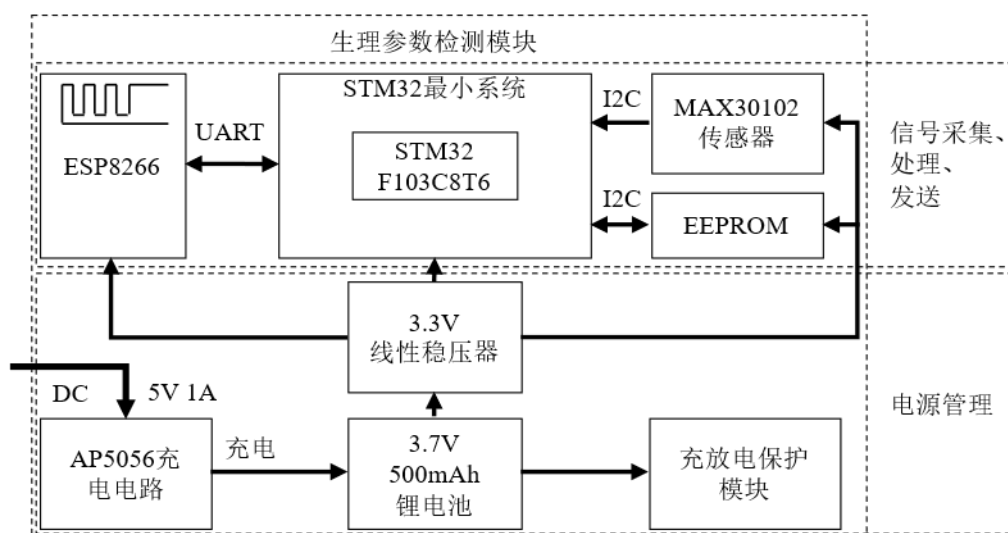


图 2-2 硬件整体架构

以下是各模块的详细硬件设计。

### 2.2.2 电源管理模块设计

电源管理模块电路包括 AP5056 充电电路、充放电保护电路。

充电电路中, AP5056 作为核心电源管理芯片,负责对 3.7V 锂电池进行恒流恒压充电,该芯片支持通过 Micro-USB 接口输入,具备充电状态指示、过温保护、短路保护等功能。充电电路如图 2-3 所示。



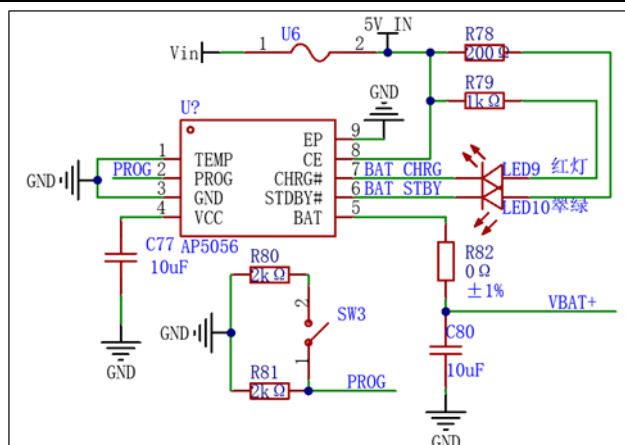


图 2-3 AP5056 充电电路

充放电保护模块采用 DW01 与双 NMOS 管 8205A 组成的锂电池保护电路。DW01 负责检测电池电压与电流状态，提供过充、过放、过流与短路保护功能；8205A 控制充放电通路的导通，具备低导通电阻与高耐压特性。该保护电路可在异常状态下自动切断电池通路，并在恢复正常后自动重启。充放电保护电路如图 2-4 所示。

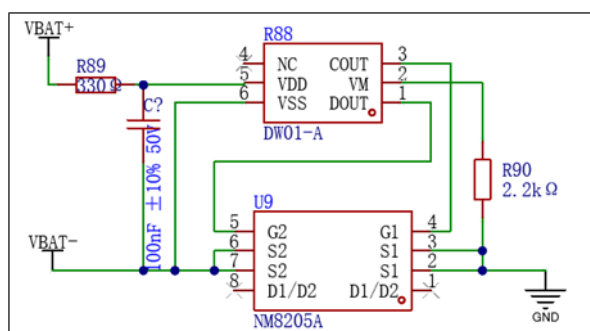


图 2-4 充放电保护电路

线性稳压器将外部输入的 5V 电源降压为 3.3V，为 STM32 芯片、EEPROM、MAX30102 传感器以及 ESP8266 提供电源。

### 2.2.3 STM32 最小系统设计

在本系统的生理参数采集模块中，采用 STM32F103C8T6 构建最小系统作为主控核心。最小系统原理图设计包括基本的电源管理、时钟电路与复位电路，电路图如图 2-5 所示。





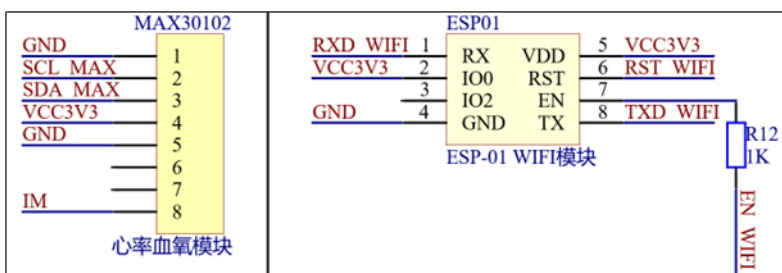


图 2-6 传感器接口电路图、ESP8266 接口电路图

### 2.2.5 EEPROM 模块设计

在本系统的生理参数采集模块中,外接 EEPROM 用于存储关键配置信息与临时生理数据缓存,提升系统的稳定性与断电恢复能力。其主要功能包括:

存储 Wi-Fi 配置信息（如 SSID、密码、服务器地址），避免每次上电重新配置；缓存心率与血氧历史数据，在网络中断或主控未响应时进行本地暂存。系统中采用常见的 I2C 接口 EEPROM，与 STM32F103C8T6 通过 I2C 总线连接。SCL 和 SDA 分别接至 STM32 的 PB6 和 PB7 引脚，芯片地址通过 A0~A2 引脚配置，WP（写保护）引脚接地以允许写入操作。电源由 3.3V 提供，VCC 与 GND 分别连接电源正负极，I2C 总线两端各接一只 5.1kΩ 上拉电阻。

原理图如图 2-7 所示。

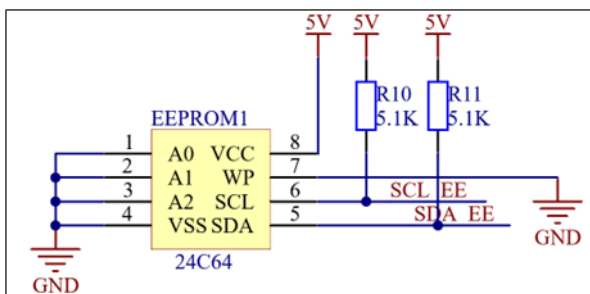


图 2-7 EEPROM 电路图

### 2.2.6 PCB 布局设计

本系统的 PCB 采用双层板设计,着重考虑了模拟信号完整性与电源完整性。PCB 布局如图 2-8 所示。

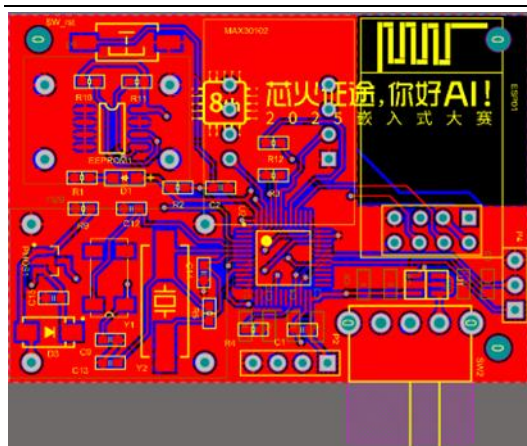


图 2-8(a) 模块 PCB (2 维视图)

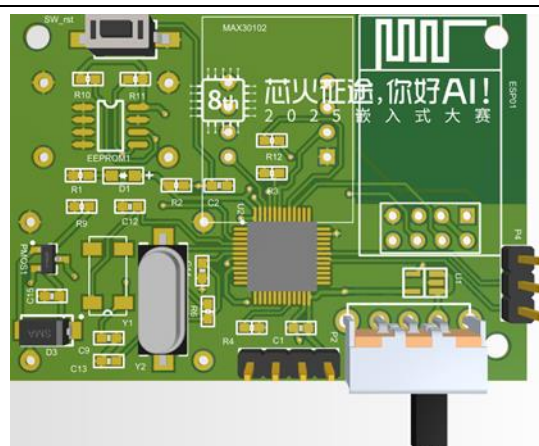


图 2-8(b) 模块 PCB (3 维视图)

器件布局方面，STM32F103C8T6 位于板中心，周边依次布置 MAX30102 传感器接口、ESP8266 通信模块接口、EEPROM 存储器及电源管理电路，避免模拟与数字信号交叉干扰。3.3V 电源通过 LDO 稳压供给，分别为主控、传感器、通信模块供电，关键器件附近布置 0.1uF 和 10uF 去耦电容，提升电源稳定性。

信号布线方面，I2C 总线（SCL、SDA）采用等长布线，避免过长引起信号反射；MAX30102 的 LED 驱动与信号采集路径远离 ESP8266 天线区域，降低射频干扰。地平面在设计时尽量保证了平面完整，同时为了保证 ESP8266 的射频性能，其天线区域下方没有铺铜。

## 2.3 软件系统介绍

### 2.3.1 软件整体介绍

软件整体分为两个部分，其一是运行于 ELF2 开发板上的安全监测终端；该终端通过摄像头和传感器采集实时数据，并通过部署于板端的 YOLOV8 模型和 MediaPipe 模型对数据进行处理，将处理结果通过 4G 模块发往云端的远程监控中心。同时使用 PYQT6 设计了界面，能够进行配置传感器和实时显示驾驶室和油箱的画面。

其二是运行于云端的远程监控中心。云端使用 QT 构建界面，便于可视化处理结果。在接受到板端发来的数据后，根据发送端的组帧协议对 TCP 进行解帧，获取数据后进行处理并在远程监控中心进行显示。

软件系统整体框图如下图 2-9 所示：

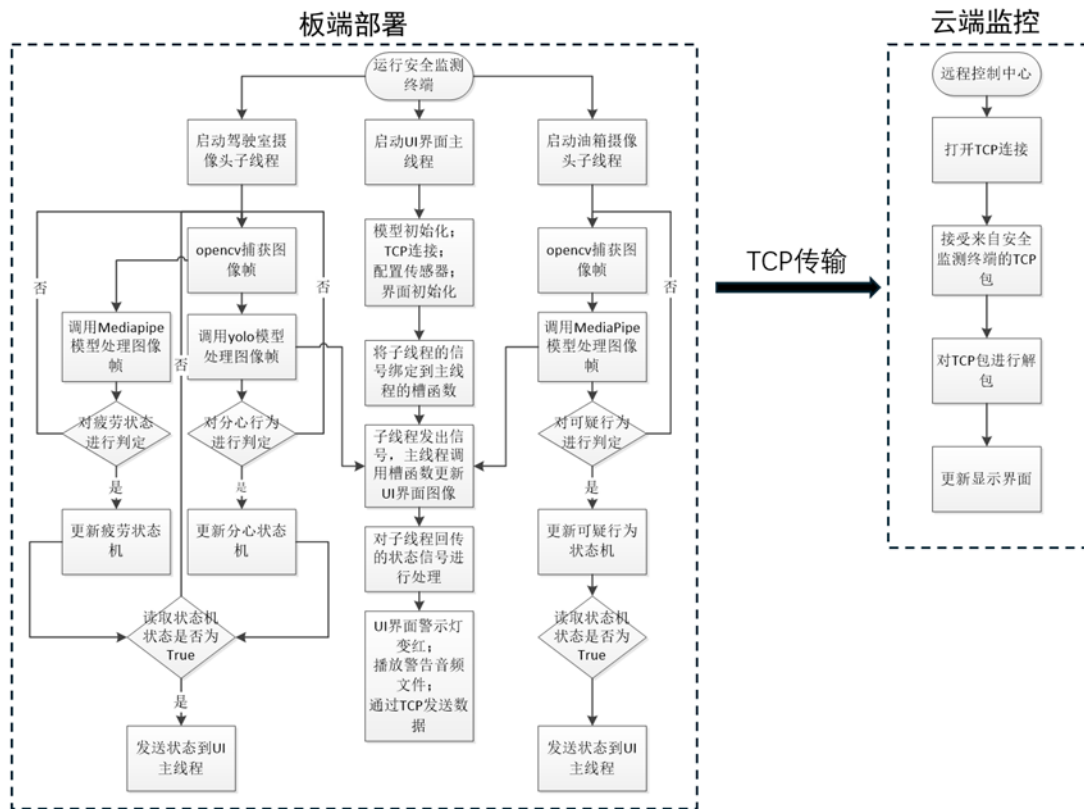


图 2-9 软件系统整体框图

### 2.3.2 驱动程序移植

为了实现本系统实际部署于车辆中时，需要采用无线传输的方式将数据回传至云端，因此在软件层面要对 4G 驱动程序进行移植；为了实现自主设计的生理参数检测模块能够与服务器端建立数据链路，需要对 WIFI 驱动进行移植。

#### (1) 4G 驱动程序移植

为了实现 ELF2 开发板在实际应用场景中的无线数据传输功能，本系统完成了 4G 通信模块的驱动移植与优化工作。系统采用合宙 AIR780 系列 4G 通信模组，该模组支持 LTE Cat1 标准，具有低功耗、高可靠性的特点，非常适合移动场景下的数据传输需求。

在软件驱动层面，我们采用分层架构的设计思想。整个驱动移植过程严格遵循 Linux 设备驱动开发规范，确保了系统的稳定性和兼容性。具体驱动设计流程如图 2-10 所示，包括添加模块的 VID&PID、AT 指令交互、网络连接建立和数据传输等关键步骤。

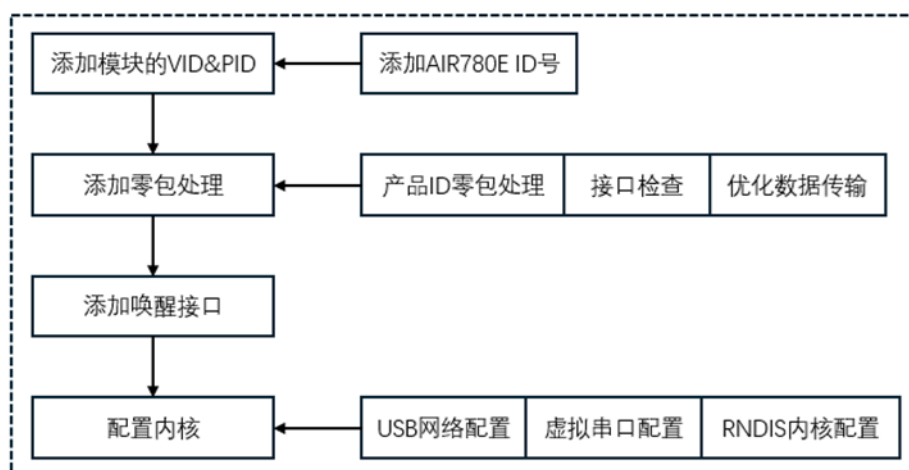


图 2-10 4G 驱动移植

## (2) WIFI 驱动程序移植

为实现自主设计的生理参数检测模块检测到的数据能够进行无线回传服务端，本系统对无线通信模块进行了深度适配与优化。针对 WIFI 数据传输需求，重点适配了 Realtek 系列无线芯片（包括 RTL8723、RTL8192、RTL8822CS 等主流型号），通过定制化驱动开发实现了稳定的无线数据传输通道。具体设计流程如图 2-11 所示，系统能够将各类传感器采集的生理参数（包括心率、血氧等）实时封装为 TCP 数据包，并可靠地传输至云端服务器。

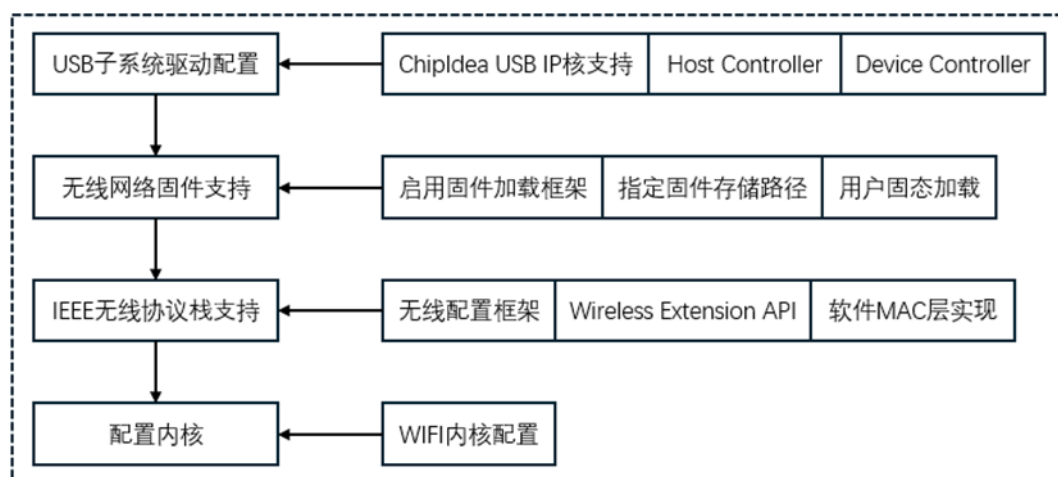


图 2-11 WIFI 驱动移植

## 2.3.3 检测算法移植

由于需要实时监测司机状态和油箱安全，因此本系统于 ELF2 开发板上部署轻量级的目标检测算法 YOLOv8，以及对关键点进行检测的 MediaPipe 模型。以

下是对模型部署的详细介绍：

### (1) 部署 YOLOv8

本系统采用 YOLOv8 对司机的分心行为，如抽烟，玩手机和喝水等进行检测。部署过程大致分为模型训练，模型转换，模型部署，其流程图如图 2-12 所示：

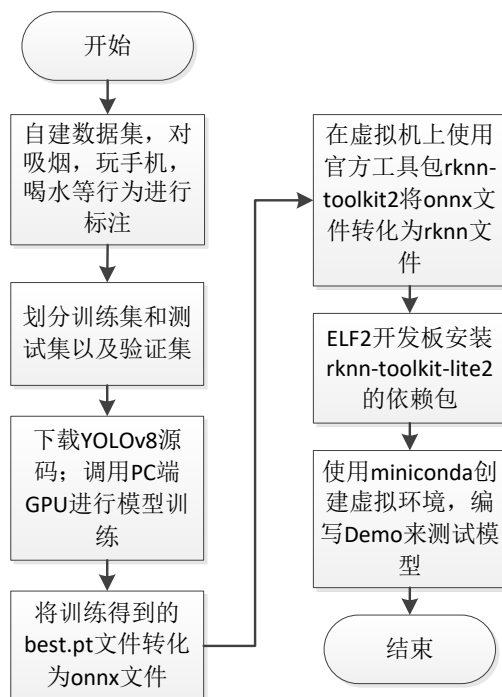


图 2-12 板端部署 YOLOv8 流程图

首先根据本系统需求，构建包含吸烟，玩手机，喝水的数据集，并按照 8: 1: 1 的比例对数据集进行划分，如下图所示为自主构建的数据集以及划分结果。

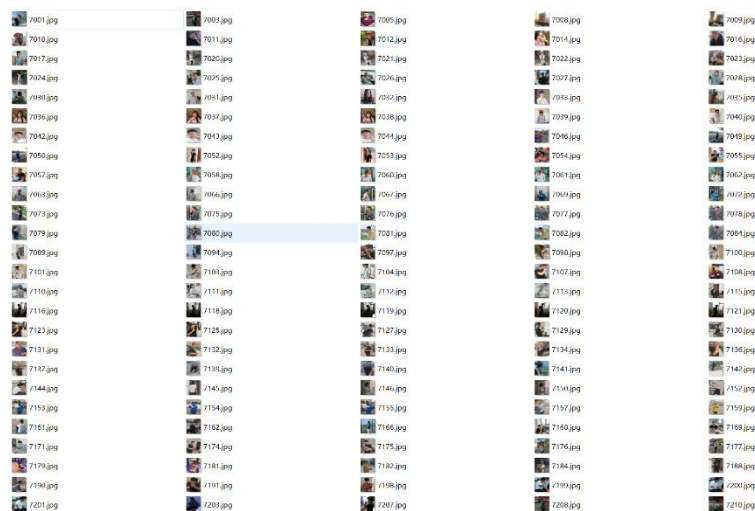


图 2-13 自主构建训练数据集



名称	修改日期	类型	大小
images	2025/6/24 16:14	文件夹	
labels	2025/6/24 17:22	文件夹	
information.txt	2025/6/24 16:25	文本文档	1 KB

图 2-14 划分数据集

接着调用 GPU 对模型进行训练，选择初始模型为 yolov8n.pt，训练次数为 300 轮，如下图所示为模型训练过程。

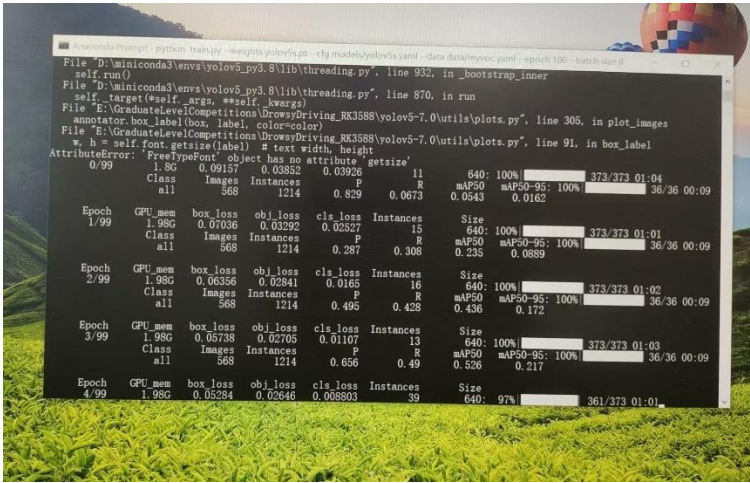


图 2-15 模型训练过程

模型训练结束后调用 YOLOv8 源码中的 export.py 文件将 best.pt 文件转化为 onnx 文件，以便后续模型部署于板端。同时可以使用 Netron 来查看已经转换的 onnx 模型。接着在所提供的开发环境中将 onnx 文件转化为 rknn 文件，如下图 2-16 所示为本系统所使用到的已经转化后的模型。

yolov8n.onnx	2025/6/26 16:36	ONNX 文件	28,433 KB
yolov8n.pt	2025/5/29 17:28	PT 文件	14,449 KB
yolov8n.rknn	2025/6/26 18:00	RKNN 文件	8,298 KB

图 2-16 模型转换

最后在板端安装软件 miniconda，并且新建一个虚拟环境。在虚拟环境中安装 rknn-toolkit-lite2 里面对应的依赖包，通过编写测试 Demo 对后处理部分进行重写，调用模型对摄像头画面进行处理来验证模型部署的成功。如下图 2-17 所示为 YOLOv8 的后处理部分函数：



```

20
21 #将任意实数 x 映射到 (0, 1) 范围内进行归一化, 方便后续进行操作
22 > def sigmoid(x): ...
24
25 #将输入图像按比例缩放到目标尺寸 new_shape, 并在周围填充指定颜色 (默认黑色), 同时保持原始图像的宽高比不变。
26 > def letter_box(im, new_shape, pad_color=(0, 0, 0), info_need=False): ...
54
55 #用于过滤低置信度检测框
56 > def filter_boxes(boxes, box_confidences, box_class_probs): ...
72
73 #去除高度重叠的冗余检测框, 保留置信度最高的框
74 > def nms_boxes(boxes, scores): ...
106
107 > def softmax(x, axis=None): ...
111
112 #DFL 的核心思想是将边界框坐标的预测建模为离散概率分布, 而不是直接回归连续值。这种方法能更精确地定位物体, 特别是对小物体效果显著。
113 > def dfl(position): ...
123
124 #用于将模型输出的边界框位置信息转换为实际坐标的关键函数
125 > def box_process(position): ...
139
140 #后处理部分, 将模型原始输出转换为最终的检测结果 (边界框、类别、置信度)
141 > def post_process(input_data): ...
189
190 #目标检测结果的可视化函数, 用于在图像上绘制检测到的物体边界框和相关信息
191 > def draw_detections(image, boxes, scores, classes): ...

```

图 2-17 后处理部分的函数

## (2) 部署 MediaPipe

板端部署的系统要求高实时性以及低延迟性, 在比较了 YOLOv8+Dlib 算法的组合以及 MediaPipe 后。我们选择了高性能, 低延迟的 MediaPipe 模型作为检测油箱附近可疑人员行为。

其官网提供了对人体姿态估计的预训练模型, 因此本文并未对其进行重新训练, 只是在此基础上构建了判断可疑行为逻辑。部署的主要过程如下所示:



图 2-18 部署 MediaPipe 流程图

### 2.3.4 边缘端多模态安全检测软件设计

为了方便司机随时掌握自身疲劳状态和分心行为, 以及油箱附近可疑人员行

踪，在边缘端设计了上位机界面。可以对自主设计的生理参数模块进行配置，实时显示当前两路摄像头的画面，以及展示当前司机的疲劳状态和分心行为，油箱附近人员的可疑行为。如下图 2-19 所示为设计的界面。



图 2-19 边缘端多模态安全检测软件界面

为了实现边缘端的实时性以及高可靠性，在进行软件设计的时候采用了多线程，模块化的技术。主要分为三个线程，其中子线程 1 为驾驶室摄像头线程，子线程 2 为油箱摄像头线程，主线程为 UI 线程。三者之间的关系如下图 2-20 所示：

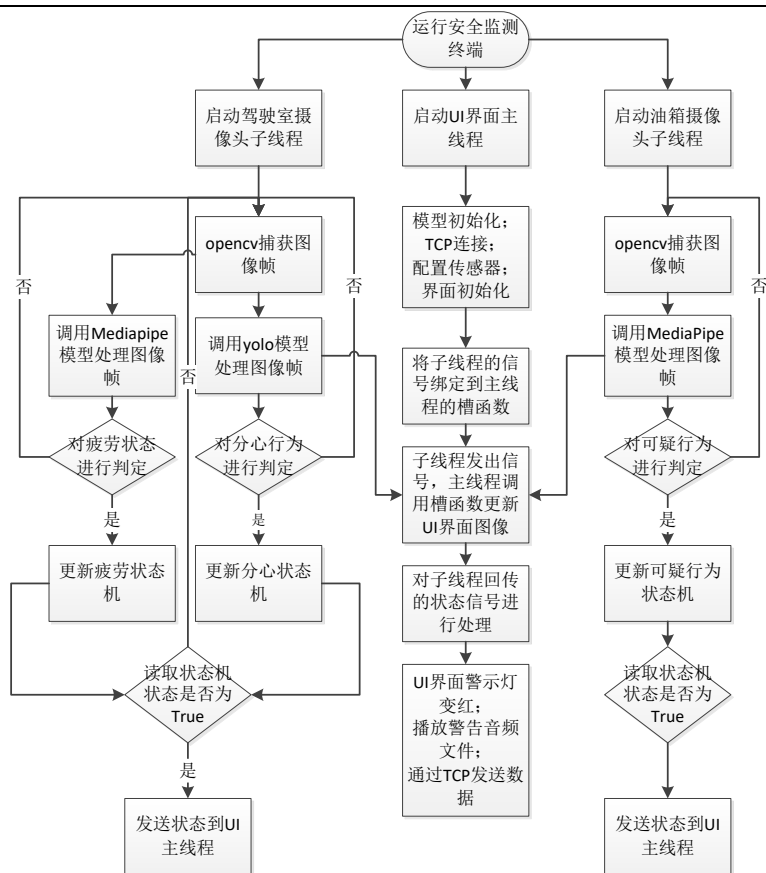


图 2-20 边缘端多模态安全检测软件流程图

### (1) 驾驶室摄像头子线程

其运行逻辑大致分为：首先通过 `opencv` 调用 MIPI 摄像头，将捕获到的图像帧进行格式转换；然后将转换后的图像帧作为输入，调用 YOLOv8 以及 MediaPipe 同时对其进行处理；对于模型处理后的数据进行逻辑判断，如果检测到了疲劳和分心行为则更新状态机的状态；最后将状态机的状态以及可视化结果的图片通过信号与槽机制发往边缘端，通过将图片和类型进行组包以 TCP 的方式发往云端。为了不让边缘端占用大量的带宽，我们设定了发往云端的数据只有在检测到疲劳状态和分心行为后才进行发送，而且上一次发送和本次发送时间间隔部的少于 5 秒。如下图 2-21 为其具体的流程图：

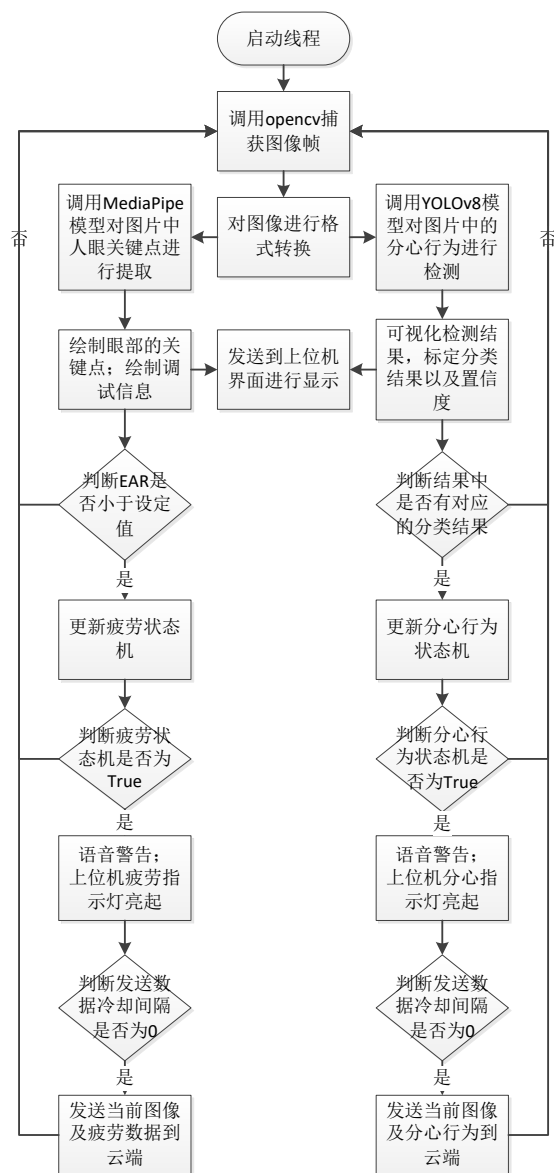


图 2-21 驾驶室摄像头子线程的流程图

## (2) 油箱摄像头子线程

其运行逻辑为：首先通过 opencv 调用 USB 摄像头获取到图像帧，并对图像帧进行格式转换；将转换完成的图像帧作为输入传给 MediaPipe 模型；模型对图像中人体姿态的关键点进行检测，并且返回带关键点的坐标；接着编写判断逻辑对可疑行为进行判定，例如：由返回的关键点计算角度，由此来判断是否弯腰；通过对比连续几帧内脸部的关键点位置来判断是否为左顾右盼。然后更新状态机，并将图像发送到上位机界面；如果状态机的状态为 True，那么则进行语音播报，邮箱安全指示灯变红。最后判断发送冷却时间，如果距离上一次发送数据到云端

的时间已经超过 5 秒，那么进行本次的发送，否则不进行发送。这样做的目的是减少边缘段所占用的带宽，去除不必要的图像，使得系统更高效化。

### (3) UI 界面主线程

最后介绍 UI 主线程，该线程的主要目的就是实时更新上位机界面，因此不能在该线程中运行耗时任务，这样会阻塞界面的更新。当程序启动后，该线程会首先加载模型，与云端建立 TCP 连接，显示初始化信息，启动子线程等，然后持续监听来自子线程的信号。如果收到了信号则调用对应封装好的函数，并对 UI 界面进行更新。

### 2.3.5 云端远程监测软件设计

为了实现终端设备数据的汇总、方便管理人员的集中管理，本文设计了云端远程监测中心软件。软件可以同时连接多个终端，实时显示终端在线数量。通过解析终端回传的 TCP 数据包，对终端上传的数据进行展示，包括司机出现疲劳状态、分心状态以及油箱附近出现可疑人员时的图像、司机的实时生理状态参数（脉搏、心率、血氧）；根据终端回传数据对司机疲劳状态、分心行为、可疑人员动作进行数据统计。所设计的界面如图 2-22 所示。



图 2-22 云端远程监测软件界面

如图所示，界面上半区域为系统功能配置区、多终端数据总览区、信息交互区。下半区域为多终端独立的 Tab 页，用于显示终端实时数据，包括驾驶室回传

图像、油箱回传图像、异常状态统计、生理状态参数曲线图。

对于后端处理模块，主要由 TCP 服务控制、客户端连接处理、数据接收处理三大核心模块组成。以下是对这三部分的详细描述。

### (1) TCP 服务控制流程

主要进行端口冲突处理、状态同步以及对端口进行监听。其执行流程图如图 2-23 所示。

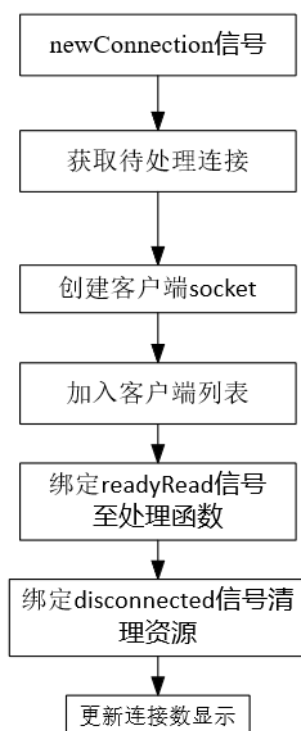


图 2-23 TCP 服务控制流程图

如图所示,TCP 服务控制流程通过 TCP 开启按钮槽函数实现服务启停控制。当点击按钮时,程序首先检查 TCP 服务器是否正在监听:若已运行,则关闭服务并更新 UI 状态;若未运行,则尝试绑定指定端口并启动监听。启动成功后会更新日志和按钮文本,失败则显示错误原因。核心是通过 QTcpServer 的 listen() 和 close()方法实现服务状态切换,同时保持 UI 与底层状态的实时同步。

### (2) 客户端连接处理流程

当新客户端连接时,newConnection 信号触发槽函数,通过 nextPendingConnection()逐个处理待连接请求。每建立一个连接,程序首先将 socket 加入客户端列表 clientSockets,然后绑定 readyRead 和 disconnected 信号到



对应的数据处理和清理槽函数，最后为该客户端单独创建心跳检测定时器。连接断开时，程序会自动移除 socket 并释放资源，同时更新界面显示的连接数。关键点是通过 `QMap<QTcpSocket*, ImagePacket>` 为每个连接维护独立的图像接收状态机。如图 2-24 所示为其流程图。

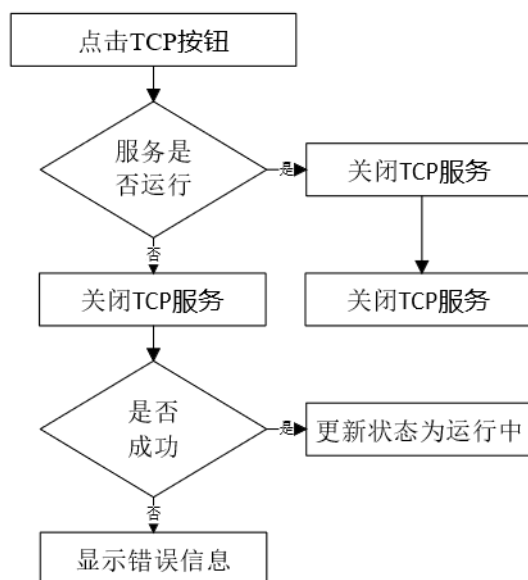


图 2-24 客户端连接处理流程图

### (3) 数据接收处理流程

主要包括心跳包处理、图像数据接收、图像数据解码、生理参数数据处理、异常数据处理。处理流程如图 2-25 所示。

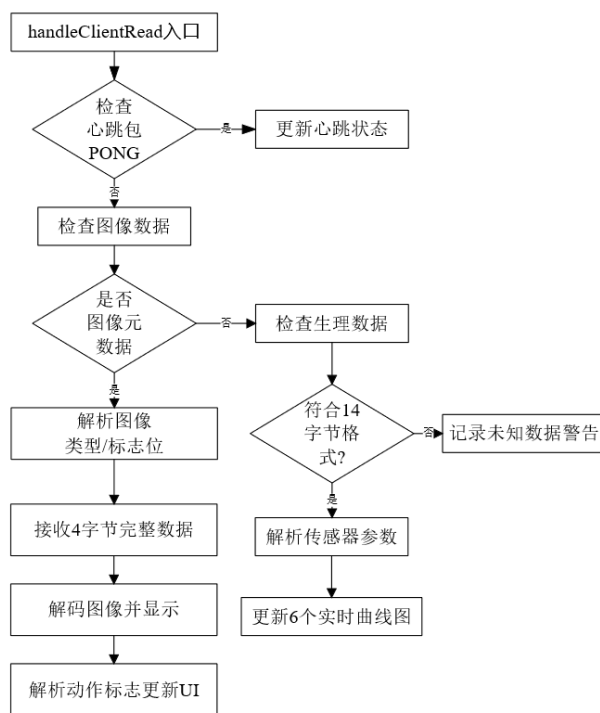


图 2-25 数据接收处理流程图

### （3.1）心跳包处理

当收到客户端发来的 PONG 数据包时，服务器会立即更新 UI 界面上的心跳状态显示为"客户端心跳回应"，并通过查找 heartbeatTimers 哈希表获取该客户端对应的心跳超时定时器，调用 start()方法重置倒计时。如果未及时收到心跳回应，定时器超时会自动触发客户端断开逻辑，将界面提示更新为"客户端断连"。这种机制确保了连接状态的实时监控，核心是通过简单的字符串比对 (buf == "PONG") 实现轻量级心跳检测。

### （3.2）图像元数据解析阶段

当检测到数据包以"IMG0"标识开始时，进入图像接收状态。首先提取 8 字节元数据，通过 QDataStream 解析出图像类型 imageType 和动作标志位 actionFlags。例如驾驶室图像类型为 1，油箱为 2，同时将动作标志位转换为十六进制格式输出到日志。此时状态机切换到"等待图像大小"阶段，清空已处理的元数据字节，准备接收后续数据。

### （3.3）图像大小接收阶段

在元数据接收完成后，程序等待接下来的 4 字节图像长度信息。使用大端序

解析将 4 字节转换为整型数值，记录到当前连接的 `ImagePacket.imageSize` 字段。例如收到字节序列 `00 00 30 00` 会解析为 12288 字节的图像数据长度。随后状态机进入最终的数据接收阶段，保留缓冲区中未被消耗的字节用于后续拼接。

#### （3.4）图像数据接收与解码

当累积数据量达到 `imageSize` 指定的大小时，程序取出完整图像字节流，使用 `QImage::loadFromData()` 进行解码。成功解码后根据 `imageType` 将图像显示到对应的 UI 标签，并自动缩放保持比例。如果解码失败，会在日志中输出"图像解码失败"，并丢弃该数据包。无论成功与否，最后都会清空该连接的状态机数据，准备处理下一帧。

#### （3.5）动作标志位处理

图像接收完成后，程序会按位解析 `actionFlags`。对于驾驶室图像（类型 1），第 0 位表示瞌睡，第 1 位表示玩手机；对于油箱图像（类型 2），第 0 位表示弯腰伸手。通过将动作标志位移位检测来检测特定位是否置位，若为真则找到对应的 `QLineEdit` 计数器，读取当前值并+1 更新。所有触发的动作名称会拼接成字符串输出到日志。

#### （3.6）生理参数数据处理

当收到 14 字节的传感器数据时，程序首先校验数据头。通过位运算组合出 Red 值、IR 值、体温等参数。例如体温字节 `0x1A 0x40` 会被解析为 26.25℃。数据验证通过后，程序计算当前时间戳，将参数追加到对应的曲线序列，并动态调整 X 轴范围实现滑动窗口效果（仅显示最近 5 秒数据）。

#### （3.7）异常数据处理

对于无法识别的数据包（非心跳、非图像、非生理数据），程序会记录警告日志，包含数据长度和原始字节的十六进制预览。遇到错误会立即中断当前数据包处理，避免状态机混乱。当客户端断开时会通过 `disconnected` 信号触发资源清理，确保无内存泄漏。

## 第三部分 完成情况及性能参数

### 3.1 整体介绍

所设计的系统主要分为两部分，一部分为部署于边缘端的货运车辆安全监测终端，一部分为部署于服务器端的云端远程安全检测中心。

终端侧基于飞凌 ELF2 RK3588 开发板设计的货运车辆安全监测终端，整体实物如图 3-1、3-2 所示。

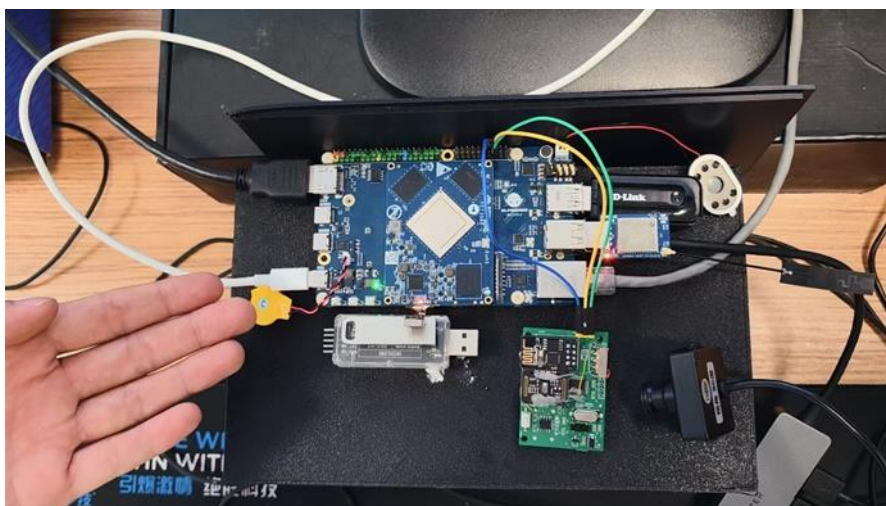


图 3-1 终端整体示意图（正视图）

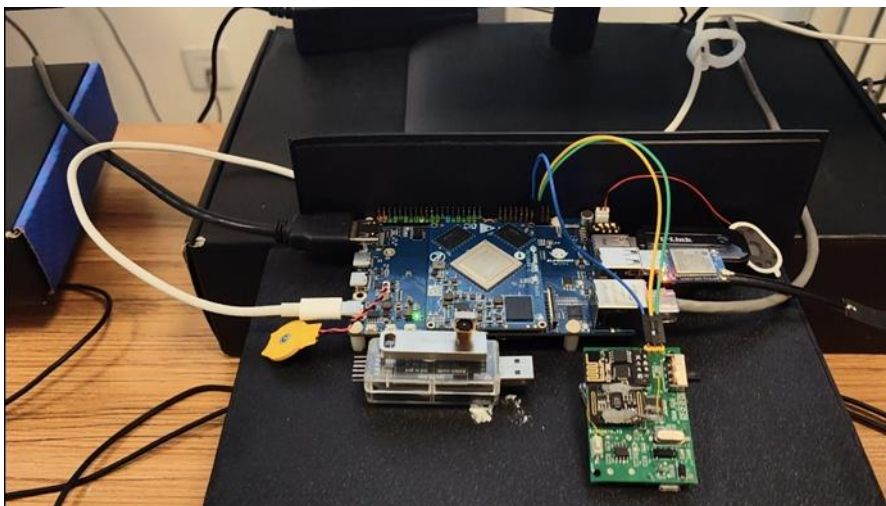


图 3-2 终端整体示意图（45°）

如图所示，终端连接有两个摄像头，一路 USB 摄像头拍摄驾驶室场景，用于判断司机疲劳状态与分心行为，一路 MIPI 摄像头拍摄货运车辆的燃油箱场景，用于监测可疑人员姿态。一个分离式生理参数检测模块，用于实时监测司机血氧、心率等生理参数，该模块依赖 TCP 与终端进行通信，通过串口配置模块的 TCP 连接参数，实际运行时不需要与终端有线连接。板上搭载一个 4G 模块以及一个

WIFI 模块，分别用于与终端与远程监测中心的通信和终端与生理检测模块通信。  
一个 I2S 接口的扬声器，用于对当前安全状态进行动态警报。

边缘端基于 PYQT6 设计了终端上位机界面，主要包括配置分离式生理参数检测模块的 IP 和端口号、两路摄像头实时画面、行为状态指示灯。设计的界面如图 3-3 所示。



图 3-3 货运车辆安全检测终端

云端侧基于 Qt 设计了货运车辆远程安全监测中心上位机软件，主要包括 TCP 解帧、图像显示、生理参数曲线显示、数据统计以及日志记录等功能。界面如图 3-4 所示。



图 3-4 云端远程安全检测中心

## 3.2 系统测试及可靠性分析



### 3.2.1 测试平台搭建

将终端的 MIPI 摄像头、USB 摄像头连接，并将生理参数监测模块与终端使用串口连接。首先打开远程安全监测中心软件，打开网络服务，然后打开终端上位机软件，终端经 TCP 连接后，远程安全监测中心显示在线设备数量加 1。整体搭建的测试平台如图 3-5 所示。



图 3-5 测试平台搭建

### 3.2.2 系统测试

#### (1) 驾驶室场景测试

首先测试驾驶员疲劳状态监测功能，驾驶员需要做出闭眼 1s 的动作。测试结果如图 3-6 所示。

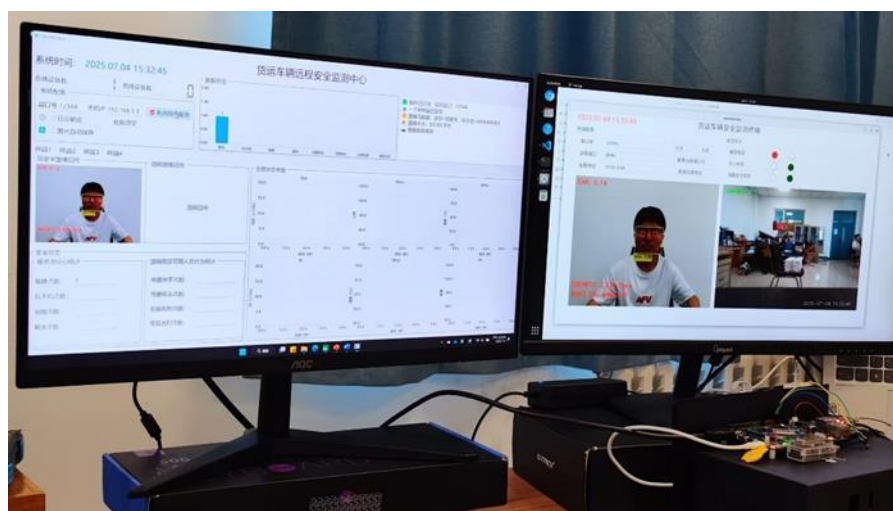


图 3-6 疲劳状态检测



可以看到，右侧安全监测终端的上位机实时显示到了驾驶员的疲劳画面，右上角状态显示区域的疲劳状态警报变红。同时，左侧云端远程安全监测中心的驾驶室图像回传区域显示到了疲劳状态照片。同时柱状图中瞌睡状态加 1。

其次，模拟测试人员作出玩手机动作，测试结果如图 3-7 所示。

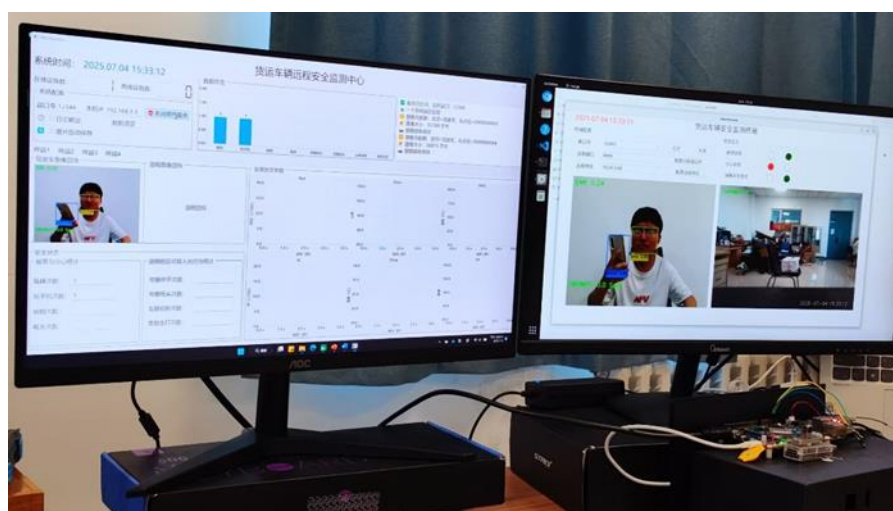


图 3-7 玩手机状态检测

可以看到，右侧安全监测终端的上位机实时显示到了驾驶员玩手机的分心画面，右上角状态显示区域的分心状态警报变红。同时，左侧云端远程安全监测中心的驾驶室图像回传区域显示到了分心状态照片。同时柱状图中玩手机行为加 1。

随后进行分心行为中抽烟行为的测试。驾驶员需要做出抽烟的动作。测试结果如图 3-8 所示。

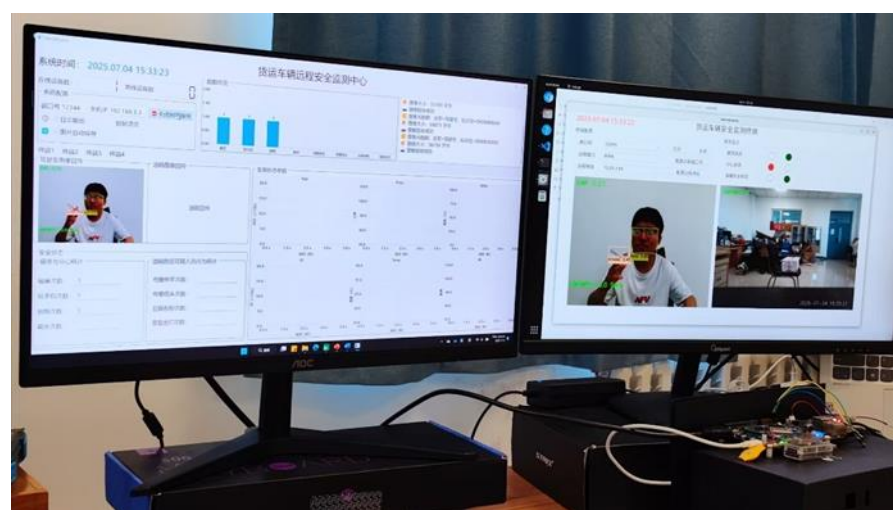


图 3-8 抽烟状态检测

右侧安全监测终端的上位机实时显示到了驾驶员抽烟的分心画面，右上角状态显示区域的分心状态警报变红。同时，左侧云端远程安全监测中心的驾驶室图像回传区域显示到了分心状态照片。同时柱状图中抽烟行为加 1。

随后进行分心行为中喝水行为的测试。驾驶员需要做出喝水的动作。测试结果如图 3-9 所示。

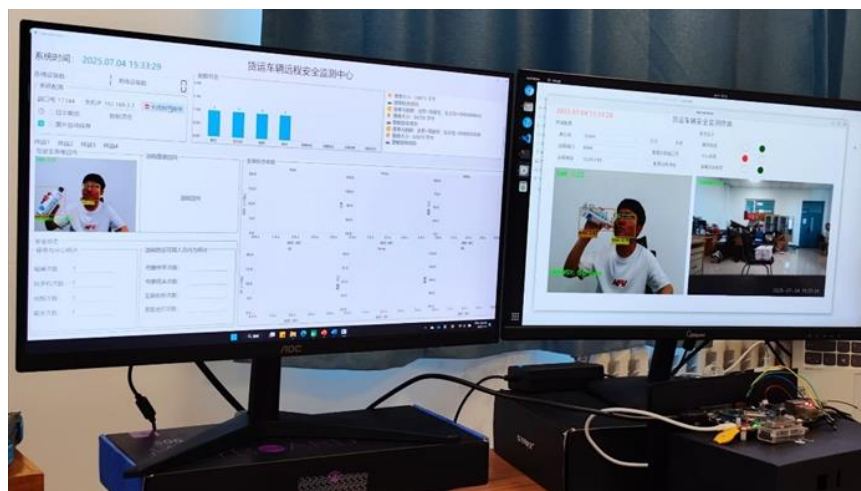


图 3-9 喝水状态检测

可以看到，右侧安全监测终端的上位机实时显示到了驾驶员喝水的分心画面，右上角状态显示区域的分心状态警报变红。同时，左侧云端远程安全监测中心的驾驶室图像回传区域显示到了分心状态照片。同时柱状图中喝水行为加 1。

最后进行生理状态参数监测测试。测试人员需要将手指与生理参数模块上的传感器轻轻接触，测试结果如图 3-10 所示。

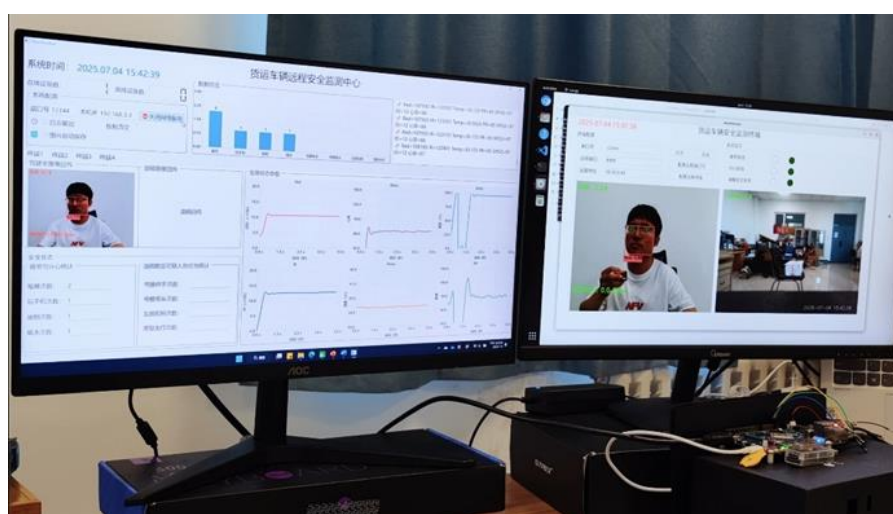


图 3-10 生理参数采集

可以看到，测试人员轻触生理参数监测模块，左侧云端远程安全监测中心的生理参数曲线区域稳定后，数值处于正常范围。

## （2）油箱场景测试

油箱场景中，当画面中出现人员做出弯腰伸手、弯腰低头、左顾右盼以及使用工具重复击打的动作时，摄像头能够进行识别、警报并上传至远程安全监测中心。

测试人员做出弯腰伸手的动作后，可以看到终端侧和云端侧均出现了监测结果，如图 3-11 所示。

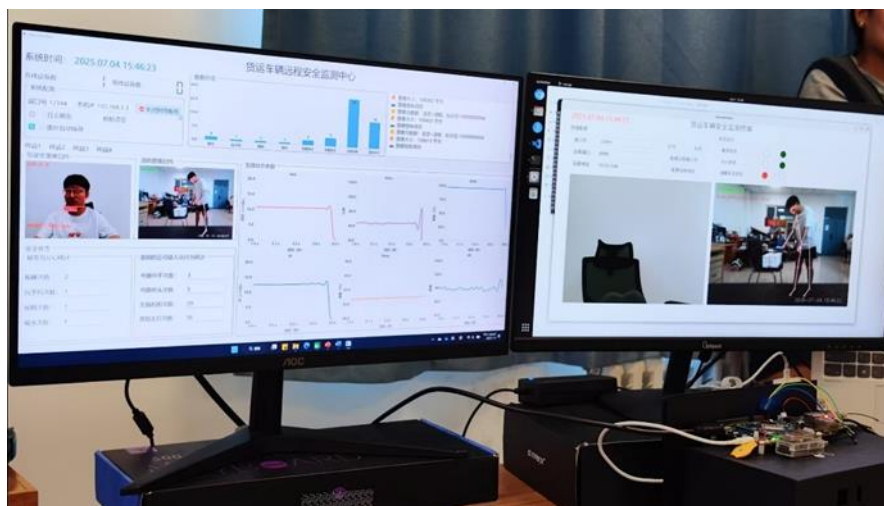


图 3-11 可疑动作检测——弯腰伸手

可以看到，测试人员做出弯腰、伸手的动作后，摄像头捕捉到的画面经 MediaPipe 提取人体关键点后，正确识别到了弯腰伸手动作，同时左侧云端远程安全监测中心油箱场景接收到测试人员的可疑动作，并且“弯腰伸手”动作加 1。

测试人员做出弯腰低头的动作后，可以看到终端侧和云端侧均出现了监测结果，如图 3-12 所示。

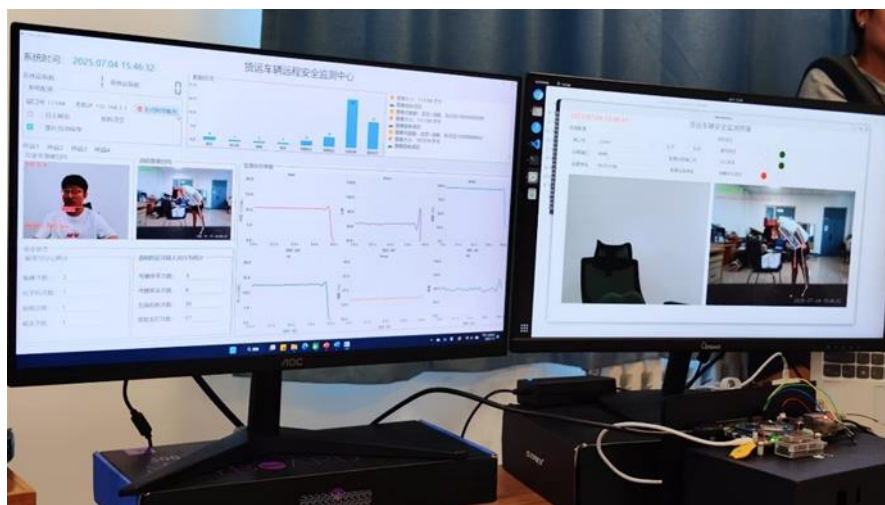


图 3-12 可疑动作检测——弯腰低头

测试人员做出左顾右盼的动作后，可以看到终端侧和云端侧均出现了监测结果，如图 3-13 所示。

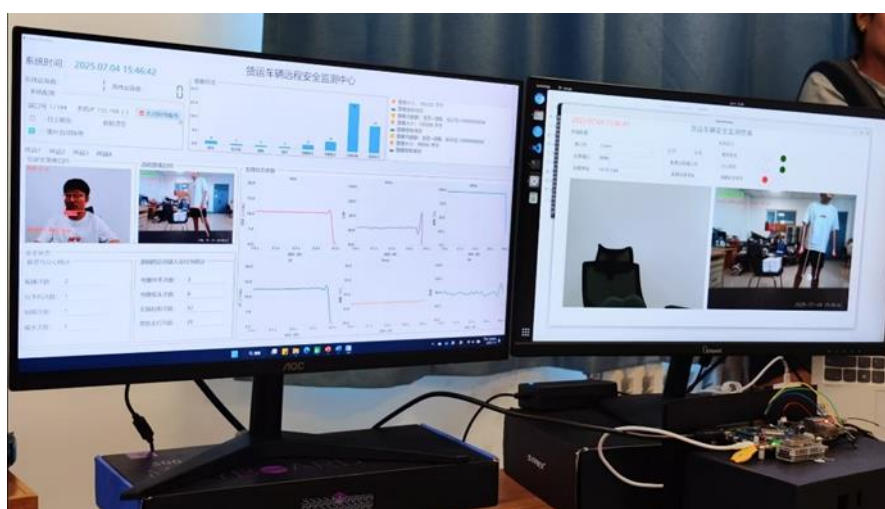


图 3-13 可疑动作检测——左顾右盼

测试人员做出使用工具重复击打的动作后，可以看到终端侧和云端侧均出现了监测结果，如图 3-14 所示。



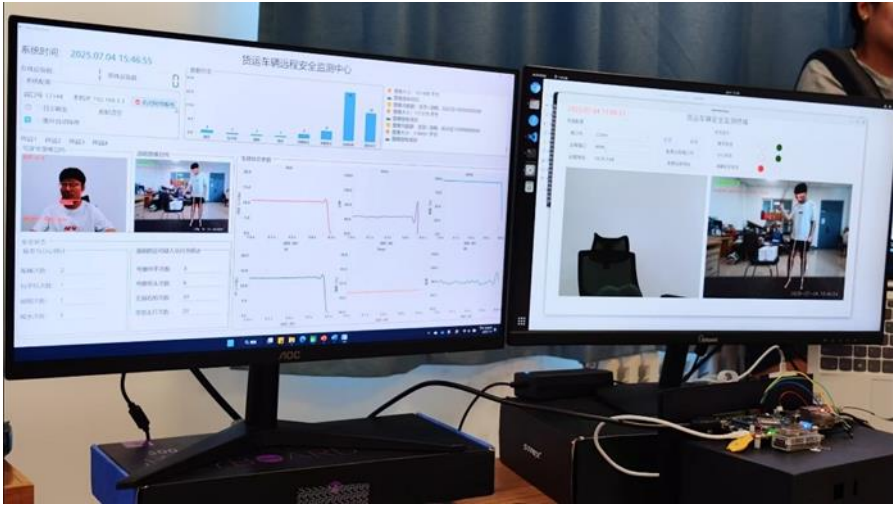


图 3-14 可疑动作检测——使用工具重复击打

3.2.3 可靠性分析

通过对系统的全面测试验证，本系统在安全监测覆盖性、准确性和稳定性三个关键维度均表现出色。在安全监测覆盖性方面，系统完整实现了驾驶室场景的疲劳检测（闭眼 $\geq 1s$ ）、分心行为识别（玩手机、抽烟等）和生理参数监测功能，识别准确率达到 100%；同时油箱场景的可疑动作识别（如弯腰、击打等）也实现无误报、漏报。在准确性测试中，各类复杂动作（包括复合动作如弯腰同时使用工具击打）均能准确识别，未出现行为混淆现象。系统稳定性测试表明，在多任务并发处理驾驶室和油箱监测场景时，系统运行稳定，未出现性能下降情况。这些测试结果充分验证了系统设计的可靠性和实用性。

具体测试结果如表 3-1 所示。

表 3-1 可靠性测试结果

测试类别	内容	结果
安全监测覆盖性	驾驶室场景（疲劳、分心、生理监测）	所有功能模块均被验证，识别准确率 100%
	油箱场景（可疑动作识别）	弯腰、击打等动作识别无误报/漏报
准确性	疲劳/分心行为识别	闭眼 $\geq 1s$ 、玩手机、抽烟等动作识别准确率 100%

	油箱动作识别	复杂动作（如弯腰+工具击打）无混淆
稳定性	多任务处理	同时处理驾驶室与油箱监测时无性能下降

### 3.3 工程成果

#### 3.3.1 电路成果

模块主要结构包括电源模块电路、STM32 最小系统、通信模块电路以及传感器接口电路构成。实物图如图 3-15 所示。



图 3-15 分离式生理参数监测模块

#### 3.3.2 软件成果

软件界面设计包括远程安全监测中心上位机软件和运行于终端的安全预警上位机软件。

远程安全监测中心上位机软件界面如图 3-16 所示。





图 3-16 远程安全监测中心

货运车辆安全监测终端上位机软件如图 3-17 所示。



图 3-17 货运车辆安全监测终端

第四部分 总结

4.1 可扩展之处

本终端还可增强对货运车辆其他部分参数的监测。增加或 CAN 总线采集

车辆油耗、转速、车速等运行状态，通过读取得到发动机转速、瞬时油耗、节气门开度、车速等诊断数据，与现有的行为检测、心率数据融合，以更精准地判断驾驶风险。

可以将边缘采集的多模态数据汇总至云端，基于驾驶员历史生理与行为特征训练个性化疲劳/分心检测模型，持续提升警报准确率；并能将通用模型与个性化模型相互迭代提升识别准确率与预警智能度。

#### 4.2 心得体会

本作品以 RK3588 为核心打造的多模态货运安全监测终端，首要任务是精准提炼“安全监测”核心需求：驾驶行为识别、生理信号采集与环境监控。我们决定先用最少的硬件模块（摄像头、心率带、DHT11 传感器），在 RK3588 开发板上完成端到端数据流通的 Demo。通过一天内几次迭代，我们迅速确认了摄像头分辨率、数据传输接口及实时预警阈值，避免在细节上浪费宝贵时间。

我们首先在 PC 上训练出精度较高的疲劳检测模型，但移植到 RK3588 后出现延迟。经过性能剖析，团队对神经网络进行剪枝与量化，删去冗余通道，将模型体积缩减近一半；同时借助 NPU 加速，把单帧处理时延从 120 ms 降至 45 ms，确保 30 fps 的流畅帧率。

这次嵌入式大赛不仅检验了我们在 RK3588 平台上多模态融合的技术深度，也让我们收获了面对压力、快速迭代与系统化表达的重要经验。比赛环境迫使我们不断在“可行性”与“性能”之间做取舍，培养了快速决策与风险预判能力。未来，我们将继续在“高效”“鲁棒”“可扩展”三大方向上精耕细作，为智能交通与车辆安全贡献更多创新实践。

### 第五部分 参考文献

- [1]. 王亮,夏舟勇,胡营营,等. 基于 CLIP 的多模态融合视频描述生成[J]. 计算机工程与设计,2025,46(2):384-391. DOI:10.16208/j.issn1000-7024.2025.02.009.
- [2]. 廖强,王宇. 基于多模态融合的情绪识别[J]. 科技传播,2024,16(7):12-15. DOI:10.3969/j.issn.1674-6708.2024.07.004.

- 
- [3]. 张连庆.边缘计算安全综述[J]. 智能安全,2023,2(3):105-112. DOI:10.12407/j.issn.2097-2075.2023.02.105.
- [4]. 张俊杰. 基于嵌入式 AI 处理器的安全场景人工智能应用[D]. 陕西:西安电子科技大学,2022.
- [5]. 苏为斌. 论边缘计算元操作系统 [J]. 自动化博览 ,2024,41(2):59-63. DOI:10.3969/j.issn.1003-0492.2024.02.038.
- [6]. 高凤梅,吴攀. 基于嵌入式 AI 的可穿戴健康管理系统设计 [J]. 现代信息科技,2020,4(15):95-97. DOI:10.19850/j.cnki.2096-4706.2020.15.029.
- [7]. 杨涛,杨博雄,尹萍,等. 基于高性能嵌入式 AI 计算平台的人机交互手势控制识别研究[J]. 信息记录材料,2019,20(11):175-177.
- [8]. LIREN YANG, WEI TANG. A Lightweight Stereo Depth Estimation Network Based on Mobile Devices[C]//Seventh International Conference on Computer Graphics and Virtuality (ICCGV 2024): 23-25 February 2024.Hangzhou, China. 2024:1315805.1-1315805.12.
- [9]. 刘鹏,张天翼,冉鑫,等. 基于 PBM-YOLOv8 的水稻病虫害检测 [J]. 农业工程学报,2024,40(20):147-156. DOI:10.11975/j.issn.1002-6819.202405194.
- [10]. 高子召. RK3588 物体检测方法研究[D]. 长江大学,2024.
- [11]. 杜威,宁武,孟丽因,等. 基于改进 YOLO 的矿卡驾驶员疲劳检测算法[J]. 现代电子技术,2025,48(7):126-131. DOI:10.16652/j.issn.1004-373x.2025.07.018.
- [12]. YUXUAN CHEN, RONGPENG LI, XIAOXUE YU, et al. Adaptive layer splitting for wireless large language model inference in edge computing: a model-based reinforcement learning approach[J]. Frontiers of Information Technology & Electronic Engineering,2025,26(2):278-292. DOI:10.1631/FITEE.2400468.