

“AI 算法与智能芯片”冬令营结题报告



作品名称*: _____ 基于 k230 的工地安全综合管理平台 _____

队伍成员*: _____ 吕香辰 金奕霏 张华廷 _____

团队名称*: _____ RISE.L -Recognition Intelligence Systems Edge Lab _____

目录

一、 应用场景 (Application)	1
二、 系统设计 (System Architecture)	2
三、 具体实现 (Implementation)	4
四、 效果展示 (Evaluation)	6
五、 组内分工 (Contribution)	9
六、 未来发展与提升 (Future Work)	10



一、应用场景 (Application)

I. 核心应用场景

该系统聚焦需重点关注头盔佩戴的入口通行场景适配工地入门口等关键管控区域，以“精准统计、身份溯源、数据赋能、设备可控”为核心目标，既实时捕捉人员通行状态，又实现设备在线监控与多终端统一管理，为安全管理提供全维度的数据支撑与决策依据。

II. 核心功能点

功能类别	具体核心功能点
基础硬件适配	K230 硬件资源调度、CanMV 系统适配、LCD/HDMI 显示输出、SD 卡文件读写、设备 MAC 地址获取
网络通信	WiFi 自动连接、HTTP POST 告警上报、飞书 Webhook 消息推送、心跳数据定时上报
头盔检测	基于 YOLOv8 的实时头盔检测、佩戴 / 未佩戴分类统计、检测框可视化绘制、置信度阈值自定义
人脸识别	人脸检测（关键点提取）、人脸特征提取、余弦相似度匹配、本地人脸库管理、身份信息关联
告警管理	违规告警触发、告警频率控制（防刷屏）、告警数据结构化封装、告警信息设备标识关联、离线告警缓存
设备管理	设备心跳上报（含 MAC/IP/ 运行时长 / 固件版本）、设备在线状态监控、多设备统一接入适配
可视化 UI	实时统计面板、系统日志展示、警告横幅、WiFi 状态 / 时间 / 设备标识等系统信息展示
工程化能力	异常捕获、内存回收（gc）、资源释放、调试模式支持、数据格式兼容处理、离线数据补报
数据统计分析	合规通行率统计、违规次数统计、违规人员 TOP 榜单、设备运行状态统计、可视化报表生成

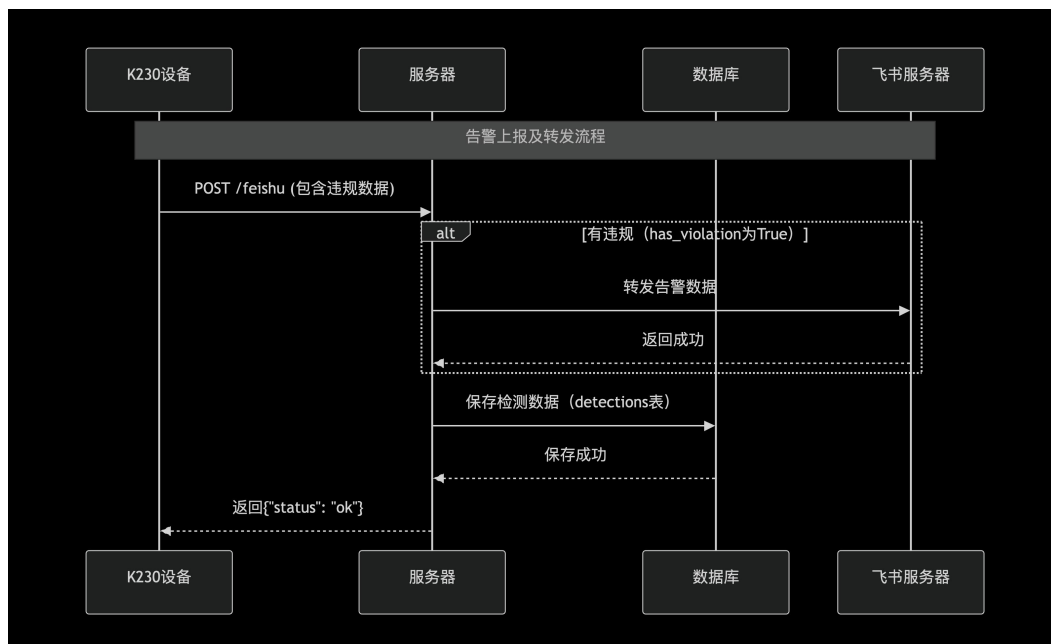
二、系统设计 (System Architecture)

1. 系统框架

该系统基于 K230 嵌入式硬件与 CanMV 开发环境构建，整体遵循 “感知层 - 处理层 - 传输层 - 应用层” 的核心逻辑，所有核心检测逻辑在 K230 本地完成，仅将告警、心跳等关键数据上传至后端，既保障实时检测性能，又兼顾设备管控与数据溯源的工程化需求。

且我们制作了 RISE.L.net - 边缘智能设备通信标准与库，它具备：低门槛，高性能，高可靠性，高扩展性，生产就绪的特点，实现由实践到抽象，后标准化最后反哺实践的转化

框架核心特点突出：一是端侧轻量化，头盔检测、人脸识别等 AI 推理本地完成，单帧检测耗时 ≤ 1 秒，适配门禁场景实时性要求；二是数据结构化，所有上传数据均封装为标准 JSON 格式，便于后端解析与多设备统一管理；系统按 “基础支撑 - 核心检测 - 设备管理 - 通信交互 - 工程保障” 维度划分为 8 大模块



II. 模块介绍

1. 网络通信模块

核心职责是实现终端与服务器的网络连接及数据传输。具备 WiFi 自动连接功能，支持 10 秒超时重试，连接成功后自动获取 IP 地址；基于 socket 封装轻量级 HTTP 客户端，支持 JSON 格式数据上传。

2. 头盔检测模块

基于 YOLOv8 模型实现头盔佩戴状态的实时检测与统计。支持置信度阈值 (0.45)、NMS 阈值 (0.5) 自定义配置，区分“佩戴头盔”与“未佩戴头盔”两类标签，统计对应人数；在显示界面绘制检测框与类别标签，直观呈现检测结果。

3. 人脸检测与识别模块

负责完成人脸关键点提取、特征提取与身份匹配，实现违规人员身份溯源。人脸检测子模块加载专用模型，提取人脸关键点坐标，通过置信度 (0.5)、NMS (0.2) 阈值过滤无效检测结果，人脸数据库管理子模块加载本地特征库，基于 0.65 的余弦相似度阈值实现身份匹配，返回匹配姓名与相似度得分。

4. 设备管理模块

实现设备运行状态的监控与上报，保障终端可管可控。按 30 秒默认间隔向服务器上报心跳数据，实时计算设备开机时长，为后端监控提供数据支撑；基于 MAC 地址的唯一标识设计，支持多台终端接入后端平台，实现检测数据、心跳信息的统一汇总管理。

三、具体实现 (Implementation)

I. 检测能力

系统支持批量人员快速通行检测，单人均检测耗时控制在 1 秒内，可高效应对上下班高峰、施工换班等人流密集场景；同时具备极高的识别精度，能精准区分“佩戴头盔”与“未佩戴头盔”个体，确保统计数据的真实性。网络中断时自动缓存违规记录与设备数据，待网络恢复后自动补传至后端，保障数据不丢失。

II. 身份关联与责任追溯

系统将人脸识别技术与头盔检测结果深度绑定，通行人员的佩戴状态会同步关联其注册的身份信息，同时通过飞书 Webhook 将告警信息精准推送至对应区域管理人员，内容清晰呈现检测时间、违规人数、身份信息，且支持配置告警间隔，避免高频重复告警干扰管理工作，让安全管理更具针对性。

III. 设备管控与规模化适配

系统自动获取设备 MAC 地址作为唯一标识，每 30 秒向后端服务器上报心跳数据，含设备 IP 地址、运行时长、在线状态等信息，管理人员可在后端实时查看所有终端的运行状态，及时发现设备离线、故障等异常情况，保障检测系统持续稳定运行。同时支持多台检测终端同时接入后端平台，所有设备的检测数据、心跳信息、告警记录统一汇总管理，适配大型工地、多门禁点的规模化应用场景，降低跨区域管理成本。

IV. 数据统计与分析

系统具备自动化的数据汇总能力，可实时统计当日、当周、当月的合规通行率（佩戴头盔通行人数 / 总通行人数）、违规次数，并生成可视化报表。这些数据不仅能直观反映当前安全防护落实情况与设备运行状态，还能帮助管理人员发现安全管理薄弱环节（如特定时段违规频发、特定设备离线率高等），推动安全管理从“被动检查”向“主动预防”转变。

V .RISE.L.net

我们基于课题总结了 RISE.L.net。RISE.L.net 是轻量级、高可扩展的边缘设备通信标准与库。该项目源于我们在智能安全监测系统，RISE.L.net 将这些复杂逻辑封装为简洁的 API，使得设备接入只需 3 行代码：

```
服务器端:
from risenet_server import quick_server
quick_server(8080, lambda device_id, data: print(f"设备 {device_id} 上报: {data}"))

客户端 (MicroPython/嵌入式设备):
from risenet import quick_start
device = quick_start("http://server:8080", "WiFi_SSID", "password")
device.report("detection", {"helmet": 1, "no_helmet": 0})
```

此库还具备完整的设备生命周期管理,可以实现自动设备注册, 周期性心跳机制, 离线超时检测, 以及高度可扩展架构, 采用分层、模块化设计, 在保持基础用法简洁的同时, 提供强大的扩展能力, 并且多平台支持。

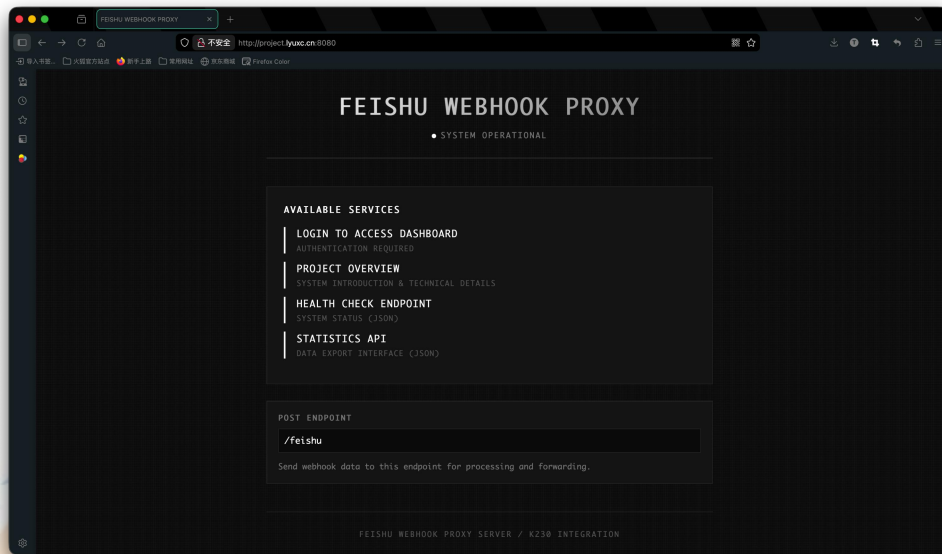
► 在我们开发的头盔检测系统中, RISE.L.net 实现了:

1. 边缘 AI 推理: K230 芯片运行 YOLOv8 进行实时头盔检测
2. 云端管理: 服务器统一管理多个检测设备, 实时收集检测结果
3. 即时告警: 检测到违规 (未戴头盔) 时立即推送飞书通知
4. 设备监控: 自动检测设备在线状态, 设备离线时及时告警
5. 数据统计: 按日统计检测数据, 生成违规趋势报告

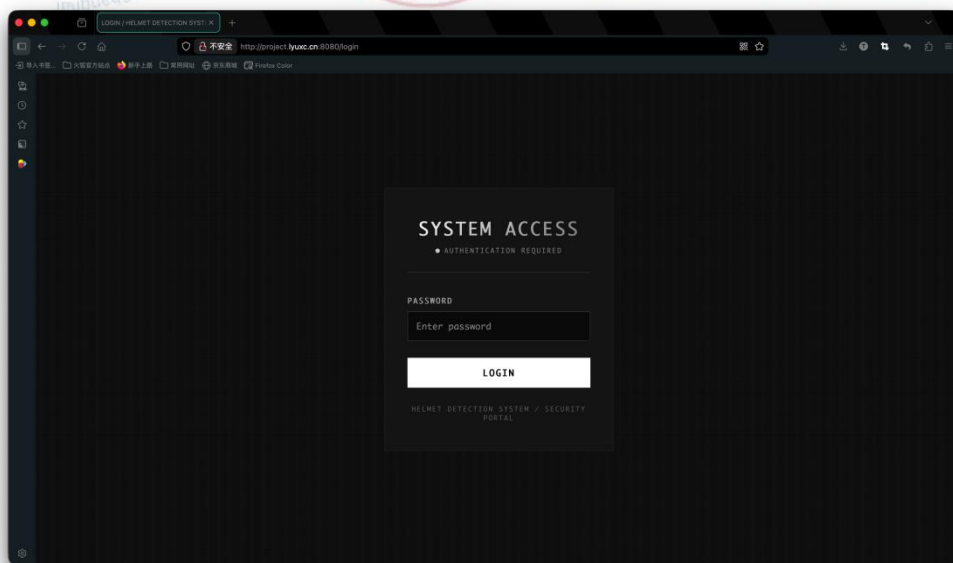
RISE.L.net 的开发降低了边缘 AI 设备接入云端的技术门槛, 使得我们和其他开发者能够专注于核心的 AI 算法和业务逻辑, 而无需在通信协议、设备管理等基础设施上花费大量时间。该项目体现了 RISE.L 课题组在“简化复杂性、提升可复用性”方面的技术理念。

通过将头盔检测系统的通信层抽象为 RISE.L.net 标准, 课题组不仅提升了自身项目的工程质量, 在实战中更形成了可对外输出的技术资产, 体现了从“做项目”到“做产品”的技术成熟度提升, 同时增加项目的可扩展性。

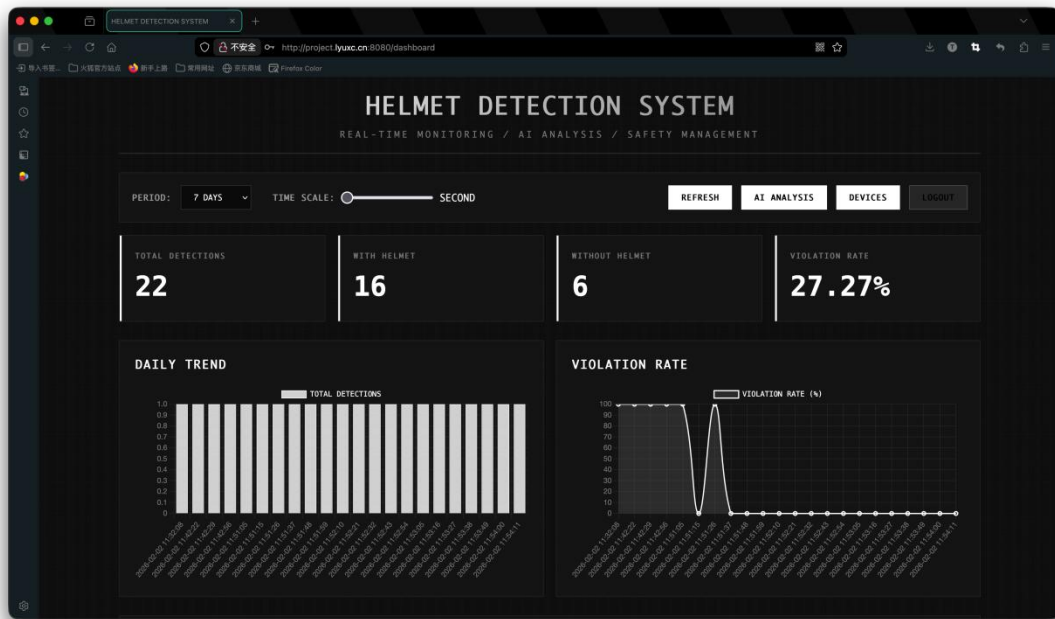
四、效果展示 (Evaluation)



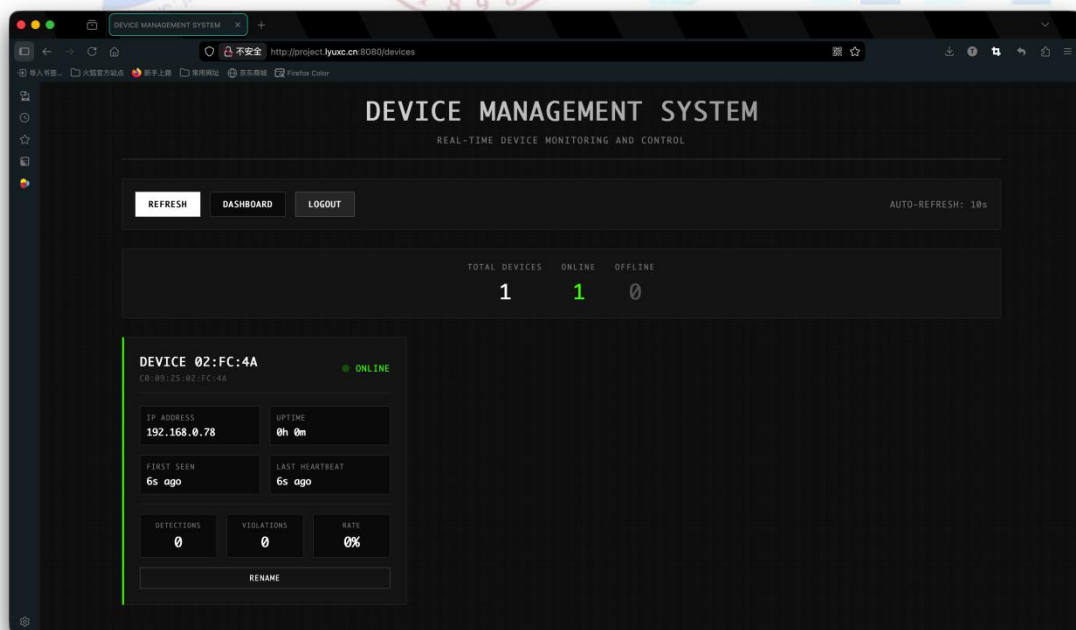
主界面 ↑



登录界面 ↑



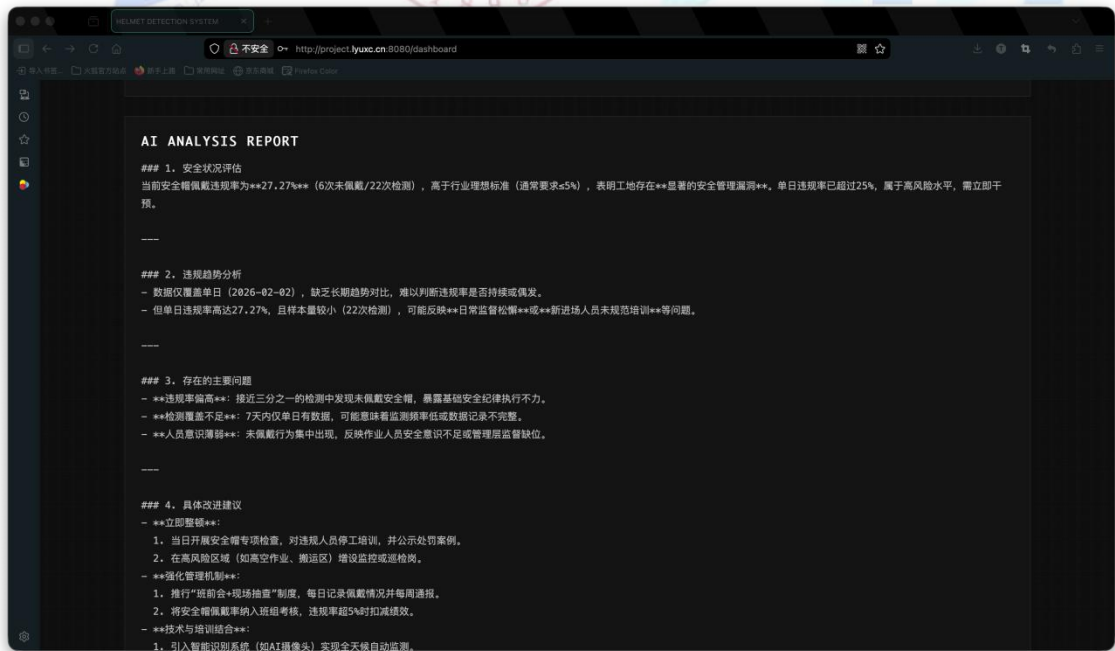
统计界面 ↑



开发板健康检测 ↑



飞书短信提醒 ↑



AI 总结 ↑

五、组内分工 (Contribution)

吕香辰 (组长) : 开发

- 修复代码明显 bug (心跳上报失败、飞书告警不触发、人脸数据库加载异常、WiFi 断连不重连等)。
- 确保头盔检测、人脸识别、心跳上报、飞书告警四大核心功能的逻辑闭环, 检测结果与告警触发精准关联。
- 简单优化 feishu_proxy.py, 确保告警消息正常转发;
- 增加关键运行日志 (检测结果、网络状态、识别得分), 方便测试岗定位问题

金奕霏: 测试

- 修改配置参数 (WiFi / 服务器地址 / 模型路径 / 检测阈值等)。调整阈值 (相似度)、告警间隔, 测试系统在边界条件下的表现, 避免误报 / 漏报、频繁告警。
- 记录测试问题, 向开发同学精准反馈
- 调通 face_registration.py 人脸注册脚本, 实现人脸特征采集、保存与数据库对接;

张华廷: 文档

- 查找并确认课题相关资料, 收集项目过程中的所有技术文档、测试数据、硬件清单等, 确认课题方向
- 准备汇报相关 PPT
- 统筹规划

六、未来发展与提升 (Future Work)

希望实现放在工地门口的安全卫士向人员入园全路径智能安防卫士的创造性转化。

I. 增强集群控制

能够结合机器狗等现有设施，由于我们可以实现集群控制，希望展现出人员进入场景的完整路径。

实现从工地入口、通道至施工 / 作业区域的全流程人员动态追踪与安全管控，完整还原人员进入场景的行进路径，让安全监测从“单点防控”升级为“全链路防护”。

II. 优化检测模型

系统现阶段仍存在部分待优化的技术痛点，头盔检测模型因训练样本与场景覆盖不足，对人员低头、侧身、转头等非正面动作的适配性较差，会出现人员已佩戴头盔却检测不到的漏检情况；且人脸识别模型精度不高，可加以改善，实际检测中会出现误检、漏检情况，可以联动集群设备协同识别依托系统集群控制能力。

系统当前搭载的人脸识别模型仍存在精度短板，实际检测中易出现误检、漏检情况，后续将从模型优化，提升身份识别的精准度与稳定性。

III. 差异化风险管理

针对未注册人员管理，目前仅能在飞书告警中提示“未佩戴头盔 + 未注册人员”，但未注册人员作为现场安全管控的高风险群体，现有管控手段不足、安全系数较低，需要改进，可以尝试制作未注册人员抓拍图像本地自动保存机制，将实时抓拍人员面部及现场画面并本地存储，为现场安全核查、违规追溯提供可视化依据。

同时可以联动机器狗，对进入高风险区域的未注册人员进行近距离抓拍、轨迹追踪，并实时推送告警至现场管理人员，实现精准管控。

IV .RISE.L.net

针对于 RISE.L.net 的展望:

- ❖ 添加 TLS/SSL 支持
- ❖ 实现命令下发功能
- ❖ 提供 Web 管理界面
- ❖ 开发 C/C++ 版本
- ❖ 支持消息队列 (Redis/RabbitMQ)
- ❖ 添加更多内置插件 (Telegram、邮件通知等)

