基于 RK3588 平台构建的边缘智能 AI 教育直播系统

摘要

随着人工智能与边缘计算技术的快速发展，AI 技术在直播教育领域展现出显著的应用价值。传统基于电脑录屏的直播教育模式存在三大核心痛点：其一，缺乏真实的师生互动场景；其二，不具备实时字幕辅助功能；其三，缺少基于大模型的智能课堂摘要能力。针对上述问题，我们基于 RK3588 芯片设计了一套创新的 AI 教育直播系统。具有以下核心功能：1. 实时交互模块：构建沉浸式教学场景，支持师生实时视频互动。2. 智能字幕模块：通过语音识别技术自动生成实时授课字幕。3. 小瑞助手模块：集成大语言模型的智能助教，支持自然语言交互。4. 智能跟拍模块：教师端配备动态目标跟踪算法自动调整拍摄视角，确保移动中的教师始终处于最佳画面位置。本系统通过边缘计算与 AI 技术的深度融合，有效解决了传统教育直播中互动性差、辅助功能缺失、拍摄体验不佳等关键问题，为远程教育提供了更智能、更人性化的解决方案。本系统依托 Rockchip RK3588 的强大异构计算能力（集成 6TOPS NPU 和 8 核 CPU），构建了一体化的 AI 推理 + 实时传输 + 人机交互平台。

第一部分　作品概述

1.1 功能与特性

(1) UI 可视化界面和人机交互

我们设计了嵌入式客户端和 Web 客户端。嵌入式客户端可以通过按键控制直播的开始和结束，通过 UI 显示视频。Web 客户端则分成学生端和教师端。学生端可以看到教师直播讲课和实时字幕，可以发布弹幕。教师端可以看到个人在视频中的位置，并让摄像头对自己进行跟踪。

(2) 目标检测与跟踪

开启直播后，会对摄像头视野中的所有人进行目标检测。可以点击 UI 界面（两种客户端均可）的检测框进行单目标跟踪。

(3) 语音识别

开启直播后，对教师说的话进行语音识别，并通过网络传输到 Web 学生端，成为实时字幕。

(4) 大模型问答

按住 UI 界面的小瑞按键，然后输入语音，直到松开按键为止。大模型会接收到语音识别后的文本，并把回答后的结果反馈到 Web 客户端中。

(5) 视频实时推流

流媒体服务器会接收到 RK3588 的原始视频数据和目标检测推理数据。Web 学生端和教师端分别能够拉取服务器的两种视频数据。

(6) 远程控制与通信

RK3588 将目标跟踪的中心坐标和显示中心坐标计算出偏移角度，通过串口传输给下位机。下位机处理角度换算成占空比后控制云台舵机，使得人可以保持在显示中心。

1.2 应用领域

远程教学与互动展示

教师可通过语音控制摄像头视角或目标追踪，实现智能跟拍教学。学生端通过浏览器实时观看，并能通过语音提问系统，或者对于教学的内容摘要，提升教

学互动性。

公共场景导览与讲解

系统可部署于博物馆、展览馆等场所，自动识别参观者并进行语音讲解。观众提问可实时识别并由大模型生成自然语言回答，构建智能化自助导览体验。

1.3 主要技术特点

本系统基于 RK3588 平台和裁剪后的 Buildroot 操作系统，融合多模态 AI 技术与边缘计算能力，具备智能感知、语音交互、远程推流与控制等多种功能，具有以下主要技术特点：

(1) 高性能边缘推理能力

集成图像（目标检测与跟踪）、语音（语音识别）、语言（大语言模型）三大 AI 模块，实现“看得见、听得懂、能理解”的智能交互体系。

YOLOv5s[1, 2] 是 YOLOv5 系列中最轻量化的模型版本，具有体积小、推理快、部署灵活等特点。其网络结构经过优化，能在边缘设备上实现高速的目标检测，适用于低功耗、高实时性的场景。Whisper[3] 是由 OpenAI 开源的端到端语音识别模型，具备强大的多语种识别能力和高鲁棒性，尤其适应嘈杂环境。Whisper 可离线运行，无需依赖云端服务，保护用户隐私，非常适合部署在边缘端进行本地语音识别。DeepSeek[4] 是新一代多模态大语言模型，擅长自然语言理解、问答生成、文本摘要等任务。配合语音识别模块，它可实现人机对话和语音指令理解。在边缘端，可通过模型量化或本地微调部署，适合有限资源下运行简单语义处理任务，为移动终端赋予基本的“思考”与“理解”能力。

(2) 实时视频推流与前端交互

采用 WebRTC 技术实现超低延迟音视频推流，支持浏览器端实时访问与远程互动；配合双通道图像输出（原始图像 + 检测图像），满足多样化的展示与分析需求，提升远程可视化体验。

WebRTC[5] 是一套支持浏览器实时音视频通信的开源技术，具有低延迟、高兼容、点对点传输等特点。它不依赖传统服务器中转，在边缘端可实现低成本的实时视频直播或远程监控，特别适用于公网访问困难或带宽受限的移动应用场景。

(3) 模块化 Qt 架构设计

系统采用 Qt/C++ 架构设计，模块间高度解耦，推理、采集、推流等任务通过多线程并发执行，保障系统在高负载下的稳定性与实时性，同时方便功能拓展与平台迁移。

(4) MQTT 远程控制和串口通信

Web 客户端通过轻量级 MQTT 协议与 RK3588 进行控制，RK3588 通过串口 UART 与下位机进行通信。保证了低延迟、高可靠的特性。

MQTT[6] 是一种轻量级的消息传输协议，专为低带宽、高延迟、不稳定网络设计。它通信开销小、实现简单，非常适合边缘设备间的控制指令传输。

1.4 主要性能指标

指标类别	性能指标
原始帧显示	帧率 ≥ 30 FPS(1280×720)
目标检测	帧率 ≥ 15 FPS(640×640)
语音识别	识别时间≤ 500 ms
LLM 回答	结束时间≤ 2s
原始帧推流延迟	≤ 100 ms
推理帧推流延迟	≤ 500 ms

1.5 主要创新点

(1) 多模态 AI 模型集成于本地边缘系统

将视觉（YOLOv5s）、听觉（Whisper 语音识别）、语言理解（DeepSeek 大模型）三类 AI 模型融合部署于 RK3588 边缘平台。

(2) 高并发多线程任务调度架构

采用 Qt 多线程异步架构，解耦摄像头采集、AI 推理、音频识别、推流与控制任务，保障在边缘平台上多任务并发运行下的稳定性和流畅性，有效释放 RK3588 的计算资源，实现各类任务的动态负载调度与扩展。

(3) 应用于远程教育直播

传统基于电脑录屏的直播教育模式存在三大核心痛点：其一，缺乏真实的师生互动场景；其二，不具备实时字幕辅助功能；其三，缺少基于大模型的智能课

堂摘要能力。针对上述问题，针对上述问题，我们基于 RK3588 芯片设计了一套创新的 AI 教育直播系统，具有以下核心功能：1) 实时交互模块：构建沉浸式教学场景，支持师生实时视频互动。2) 智能字幕模块：通过语音识别技术自动生成实时授课字幕。3) 小瑞助手模块：集成大语言模型的智能助教，支持自然语言交互。4) 智能跟拍模块：教师端配备动态目标跟踪算法自动调整拍摄视角，确保移动中的教师始终处于最佳画面位置。

1.6 设计流程

主要分为：需求分析、架构规划、模块设计、模块优化、模块测试。

(1) 需求分析：调研边缘设备对于各领域的作用

(2) 架构规划：设计多层次的系统架构

(3) 模块设计：设计多个 AI 模型，实现多个功能

(4) 模块优化：合理分配硬件资源，解决资源竞争问题

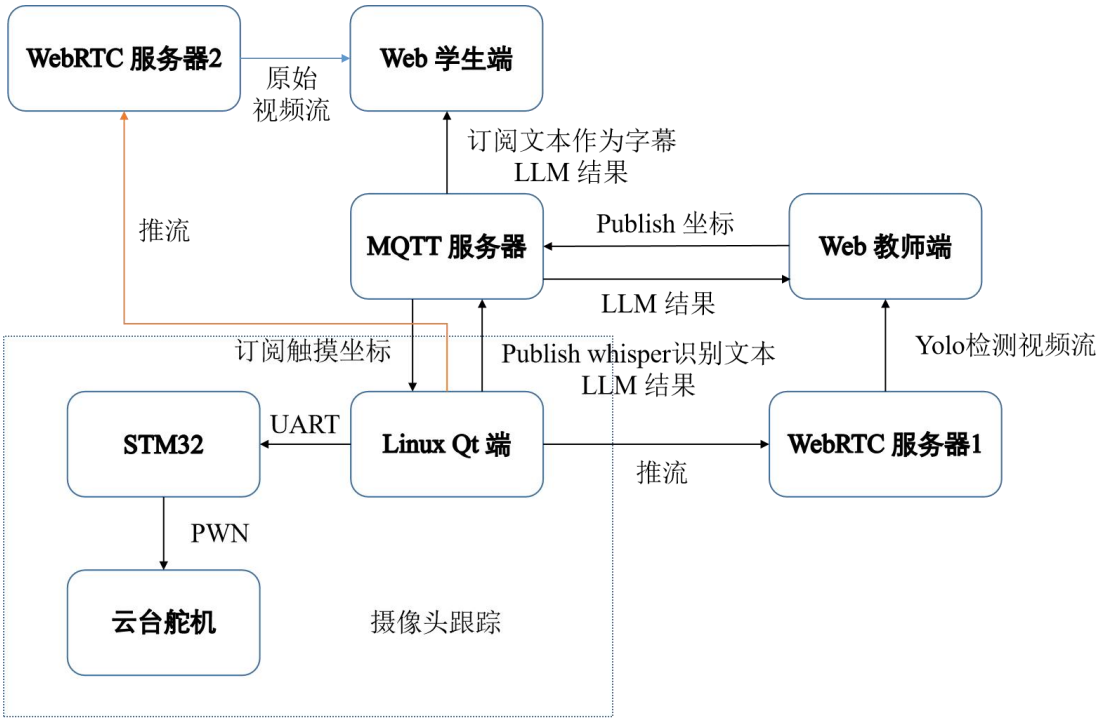
(5) 模块测试：测试性能，排查问题

第二部分　系统组成及功能说明

2.1 整体介绍

给出系统整体框图，各子模块标注清楚，并进行整体的文字说明，需要表达出各模块之间的关系。

(1) 整体框架



(2) 说明

我们的系统主要分为 Web 端、Linux Qt 端和设备端。其中 Linux Qt 端为核心，实现 AI 模型的推理，视频和音频数据的处理，并通过网络和串口分别于 Web 端和设备端进行通信。

2.2 硬件系统介绍

2.2.1 硬件整体介绍；

(1) RK3588

项目	参数
SoC	Rockchip RK3588
CPU	8 核 ARM 处理器（4×Cortex-A76 + 4×Cortex-A55），主频高达 2.4GHz
GPU	ARM Mali-G610
NPU	6 TOPS（INT8）独立神经网络处理器，支持 RKNN 模型加速
RAM	4GB DDR4
存储	eMMC
视频编解码	支持 8K H.265/H.264 解码、4K 编码，适合高分辨率直播

音频支持	支持 I2S、SPDIF、ALSA 接口，适配麦克风与音频采集模块
系统	Linux Buildroot/Ubuntu

(2) 主要外设

摄像头	OV13588
网络通信	AX200
触摸屏	HDMI 7 寸屏
下位机	STM32F103
云台舵机	SG90

2.3 软件系统介绍

2.3.1 软件整体介绍（含 PC 端或云端，结合关键图片）；

(1) Linux 端

Linux 端采用 Buildroot 操作系统，Qt/C++进行开发。大致可以分为：视频模块、语音模块、网络和串口模块以及主窗口模块。Qt 设计 6 个控件如下图所示：1) 开始，用来开启摄像头；2) 结束，关闭摄像头；3) 小瑞，使用大模型功能；4) 取消跟踪；5) 退出；6) 视频显示区域。

(2) Web 端

该系统的客户端为 web 端程序，web 端使用 Vue.js 进行开发，充分发挥其组件化、响应式数据绑定和虚拟 DOM 的特性，以实现高效、灵活且易于维护的用户界面。在页面设计上，运用 Vue.js 的单文件组件（.vue）模式，将 HTML、CSS 和 JavaScript 封装在同一文件中，便于代码的组织与复用。为了实现 web 端整体风格的统一性，引入 TailwindCSS 进行页面样式的设计，并自行封装页面使用的组件，如按钮，输入框等，能在各个页面复用并且具有可扩展性。同时自行封装组件能够减少冗余的依赖，带来更多自定义的空间。

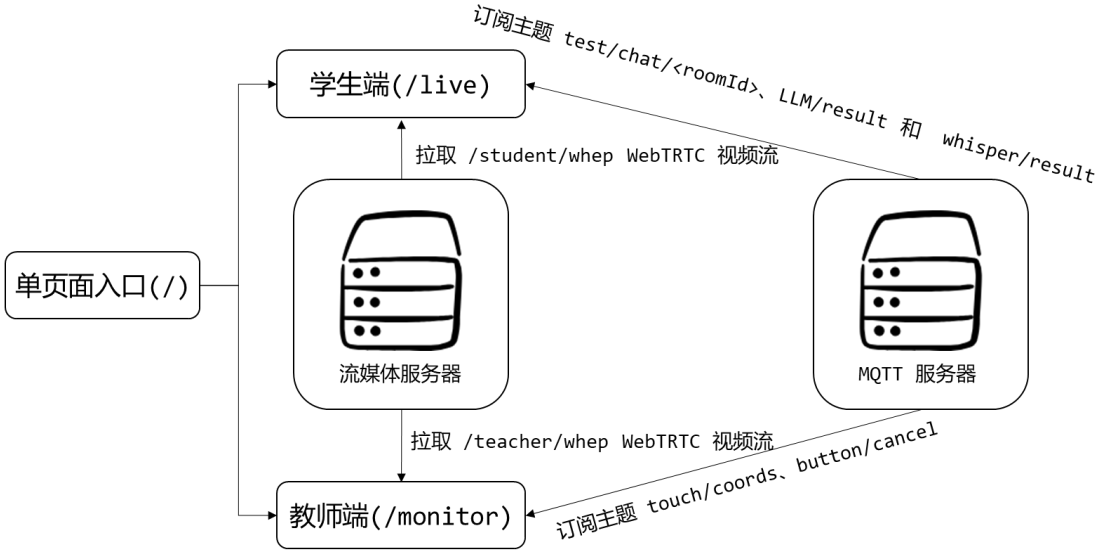
客户端目前共有两个主要页面，分别为学生端和教师端页面，其中，学生端接收摄像头的推流并展示摄像头所拍摄的视频：

教师端接收板端目标识别的画面，并且点击目标识别框选区域，板端能够介绍消息并对选中的目标进行持续追踪，在监控端也能够进行追踪功能的取消：

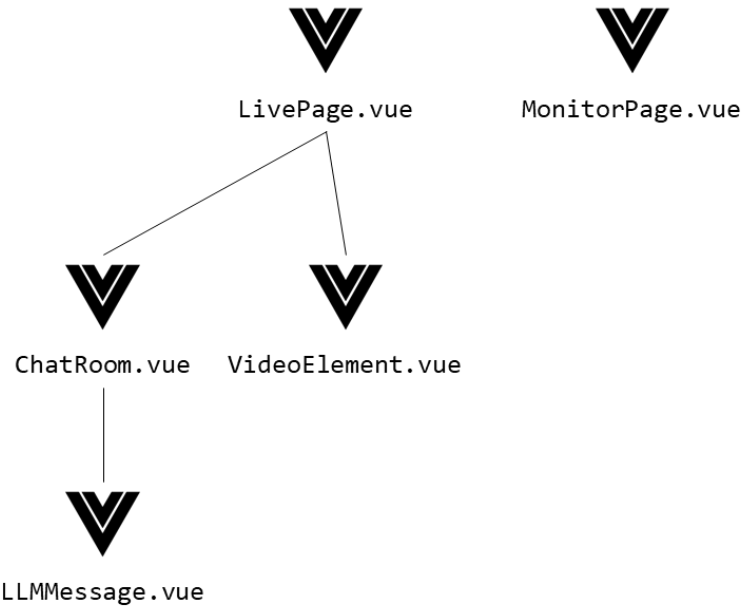
2.3.2 软件各模块介绍（根据总体框图，给出各模块的具体设计说明。从顶层到底层逐次给出各函数的流程图及其关键输入、输出变量）；

(1) 前端设计

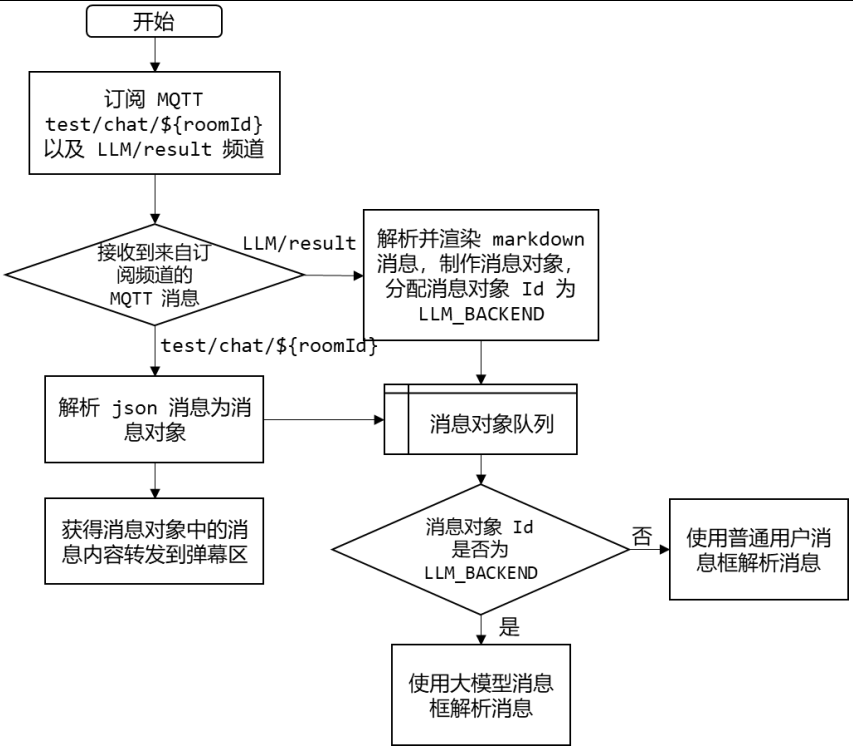
主要页面视频拉流以及消息订阅设计：



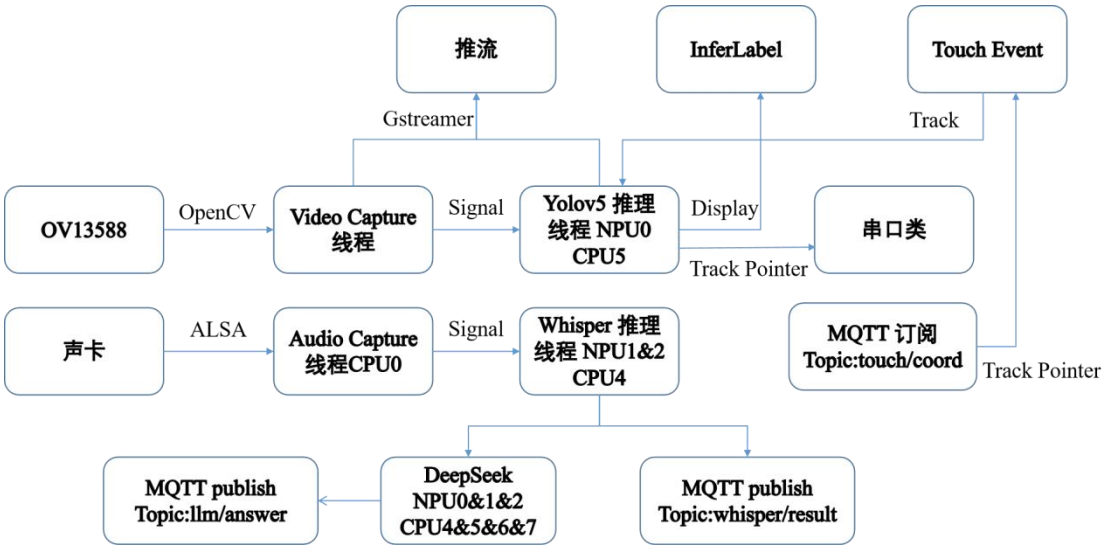
主要页面以及组件间关系：



消息列表以及弹幕渲染逻辑：



(2) RK3588 设计



① 主 UI 模块：Widget

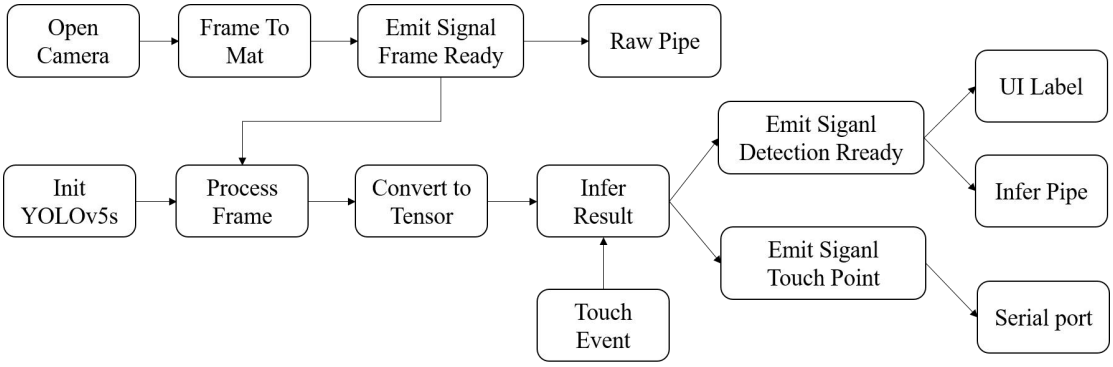
关键函数：

函数名	说明
initThreads()	初始化推理线程
updateInferResult(const QImage &result)	显示推理视频和推流推理帧
initConnections()	初始化各种信号的连接

② 视频采集和目标检测模块

功能：

- 使用 OpenCV/V4L2 采集摄像头图像
- 对摄像头数据进行目标检测
- 接收 UI 的触摸事件，进行目标追踪



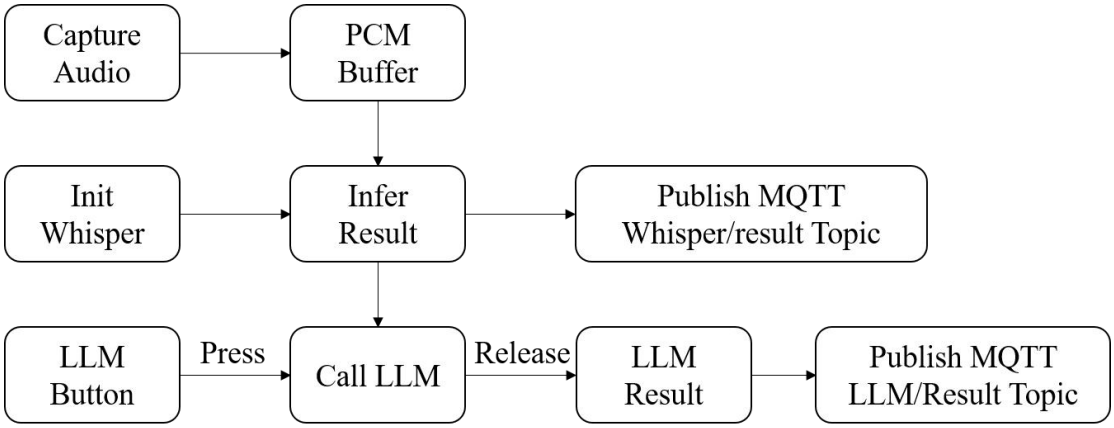
输入变量：摄像头设备路径和 YOLOv5s 模型路径

输出变量：输出带有检测框的推理帧

③ 语音识别和 LLM 模块

功能：

- 使用 ALSA 获取音频存入缓冲区
- 将缓冲区的音频进行语音识别，并将结果上传相应主题
- 将语言识别后的文本，进行 LLM 处理，并将结果上传相应主题

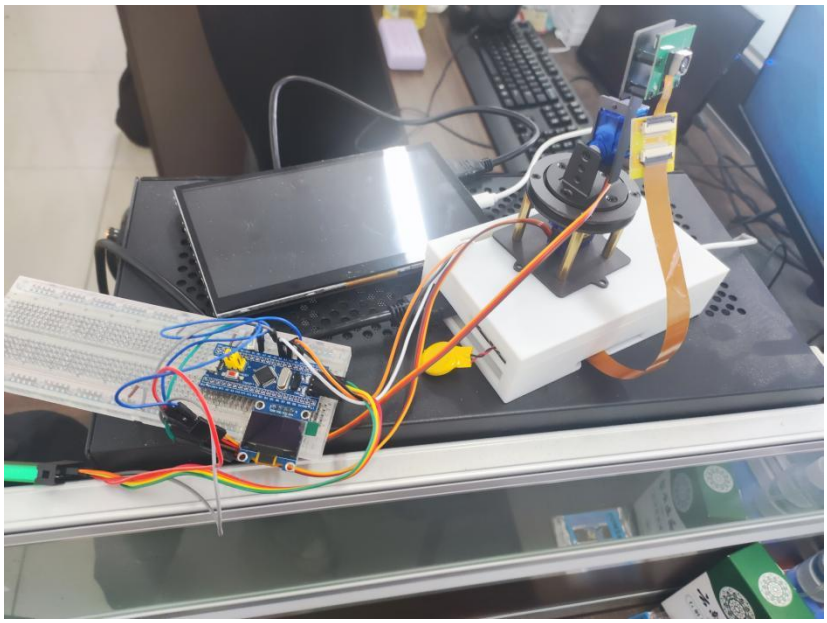


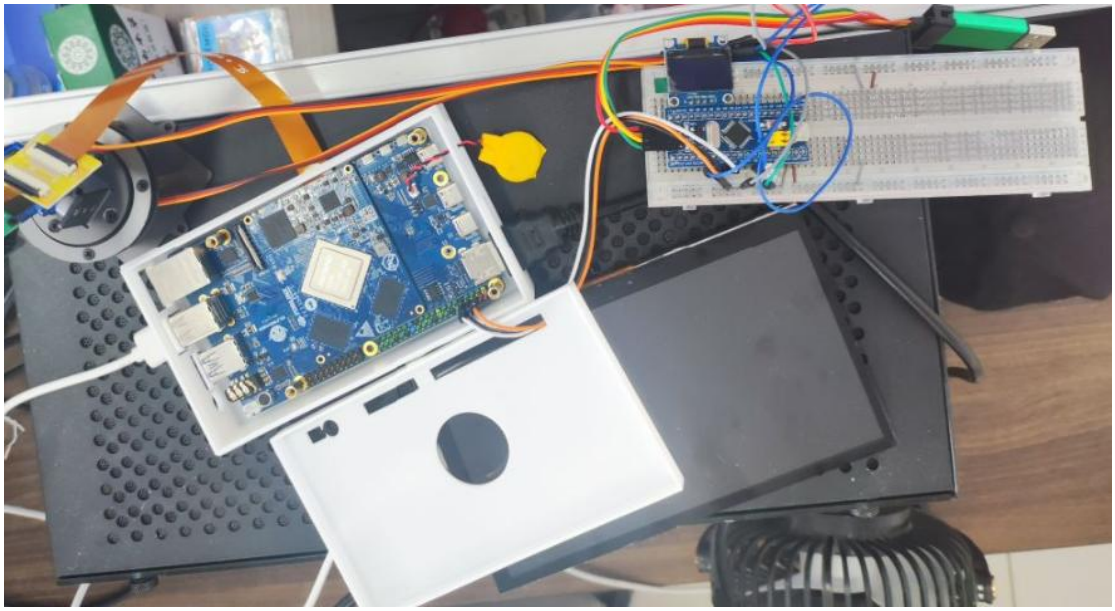
输入变量：音频采集设备和 Whisper 模型路径

输出变量：输出中文文本、LLM 输出结果

第三部分　完成情况及性能参数

3.1　整体介绍（整个系统实物的正面、斜 45°全局性照片）



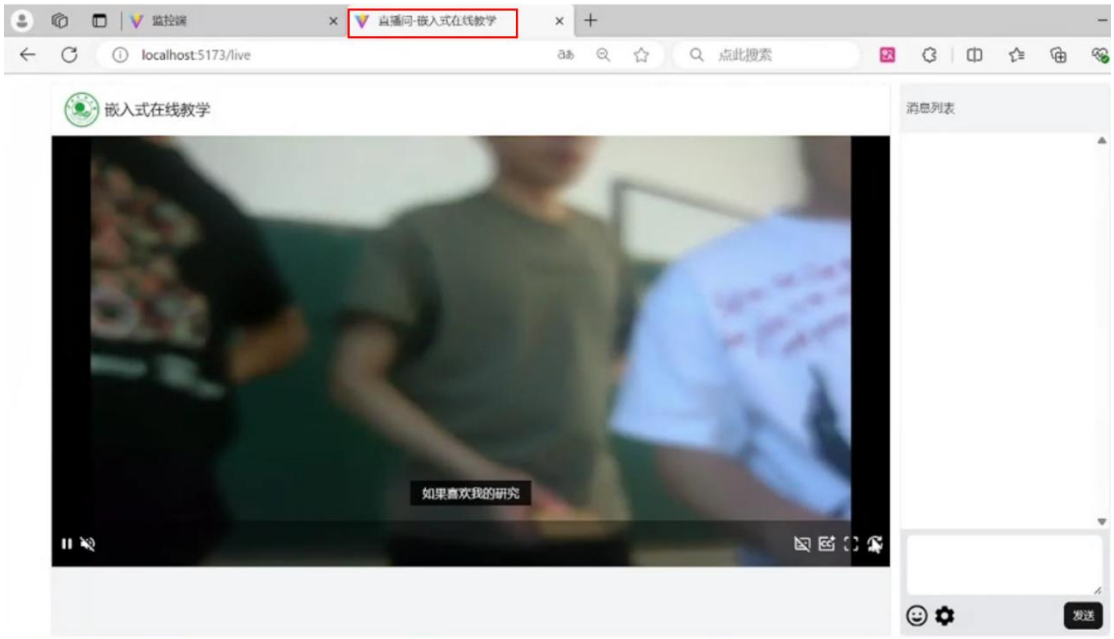


3.2　工程成果（分硬件实物、软件界面等设计结果）

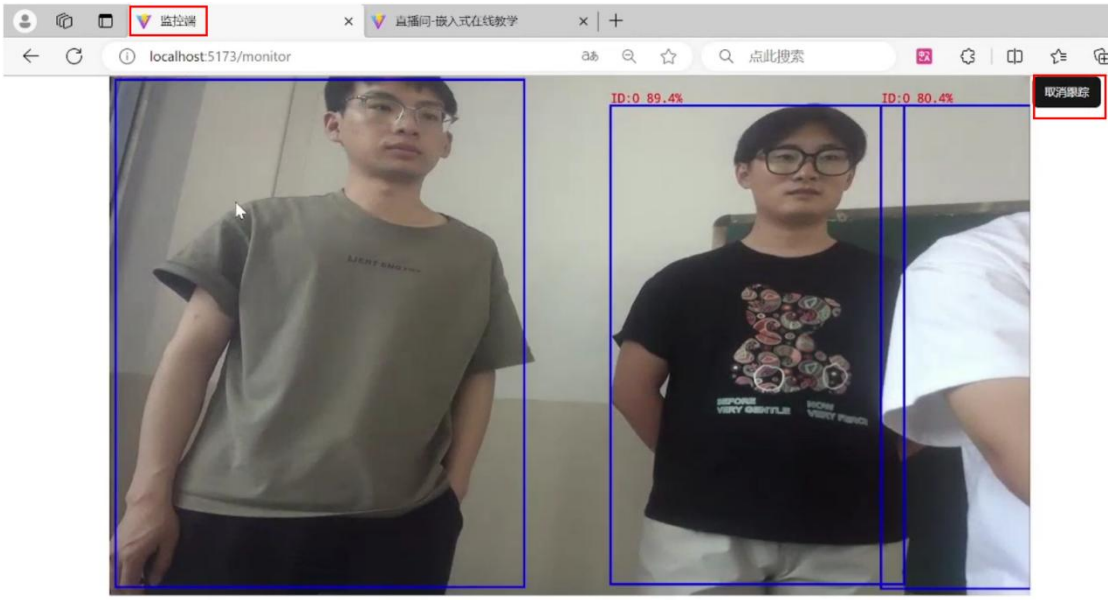
3.2.3　软件成果；

(1) Qt UI　界面

(2) Web 学生端



(3) Web 教师端

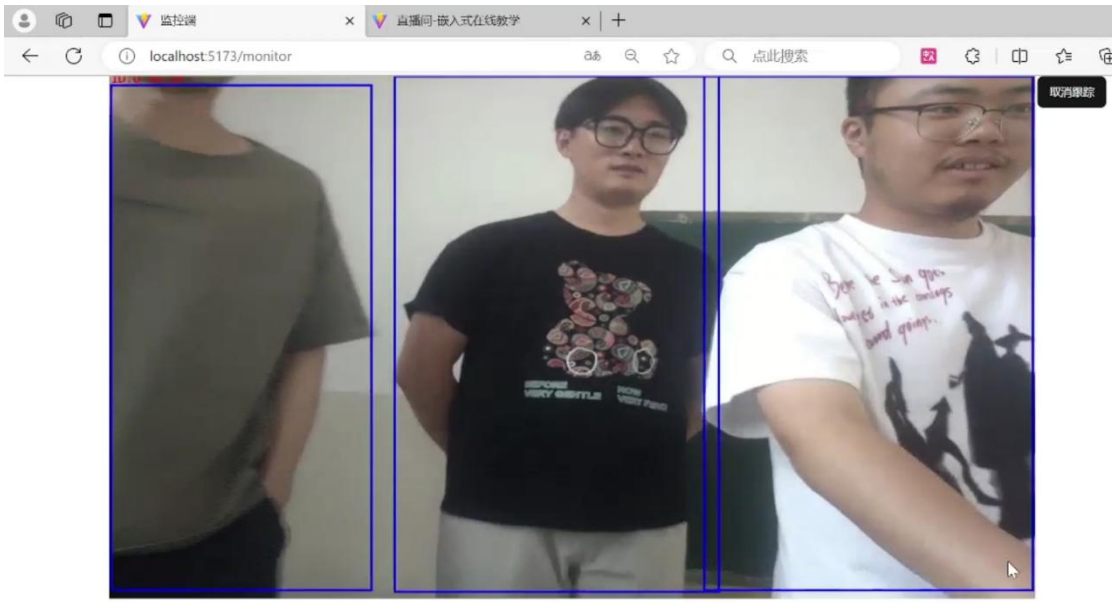


3.3 特性成果（逐个展示功能、性能参数等量化指标）（可加重要仪器测试或现场照片）；

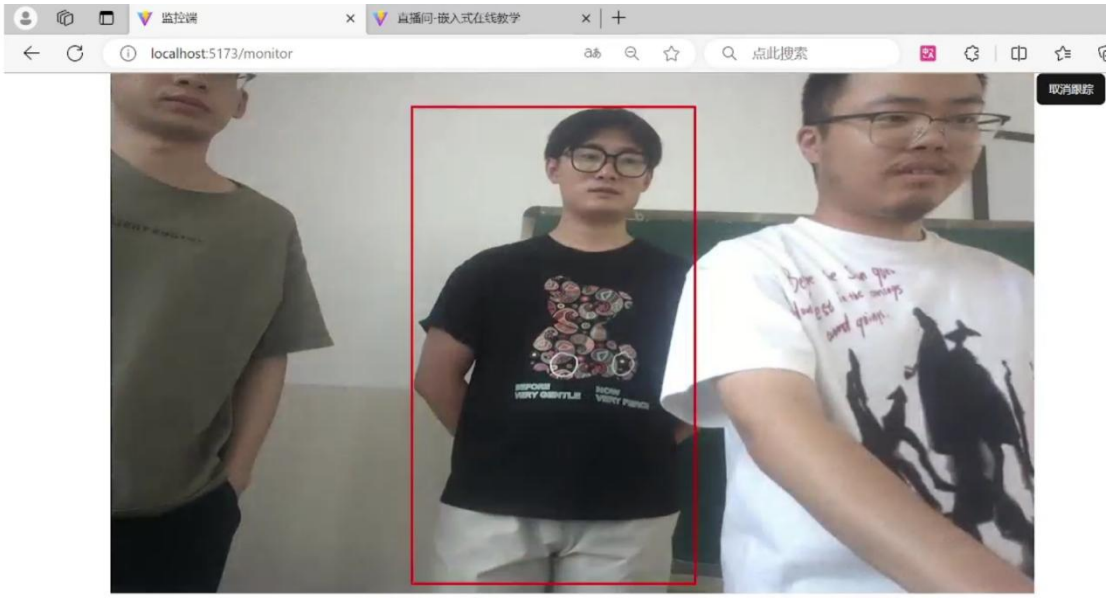
(1) Yolov5s 目标检测

① Qt 端目标检测

② Qt 端目标追踪



③ Web 教师端目标检测



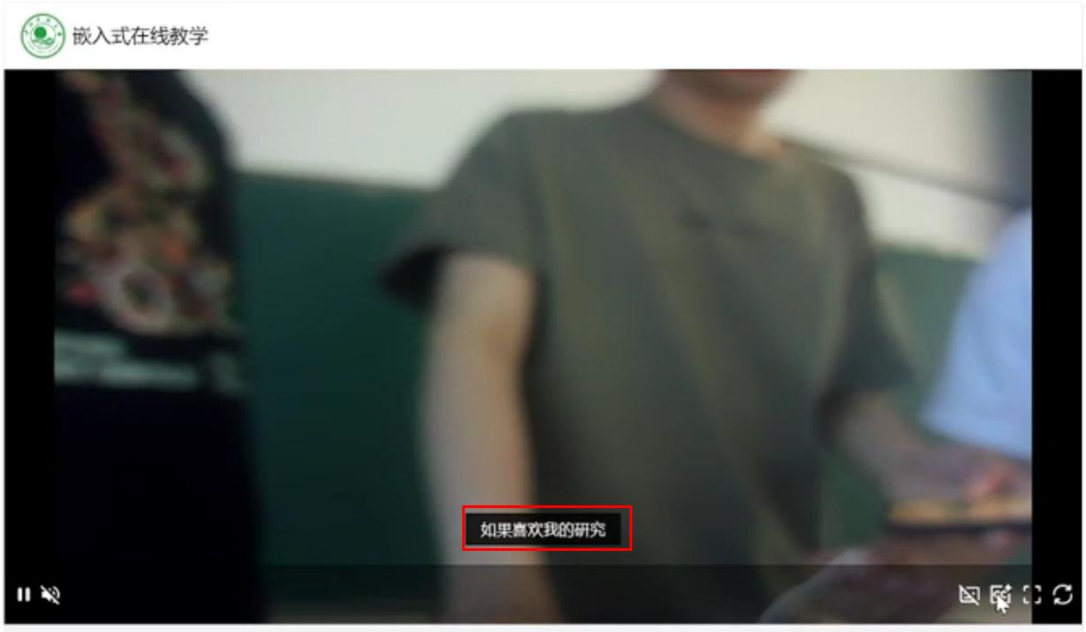
④ Web 教师端目标跟踪

帧率：

[2025-07-06 14:57:08.196] 开始转录，输入帧数: 14.7638

[2025-07-06 14:57:09.259] 开始转录，输入帧数: 15.0659

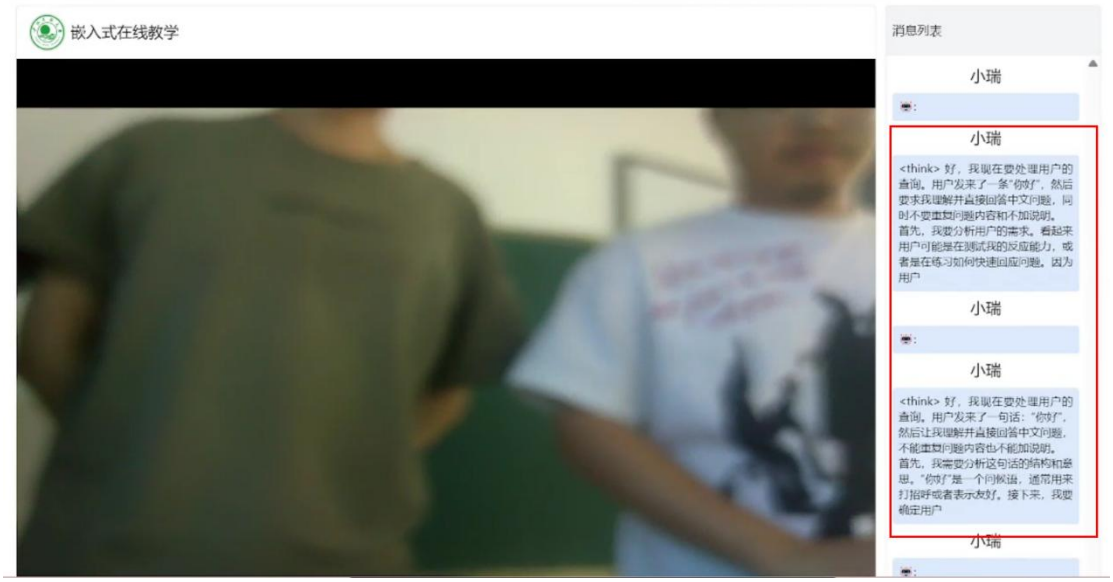
(2) Whisper 语音识别



[2025-07-06 14:57:05.632] 开始转录，输入帧数: 32000

[2025-07-06 14:57:06.044] 推理耗时: 397 ms

(3) DeepSeek 回答



第四部分　总结

4.1　可扩展之处

(1) YOLOv5 还可以设计成线程池，并使用 NPU 三核进行推流。FPS 能够提升到 30 帧甚至 60 帧。

(2) 还可以增加 TTS 文本转语音，让大模型能够“说话”。

4.2　心得体会

研发过程中，我选择使用 PC 端 Ubuntu 环境和 RK3588 Buildroot 环境。学会了 CMake 构建工程，学会了 Qt 集成开发。在模型转换过程中，遇到了一些困难，会遇到精度不够，模型结构不符合等问题。在模型部署的过程中，三个模型对于 4G 的内存可谓是十分紧张。开发过程中，经常遇到程序卡顿情况，需要考虑资源竞争、异步运行、线程调度等问题。当然段错误可谓是最头疼的，起初使用 QDebug 进行排查，后期使用 GDB 进行远程调试。在与前端和下位机进行对接的时候，也有一堆问题。例如前端通信的网络卡顿。在测试阶段中，也是一堆 BUG 需要处理，例如连续点击 Qt 的 Widget 就会段错误，跟踪的过程中不能再点跟踪框否则也会段错误，清理线程的时候有些线程不能简单的用线程清理。

第五部分 参考文献

[1] JOCHER G, STOKEN A, BOROVEC J, et al. ultralytics/yolov5: v3. 0 [J]. 2020.

[2] IMANI H, HOSEN M I, FERYAD V, et al. Efficient object detection model for edge devices; proceedings of the International Conference on Advanced Engineering, Technology and Applications, F, 2023 [C]. Springer.

[3] RADFORD A, KIM J W, XU T, et al. Robust speech recognition via large-scale weak supervision; proceedings of the International conference on machine learning, F, 2023 [C]. PMLR.

[4] LU H, LIU W, ZHANG B, et al. Deepseek-vl: towards real-world vision-language understanding [J]. 2024.

[5] BLUM N, LACHAPELLE S, ALVESTRAND H J C O T A. Webrtc: Real-time communication for the open web platform [J]. 2021, 64(8): 50-4.

[6] NAIK N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP; proceedings of the 2017 IEEE international systems engineering symposium (ISSE), F, 2017 [C]. IEEE.