

UAgentOS-XiUOS：端侧视觉检测与现实反馈系统技术报告

队伍名称：UAgentOS

参赛方向：基于 XiUOS 开发板的 AI 视觉目标检测与状态判别

1. 系统总体成果

本项目完成了一套可运行的视觉监护原型。XiUOS/K210 开发板持续采集摄像头画面，在板端运行 YOLOv2/KPU 完成目标检测，并把检测框实时显示在 LCD 上。当目标检测成立时，系统保留该时刻的触发帧并上报检测事件；UAgentOS 接收事件与图片，生成自然语言提醒，再通过 QQ 发送图文消息，同时请求蓝牙音箱进行现场播报。

因此，用户能够同时获得两种反馈：离开现场时在 QQ 查看触发图片和解释，身处现场时直接听到语音提醒。没有检测到目标的画面只用于 LCD 实时预览，不进入通知链路。



这一原型完整覆盖赛题要求的三段流程：**视觉数据采集与预处理、轻量模型部署与推理、结果反馈与系统联动。**

2. 赛题任务与完成内容

赛题任务	赛题关注点	本项目完成内容	可核验材料
任务一：视觉数据采集与预处理	摄像头接入、分辨率配置、裁剪、灰度化、缩放	通过 XiUOS 摄像头设备接口持续获取 256×256 画面；LCD 实时显示；触发帧完成目标区域裁剪与灰度转换	XiUOS 工程、LCD 实拍、运行说明
任务二：轻量 AI 模型部署与推理	资源受限平台上的目标或状态判别	将 YOLOv2 KModel 嵌入固件，使用 K210 KPU 推理，输出类别、目标数量和检测框	模型配置、固件、出框演示
任务三：结果反馈与系统联动	屏幕、网络或其他方式输出，并触发后续行为	LCD 本地画框；目标事件与对应帧通过网络上报；UAgentOS 生成解释并触发 QQ 图文与蓝牙音箱播报	事件日志、QQ 截图、播报演示

3. 系统结构与职责边界

系统由三部分构成：

- 视觉开发板：**负责摄像头取流、图像预处理、KPU 推理、检测框与本地显示，并在目标成立时发布事实和对应图片。
- UAgentOS：**运行在 Gemini-S1 嵌入式开发板上，负责接收视觉事实、组织事件上下文、生成自然语言并选择反馈方式。Agent 本身同样位于端侧，而不是只运行在桌面电脑中。
- 用户反馈端：**QQ 提供可追溯的图文通知，蓝牙音箱提供现场声音提醒，两类反馈分别返回执行状态。

这里的“节点”指一台承担明确职责的真实设备；“能力”指该设备内部可被调用的功能。

视觉检测是视觉开发板内部的能力，图文通知和声音播报是反馈设备提供的能力。

UAgentOS 只请求能力，不把不同设备的具体命令混入视觉主循环。

4. 任务一：摄像头采集与预处理

4.1 摄像头驱动接入

应用通过 XiUOS 摄像头设备接口打开设备、配置输出分辨率、设置帧缓冲地址，并等待采集完成标志。传感器输出和模型输入均使用 256×256 尺寸，使摄像头缓冲能够直接服务 KPU

推理，减少资源受限平台上的重复分配与拷贝。

4.2 同一画面的两条用途

- **实时显示路径：** RGB565 画面持续写入 LCD；无论是否检测到目标，用户都能看到摄像头当前视野。
- **推理与证据路径：** 模型使用其输入布局完成推理；目标成立后，系统从对应画面生成触发帧，并与检测状态和帧号共同上报。

4.3 基础预处理

项目完成了输入尺寸配置、目标区域裁剪、缩放采样以及 RGB565 到灰度的像素转换。触发帧来源于目标成立的同一现实画面，而不是事后寻找邻近图片。



5. 任务二：YOLOv2/KPU 轻量推理

5.1 模型部署

项目使用 K210 片上 KPU 加速 YOLOv2。模型以 KModel 形式嵌入 XiUOS 固件，启动时完成长度校验与加载。

模型项	实际配置
输入尺寸	256 × 256
输出尺寸	8 × 8 × 35
检测类别	head、helmet
KModel 大小	约 2.71 MB
推理加速	K210 KPU
后处理	Region Layer 解码与 NMS

5.2 状态判别与本地可视化

每帧推理后，Region Layer 解码候选框，非极大值抑制去除重叠结果。应用根据类别、目标数和置信结果形成检测状态，并把目标框绘制到 LCD。模型输出因此同时具备“板端可见结果”和“可供系统联动使用的结构化事实”。

5.3 资源适配价值

目标检测首先在 K210 上完成，系统不依赖把连续视频上传到上位机或云端后再推理。网络只承载任务相关的事件和关键图片，体现 XiUOS 作为低成本边缘感知节点的价值。

6. 任务三：结果反馈与系统联动

6.1 一次目标事件如何抵达用户

1. 摄像头画面进入 YOLOv2/KPU，目标成立并在 LCD 出框。
2. 系统记录目标状态、帧号、目标数量和该时刻的触发图片。
3. 网络线程把事件与对应图片发送给 UAgentOS。
4. UAgentOS 生成自然语言提醒，并分别请求 QQ 图文通知和蓝牙音箱播报。
5. 两个反馈通道返回已送达、已排队或失败原因，结果重新进入事件记录。

6.2 显示与发送解耦

摄像头、KPU 与 LCD 构成持续实时路径；网络线程只处理已经形成的任务事件。没有目标的普通画面不会排队发送。这一职责分离避免把慢速通信放进每帧显示流程，也让故障定位能够区分“摄像头是否工作”“模型是否出框”和“事件是否送达”。

6.3 嵌入式 Agent 实现

UAgentOS 的 Agent 运行在 Gemini-S1 嵌入式开发板上，而不是仅存在于桌面脚本。XiUOS/K210 在本地完成 YOLOv2 目标检测；需要进一步理解图片语义时，UAgentOS 可调多模态模型服务。视觉开发板只发布事实，自然语言解释、反馈能力选择和执行状态确认由 Agent 节点完成。这样既保持 XiUOS 视觉路径清晰，也让后续增加其他现实设备时无需改写检测模型。

7. 从比赛原型到现实设备 Agent 范式

本项目进一步抽象出一套面向真实设备的 Agent Harness。核心观点是：**端侧设备是节点，能力内生于节点，协调层负责治理能力，Agent 核心可以替换。**

- **节点是设备容器。** 一块开发板、一个音箱或一个通知服务首先作为节点存在，保存身份、连接、位置、状态和能力集合。
- **能力属于节点内部。** 摄像采集、目标检测、温度读取、通知和播报不是与节点并列的硬件，而是对应节点能够声明和执行的能力。
- **能力能够自动注册。** 节点接入后发布能力描述、参数和结果契约；Harness 完成发现、注册、健康检查与编目，使 Agent 不必硬编码每个品牌命令。
- **调度必须回到现实结果。** Agent 提出能力需求，Harness 根据节点位置、状态和可用性选择执行者；观测与动作结果再回流 Agent 上下文。
- **运行时可以替换。** 比赛原型中的 UAgentOS 部署在 Gemini-S1 上；相同设备语义也可适配自研 UAgentOS 核心或 OpenVela Agent。面向 OpenClaw 等运行时的适配属于后续开发方向，而非本阶段已完成结果。



这使 XiUOS 视觉板的价值不止是“运行一次目标检测”，而是作为可自动发现、可调度、可验证的现实感知节点进入更大的 Agent 系统。

8. 实验结果与证据

已形成的可核验结果包括：

- 摄像头持续取流并在 LCD 实时显示；
- YOLOv2/KPU 在目标进入视野时输出类别、目标数和检测框；
- 目标成立时生成对应触发帧和结构化检测事件；
- UAgentOS 接收视觉事件并生成自然语言提醒；
- QQ 接收文字与触发图片；
- 音频能力返回播报请求的执行状态；
- 构建脚本、烧录说明、运行说明和事件日志可用于复现。

9. 当前边界

当前原型已打通从板端检测到用户反馈的完整路径，但仍存在工程优化空间：触发帧清晰度、局域网传输时延、特定蓝牙音箱的原生协议兼容以及长期无人值守稳定性仍需继续完善。材料严格区分“能力请求已排队”和“物理设备已经完成动作”，不把中间状态描述为最终成功。

10. 评审维度映射

评审维度	本项目对应证据
技术完整性 30%	摄像头驱动、预处理、KPU 推理、LCD、触发帧、网络事件、Agent 解释和双通道反馈组成完整闭环
开源合规性 30%	基于 XiUOS 开源工程开发；提交源代码、构建与运行说明；凭据和个人数据不进入公开仓库
场景适配性 20%	端侧持续感知、任务事件上传、现场与远程同时提醒，适合低成本空间监护场景
创新性 20%	将 XiUOS 视觉结果从局部显示扩展为可解释、可反馈、可验证的现实事件

11. 复现顺序与交付物

复现顺序为：构建并烧录 XiUOS 视觉固件；启动 UAgentOS 接收与反馈服务；让设备进入同一局域网；确认 LCD 正常取流；让目标进入摄像头视野；依次核对检测框、触发帧、事件

记录、QQ 图文和音频状态。

提交材料包括完整工程代码、技术报告、设计文档、说明文档、网页 PPT、构建与烧录脚本。网络密码、机器人密钥和个人参考图片通过本地配置注入，不进入公开提交。

结论： 本项目完成了 XiUOS 赛题要求的采集、推理与联动闭环，并让一次端侧视觉检测真正转化为用户能够看见、听见并核验的现实反馈。