会议纪要

会议时间	2025 年 6 月 29 日 21 时—23 时 30 分	会议方式	线上会议
会议内容	主题确立、项目设计及任务分工		
主持人	林泓宇（组长）	记录人	林泓宇
参会人员	方琳涵、王芮祺、刘华杰、李明桐、陈梓渊、杜冠辰		

会议记录

一、主题确立

① 项目定位与名称

定位：提升人类视觉识别效率的深度学习方法：多场景图像识别系统

名称：慧眸科技，“慧”代表智慧与智能—AI 大模型，“眸”代表视觉与识别—YOLO 视觉模型。

② 设计愿景

我们的愿景是成为智能视觉分析领域的领先者，通过将 AI 技术与实际应用场景相结合，为社会安全、公共服务提供实时、可信赖的决策支持，实现“智慧视觉助力生活，连接技术与未来”的目标。

③ 问题陈述

各行业对自动化视觉监控的需求日益增长，但现有解决方案普遍面临以下挑战：

1)技术瓶颈：传统图像识别系统在面对复杂、多变（如弱光、遮挡、小目标）的真实世界场景时，识别精度和效率均显不足。

2）信息孤岛：绝大多数系统止步于“识别”，即仅仅输出“是什么”的标签（如“检测到火焰”），缺乏对情景的深度理解和可行动的洞察，无法回答“该怎么办”。

3）安全隐患：随着系统功能日趋复杂，用户数据安全、系统访问控制及防御网络攻击（如 SQL 注入）等安全与隐私问题日益凸显，成为阻碍我们在关键领域应用的因素。

4）场景局限：许多系统是为单一任务设计的，扩展性差，难以快速适应新的识别需求和场景。

④ 解决方案

为解决上述问题，我们提出并设计“慧眸科技”平台。它是一个端到端的解决方案，核心构成如下：

1）高效识别：采用最新的 YOLOv11 深度学习模型作为核心识别引擎，确保在烟雾、火焰、人员落水、交通违规等多种场景下实现高精度和高效率的目标检测。

2）智能分析：创新性地集成了 AI 大语言模型。系统不仅能识别异常，更能结合上下文，自动生成风险分析、提供解决方案建议，

并支持用户通过多轮对话进行深度探索，实现从“识别”到“洞察”的质的飞跃。

3）安全保障： 系统从设计之初就贯彻安全理念。通过用户角色与权限管理、密码加密、JWT 认证、Prompt 安全过滤和参数化查询防御 SQL 注入等多层防护机制，确保平台及用户数据的安全。

4）开放扩展： 采用前后端分离和微服务思想，支持 Web 端、小程序等多端接入，并能通过训练新的数据集，快速扩展至更多、更复杂的识别场景。

二、项目设计

项目由“前端应用层、后端服务层、AI 能力层和数据持久层”四大核心部分组成，是一个基于 HTTPS 协议的客户端与负责处理逻辑的服务端构成的 C/S 架构。

图一 ： 项目设计图

① 前端应用层

· 技术栈：采用当下主流的 Vue.js 前端框架。

· 职责： 提供用户交互界面，包括用户注册登录、图像/视频的上传

与展示、识别结果的可视化、与 AI 分析模块的对话交互、以及历史记录的查看与管理。支持 Web 端和移动端（小程序）两种形态。

② 后端服务层

·技术栈： 采用 Python 编程语言及 Flask 后端框架。

·职责： 作为系统的业务逻辑中枢。负责处理前端 API 请求、管理用户认证与权限、调度 AI 模型进行识别、与数据库和对象存储进行交互、并集成第三方服务（如 AI 大模型 API、内容安全服务）。

③ AI 能力层

·视觉识别核心： 封装 YOLOv11 模型，基于 PyTorch 框架进行多场景的视觉模型训练，以提供高精度的实时目标检测能力。此外服务器通过 RTX3060 显卡以确保后端保持高推理速度。

·智能分析核心： 通过 API 调用外部的大语言模型 (LLM)，实现对识别结果的深度分析与对话式交互。

④ 数据持久层

·数据库：我们选用 MySQL 作为核心关系型数据库，负责对用户信息、识别历史、对话记录等关键结构化数据进行持久化存储。为实现应用逻辑与数据库的有效解耦并确保代码的健壮性与安全性，数据库交互均通过 Python 生态中的 SQLAIchemy—ORM 框架(与基于 Java 生态的 Hibernate 类似，均使用 ORM 框架设计而成)执行。该框架通

过对象化的方式抽象了底层 SQL 操作，不仅提升了我们的开发效率，更从根本上通过参数化的查询方式，有效防御了 SQL 注入等潜在的安全威胁。

三、软件环境

① 前端应用层

前端应用层构建于现代 Web 技术栈之上，采用 Vue.3 作为核心框架，并借助 Vite 作为高效的构建工具。所有与后端的通信均通过标准的 HTTP 客户端完成。最终用户可在任何主流现代浏览器或微信小程序环境中无缝访问我们的服务，我们以此保证了卓越的跨平台兼容性和一致的用户体验。

② 后端服务层

后端服务层是整个系统的业务逻辑核心，我们选择 Python 作为主开发语言，并通过轻量级的 Flask 框架进行构建。为实现用户认证，我们还会采用 Flask-Bcrypt 进行密码安全哈希。

③ AI 能力层

视觉识别任务由 PyTorch 驱动的 YOLOv11 模型在支持 CUDA 的 NVIDIA GPU 上执行，以保障实时高效的分析能力；智能对话则通过 API 与外部大语言模型集成。

④ 数据持久层

在数据存储方面，我们采用 MySQL 作为关系型数据库来管理所有结构化数据，并通过 SQLAlchemy ORM 框架进行安全、高效的对象化操作，防止外部的 SQL 注入等有害攻击。

四、任务分工

① 数据库的设计与建立及 DDoS 攻击的防御

明确系统中存在的实体（如用户、识别任务、AI 对话），以及它们之间的关系（如一对多），并通过设计 SQL 语句，以确保组员能够在本地复现该数据库，使得项目能够在此基础上进一步设计。此外，需根据“用户 id+总攻击次数（delta 范围内）”表项确定是否存在 DDoS 攻击现象，如有则需要封禁该账号数天不等。

② 前端界面设计及相关功能的实现

负责构建与用户直接交互的界面和基础的用户认证流程，通过以下步骤完成该项任务，分别是：前端界面的设计、数据库的本地复现及最后前端-后端-数据库的三方连接。其中前端界面设计包含：首页、项目简介、产品市场、关于我们及识别结果界面与 AI 分析界面六种。

③ 用户历史（多轮及上下文对话方式）

负责实现用户与 AI 大模型之间连贯、有上下文记忆的对话功能，并提供历史记录追溯，通过以下步骤完成该项任务：将 AI 大模型在

后端接入、并通过打包上下文对话，以确保应用具备上下文记忆功能。并确保当前用户能够访问其历史对话记录，并可以进行再次对话。

④ Prompt 及等安全性设计与后端插入、SQL 注入测试

是保障 AI 交互内容安全的关键环节，我们设计了双重安全保障以屏蔽涉黄、暴力、政治敏感等信息输入大模型，具体如下：输入大模型前的敏感关键词触发机制；及输入大模型后的 Prompt 模板以让其再次复查图片和文字是否存在危险内容的自主审查机制。

⑤ 多场景视觉模型的训练及外部大模型 API 的接入

我们团队选择了十种日常所需的 AI 识别应用场景，分别是：烟雾识别、火焰识别、人员落水识别、光伏巡检识别、河道漂浮物识别、农田作物生长情况识别、行人闯红灯识别、车辆闯红灯识别、人员摔倒识别、景区客流量识别，在 Roboflow/ Kaggle 平台收集好数据集后，通过主流的 YOLOv11 视觉模型进行训练，将最终得到的.pt 文件用于后续用户上传图片后对图片进行初步的识别。

在初步识别后，我们还计划设计一具备多轮对话能力、上下文交互能力的交互界面，所以此处则涉及到大模型的 API 的调用方式，需要单独查阅官网 API Document 以确保大模型能够正常接收并相应。

学号	姓名	组内分工
2022631024	林泓宇	任务一
2022611018	方琳涵	任务二
2022611012	王芮祺	
2022611003	刘华杰	
2022331103	李明桐	任务三
2022611026	陈梓渊	任务四
2022611013	杜冠辰	
-	由全体同学完成	任务五

图二 ： 任务分工情况

五、表项确立（考虑数据库的完备性——共计 15 张）

·用户表（×4）：

用户名 — 用户 id

用户 id — 密码

用户 id — 触发关键词次数（防止涉黄、暴力等信息）

用户 id — 总攻击次数（有效）= Σ(Δt 时间内的总会话次数)

·管理员表（×2）：

管理员名-管理员 id

管理员 id-密码

·被封禁表（×1）：

被封禁用户 id

·权限表（×1）：

用户/管理员 id-权限 0-1

被封禁用户 id — NULL

·类别、上传图片、识别图片、发送内容、返回结果表（×6）：

会话 id = 当前上传的图片 id

用户 id — 会话 id

会话 id — 检测类别

会话 id — 识别结果

会话 id — 轮次 id — 发送内容（用户）

会话 id — 轮次 id — 返回结果（AI）

会话 id — 轮次 id — 是否触发关键词 KEY 0 / 1

用户 id — 触发关键词次数=上传图片 id 对应的所有轮次 id 对应的 KEY 求和） 阈值 = 10

·关键词表—发送内容（×1）：

词汇-后端-涉黄、暴力、政治等敏感词汇表