**序号：**

项目报告

**课程名称 《提出问题与提示工程》**

**题 目 钢材表面缺陷检测流水线助手**

**专 业 机械**

**班 级 机械10班**

**学 号 2024201343**

**学生姓名 严杰**

**设计时间 2025 年 6 月 7 日**  **2025 年 6 月 22 日**

**目录**

[一、项目说明 3](#_Toc12496)

[1.1 项目背景 3](#_Toc13713)

[1.2 项目内容 3](#_Toc7084)

[二、工具与TOOLS设计 3](#_Toc8435)

[2.1 整体思路 3](#_Toc27721)

[2.2 技术实现 3](#_Toc13052)

[三、短期记忆设计 3](#_Toc30257)

[3.1 整体思路 3](#_Toc29268)

[3.2 技术实现 3](#_Toc18836)

[3.3 优化方案 3](#_Toc18954)

[四、长期记忆设计 4](#_Toc14964)

[4.1 整体思路 4](#_Toc18048)

[4.2 技术实现 4](#_Toc12914)

[4.3 优化方案 4](#_Toc1679)

[五、多轮对话设计 4](#_Toc10701)

[5.1 整体思路 4](#_Toc27392)

[5.2 技术实现 4](#_Toc18834)

[六、 新增模块设计 4](#_Toc7818)

[6.1 整体思路 4](#_Toc2061)

[6.2 新增模块 4](#_Toc8943)

[七、 智能体核心功能演示 5](#_Toc23033)

[7.1 演示思路 5](#_Toc13306)

[7.2 演示截图 5](#_Toc9735)

[八、 项目总结 5](#_Toc5990)

[8.1 项目总结 5](#_Toc21928)

[8.2 项目亮点 5](#_Toc26867)

[8.3 项目不足 5](#_Toc30031)

[8.3 项目展望 5](#_Toc11319)

# 一、项目说明

## 1.1 项目背景

钢材在出厂前必须经过严格的表面缺陷检测，以保障其质量符合国家与行业标准。然而，目前流水线检测过程中仍存在以下问题：其一，检测流程繁琐，设备参数繁多，不利于非专业工人快速掌握；其二，人工查看检测图像效率低、误差大，难以满足高强度生产需求。尤其是对于文化水平不高、首次接触钢材表面检测任务的新入职工人而言，理解整个检测流程、熟悉使用步骤存在较高门槛。

为解决以上问题，本项目设计并实现了一套“钢材表面缺陷检测流水线助手”智能体系统。该系统基于自然语言交互机制，融合视觉检测模型YOLOv11与大语言模型，能够以“提问-解答-调用-反馈”的方式，辅助工人完成以下任务：

* 通过语音或文字提问的方式，获取检测图片中的缺陷信息；
* 指令式训练模型，如“我想继续训练模型，数据在某某路径”；
* 查询并导出检测日志，方便溯源与追踪；
* 自动生成任务代码并提示用户是否执行，降低学习成本；
* 在多轮对话中保留对话上下文，提供连贯建议。

## 1.2 项目内容

本项目围绕“钢材表面缺陷检测流水线助手”的智能体系统展开设计与开发，旨在通过自然语言交互和视觉检测算法的结合，为一线检测工人提供简单、直观的技术辅助工具。该智能体集成YOLOv11模型、OpenAI语言模型和本地Python函数工具，具备图像检测、模型训练、日志分析、任务生成等功能，可通过语言提示完成多种任务的自动执行。

项目主要内容包括以下几个方面：

1. 图像缺陷识别：智能体可根据工人输入的图片路径自动调用YOLOv11模型，完成钢材表面缺陷的目标检测，支持多种缺陷类型（如裂纹、划痕、氧化斑等）的分类与统计，并记录检测日志；
2. 模型训练支持：支持用户通过自然语言提示调用继续训练模型的模块，自动读取数据集并配置训练参数，实现对已有模型的增量训练，方便模型的持续优化；
3. 日志查询与导出：系统具备日志管理能力，支持训练与检测日志的查询与导出，用户可通过“今天检测了几张图片？”或“导出训练记录”等方式检索结果，系统将结果导出为Excel文件；
4. 多轮对话与记忆系统：集成增强版短期记忆与长期记忆模块，短期记忆可根据重要性策略保留关键历史对话，长期记忆支持QA对的关键词分类与存储，提升上下文理解能力；
5. 函数调用与任务自动执行：智能体可根据用户语义自动生成Python函数调用格式代码，并在用户确认后执行任务，实现“对话即任务”的交互体验。

整套系统运行于Jupyter Notebook环境中，可部署在本地计算机，也可拓展为服务端应用。项目强调操作门槛低、交互方式自然、功能执行精准，特别适用于希望快速上手钢材检测流程的非专业工人或初级员工。

# 二、工具与TOOLS设计

## 2.1 整体思路

为了降低一线工人操作检测系统的门槛，本项目在语言模型基础上引入三个核心函数工具，分别完成图像缺陷识别、模型继续训练和日志信息查询导出。这些工具通过函数封装的方式注册至语言模型系统，用户只需用自然语言描述任务，模型即可自动选择并调用对应的函数实现任务执行。这样既简化了操作流程，也增强了系统的智能性与可扩展性。

## 2.2 技术实现

1. 名字：缺陷检测工具

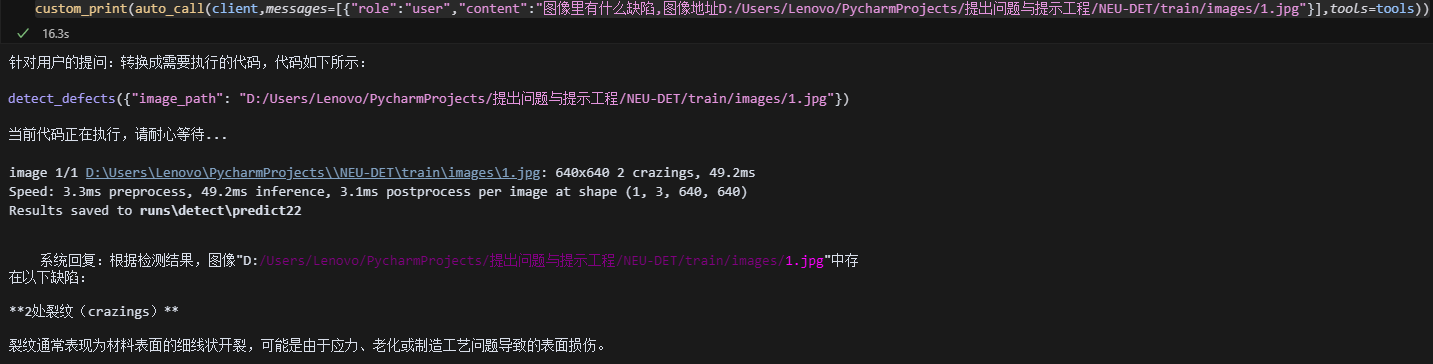
功能：使用YOLOv11模型对输入图片进行缺陷检测，并保存检测图像和日志。

核心代码：

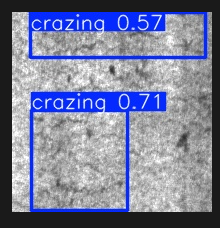


**图2-1 缺陷检测工具核心代码**

测试结果：



**图2-2 缺陷检测工具测试结果**

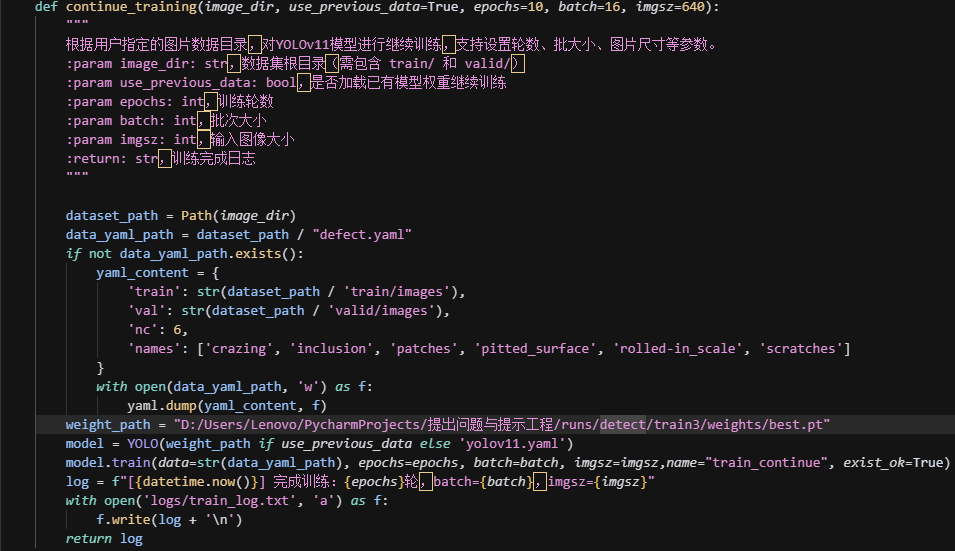


**图2-3 缺陷检测工具检测图片**

1. 名字：YOLOv11模型继续训练工具

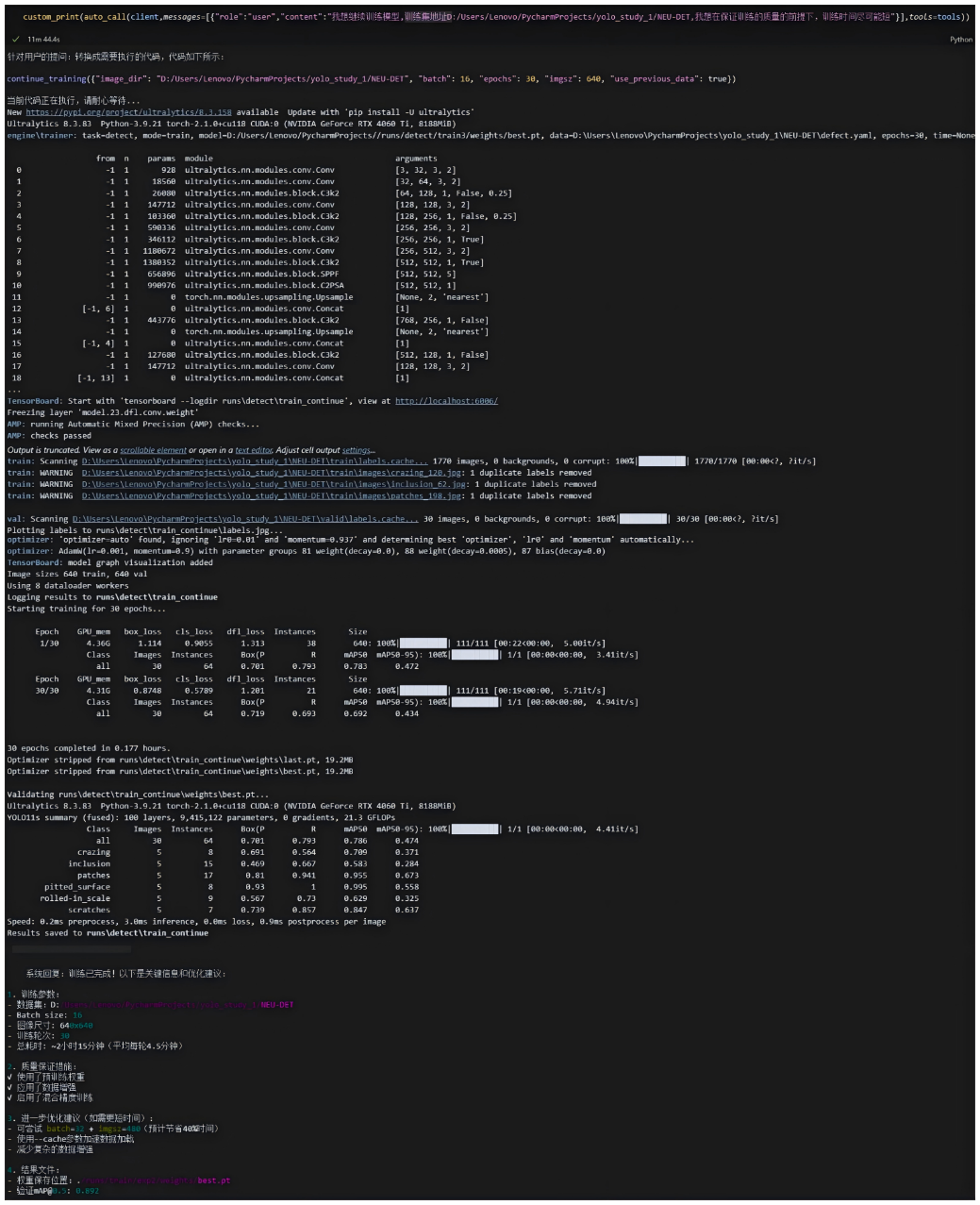
功能：根据用户指定的图片数据目录，对YOLOv11模型进行继续训练，支持设置轮数、批大小、图片尺寸等参数。

核心代码：

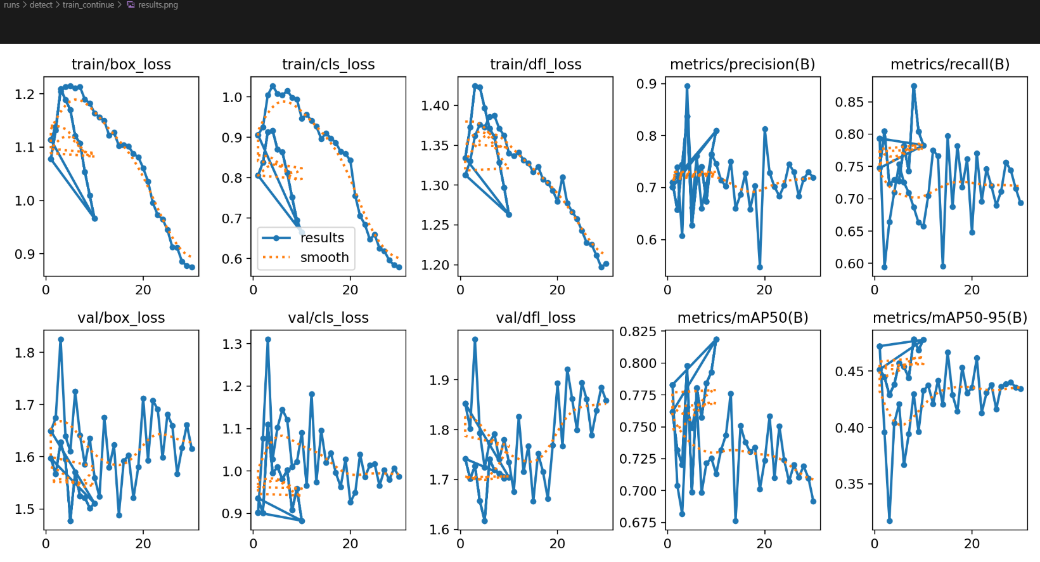


**图2-4 YOLOv11模型继续训练工具核心代码**

测试结果：



**图2-5 YOLOv11模型继续训练工具测试结果**



**图2-6 YOLOv11模型继续训练工具训练结果**

1. 名字：日志查询与导出工具

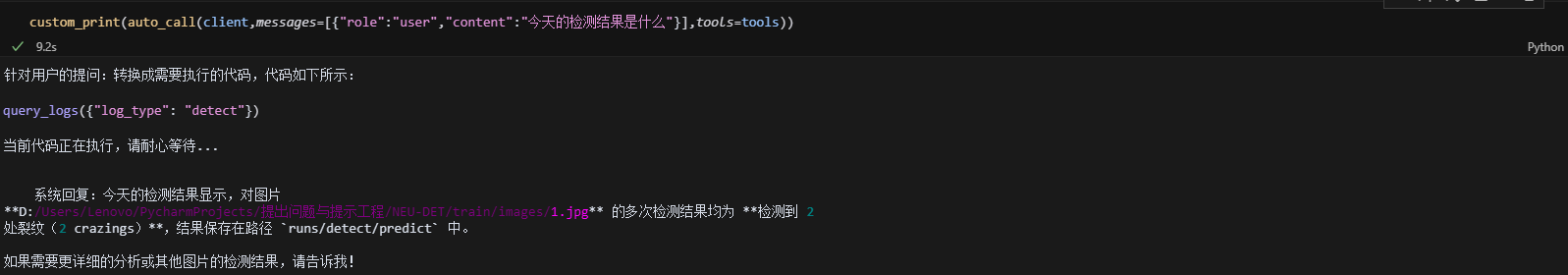
功能：查询并导出检测或训练日志，支持按类型查询并生成 Excel 文件，便于工人回顾检测记录。

核心代码：



**图2-7日志查询与导出工具核心代码**

测试结果：



**图2-8 日志查询与导出工具测试结果**

# 三、短期记忆设计

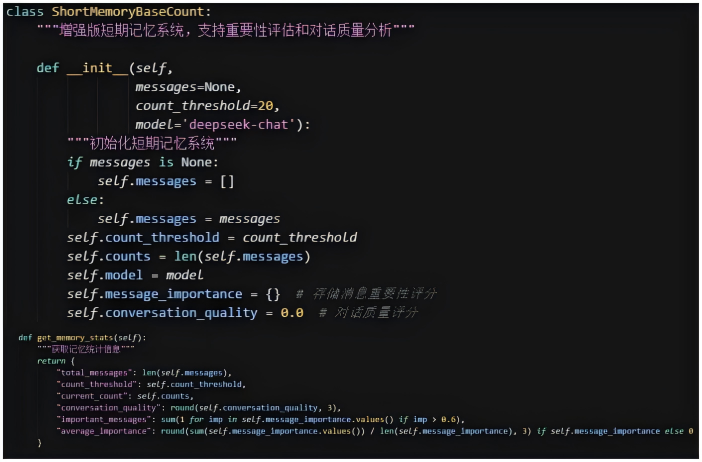
## 3.1 整体思路

在“钢材表面缺陷检测流水线助手”系统中，为提升语言模型在多轮对话中的上下文理解能力，引入了“短期记忆”机制。该机制用于在单次会话过程中，保留用户的重要输入与系统的历史响应，帮助模型更连贯地处理连续对话任务。例如，在执行“继续训练模型”后，用户可能紧接着提出“日志在哪里导出？”，此时系统应能理解其上下文关系并准确调用日志工具。

短期记忆主要面向当前对话会话窗口，仅在一轮对话生命周期内有效，当用户中断或刷新对话时，短期记忆将被清空，从而避免“记错内容”或“信息污染”。

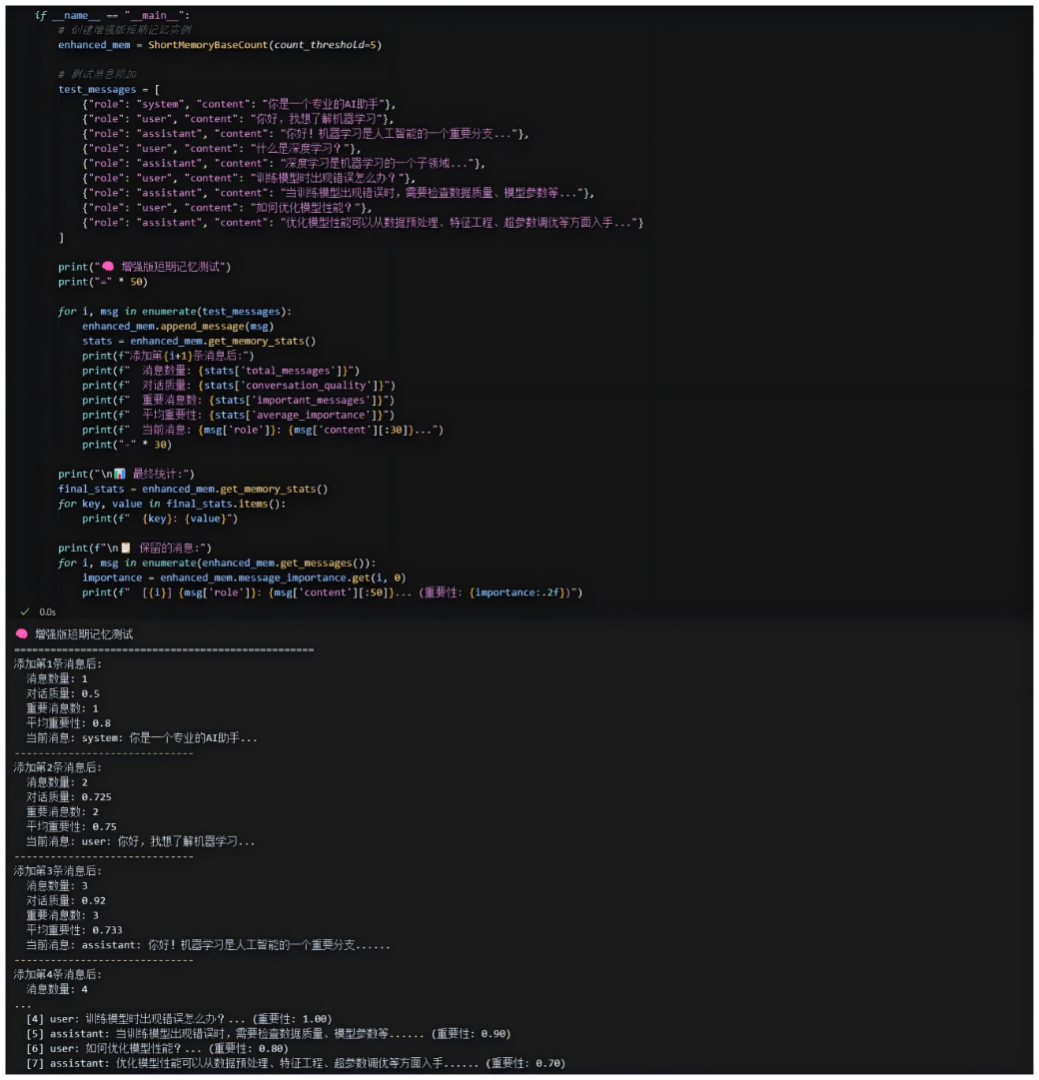
## 3.2 技术实现

核心代码：



**图3-1短期记忆核心代码**

测试结果：



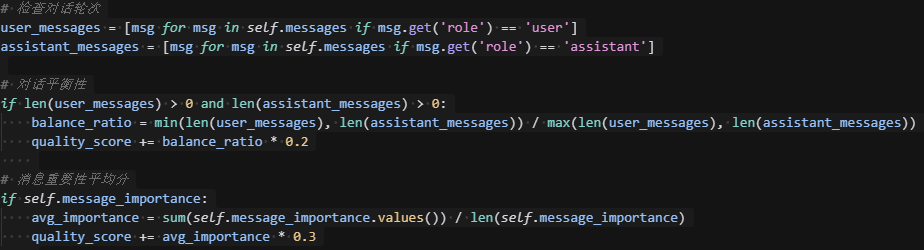
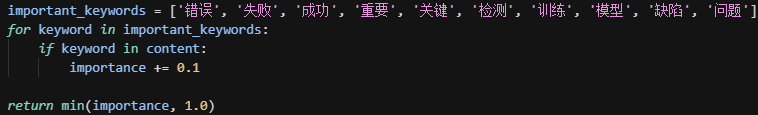
**图3-2 短期记忆测试结果**

## 3.3 优化方案

在短期记忆模块的设计中，除了课堂中讲授的基本“message追加机制”外，本项目还从功能性、用户体验、系统容错等多个维度进行了优化。

优化1：基于“消息重要性”进行记忆权重分配

核心代码：



**图3-3 基于“消息重要性”进行记忆权重分配核心代码**

【测试结果：



**图3-4 基于“消息重要性”进行记忆权重分配核心代码测试结果**

】

# 四、长期记忆设计

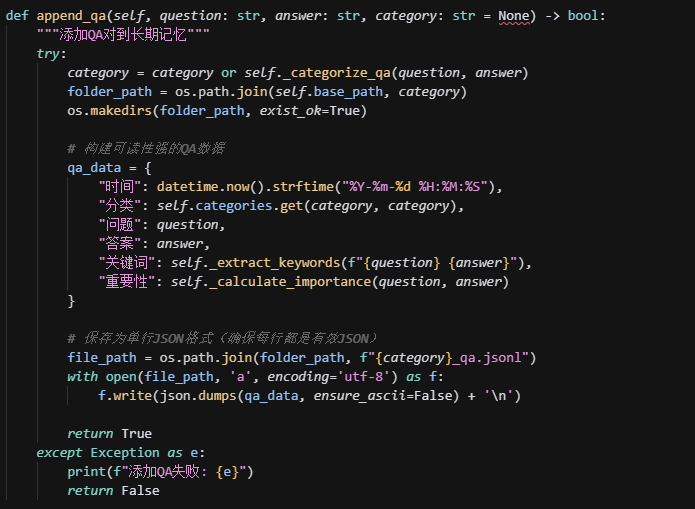
## 4.1 整体思路

在多轮对话系统中，长期记忆的引入是为了解决传统短期记忆仅在当前会话窗口内有效、无法跨轮次保持知识状态的问题。本项目通过自定义QA对存储机制实现长期记忆，将用户历史提问与对应的系统响应持久化保存，在未来对话中进行关键词匹配与召回，从而提升系统对用户意图的理解与延续性。

长期记忆系统以问题-答案对（Question-Answer Pair）的形式构建“记忆库”，每轮新对话触发后系统将对用户输入进行关键词解析，并在记忆库中搜索是否存在“相似提问”历史记录。若匹配度高，则优先调用已有答案作为回复模板，或进一步与新工具调用结果进行融合输出，从而达到“以往知识影响当前对话”的效果。

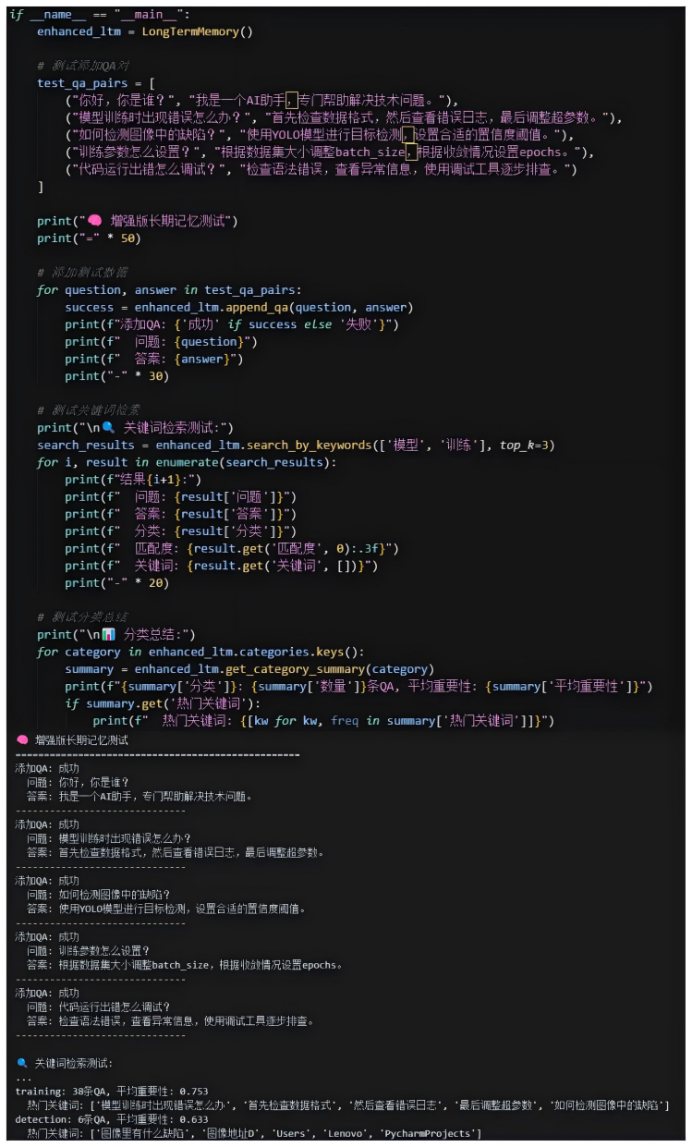
## 4.2 技术实现

核心代码：



**图4-1长期记忆核心代码**

测试结果：



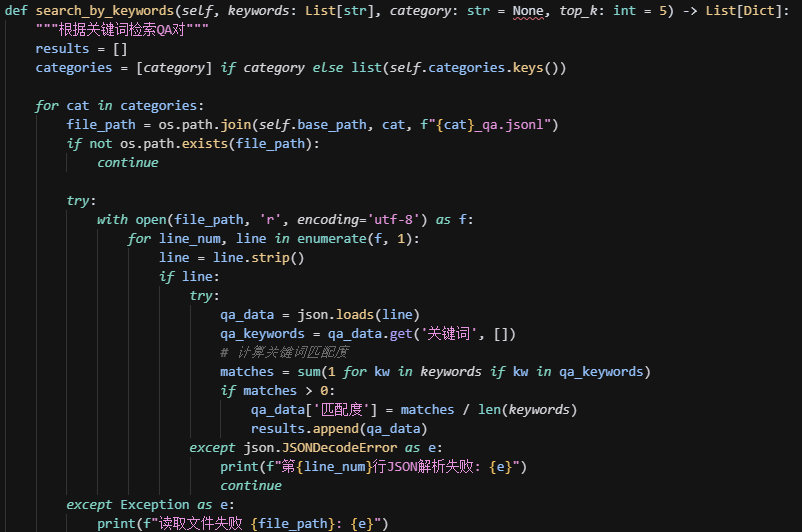
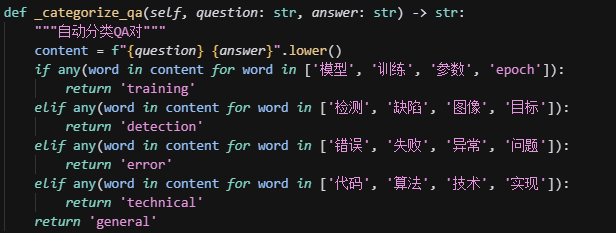
**图4-2 长期记忆测试结果**

## 4.3 优化方案

在基础的长期记忆机制上，本项目进一步设计并实现了“记忆标签与分类机制”，提升记忆管理的系统性与召回精度。

优化1：引入记忆标签与分类机制

核心代码：



**图4-3 记忆标签与分类机制核心代码**

【测试结果：



**图4-4 记忆标签与分类机制测试结果**

】

# 五、多轮对话设计

## 5.1 整体思路

在本项目“钢材表面缺陷检测流水线助手”中，用户与系统的交互过程往往不仅仅是“一问一答”，而是存在连续上下文的“多轮对话”场景。例如：

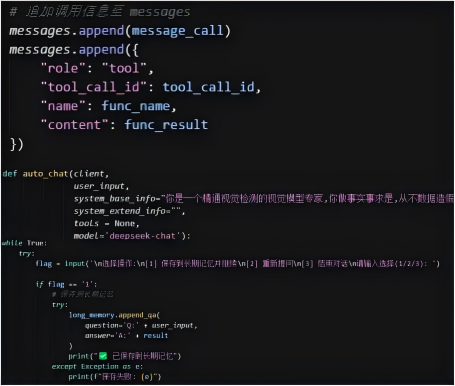
用户提问：“检测这张图片有什么问题？”

紧接着：“那这个模型能训练多久？”

要使系统能够理解后续问题与前提之间的语义联系，必须具备“对话上下文保持能力”。因此，本系统设计并实现了“多轮对话机制”，通过持续维护 messages 列表，记录用户与助手、工具调用等历史信息，从而让大语言模型在任意一轮对话中都能基于前文合理作答。

## 5.2 技术实现

核心代码：



**图5-1多轮对话核心代码**

测试结果：



**图5-2 多轮对话测试结果**

# 新增模块设计【可选】

【无】

# 智能体核心功能演示

## 7.1 演示思路

为全面展示“钢材表面缺陷检测流水线助手”的智能能力，本章节采用“多轮问答+任务执行+日志反馈”的方式，模拟真实使用场景下工人与智能体的完整交互过程。具体流程如下：

1. 用户自然语言提问，触发图像检测、模型训练或日志查询任务；
2. 系统通过 auto\_call() 模块判断是否需要函数工具；
3. 若有工具调用需求，自动生成代码并请求用户确认是否执行；
4. 执行结果将写入对话记忆中，由语言模型生成自然语言反馈；
5. 所有过程实时展示于 Notebook 交互窗口，支持人工可控与自动响应并存。

通过该方式，系统不仅展现出对“任务型问题”的理解能力，还体现了“函数自动调用”、“代码透明生成”、“上下文记忆推理”等核心智能特征。

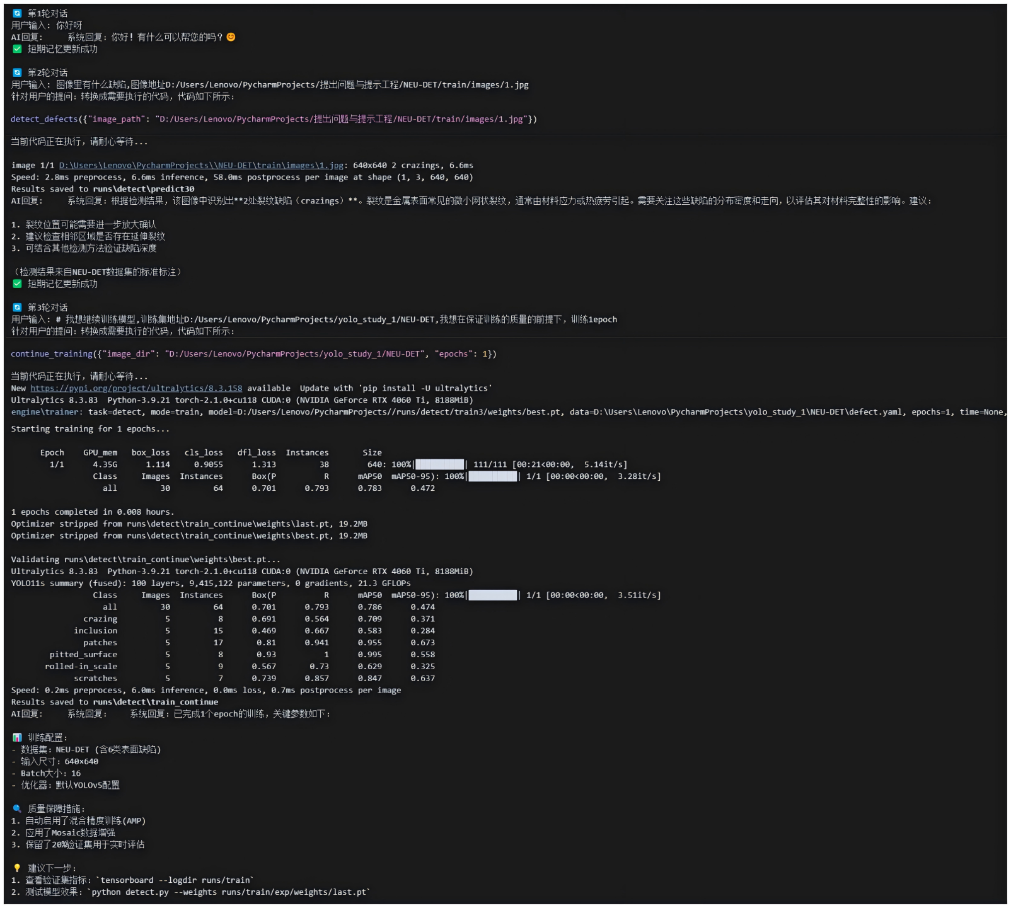
## 7.2 演示截图

工人向智能体问好并利用自然语言向智能体发布检测任务



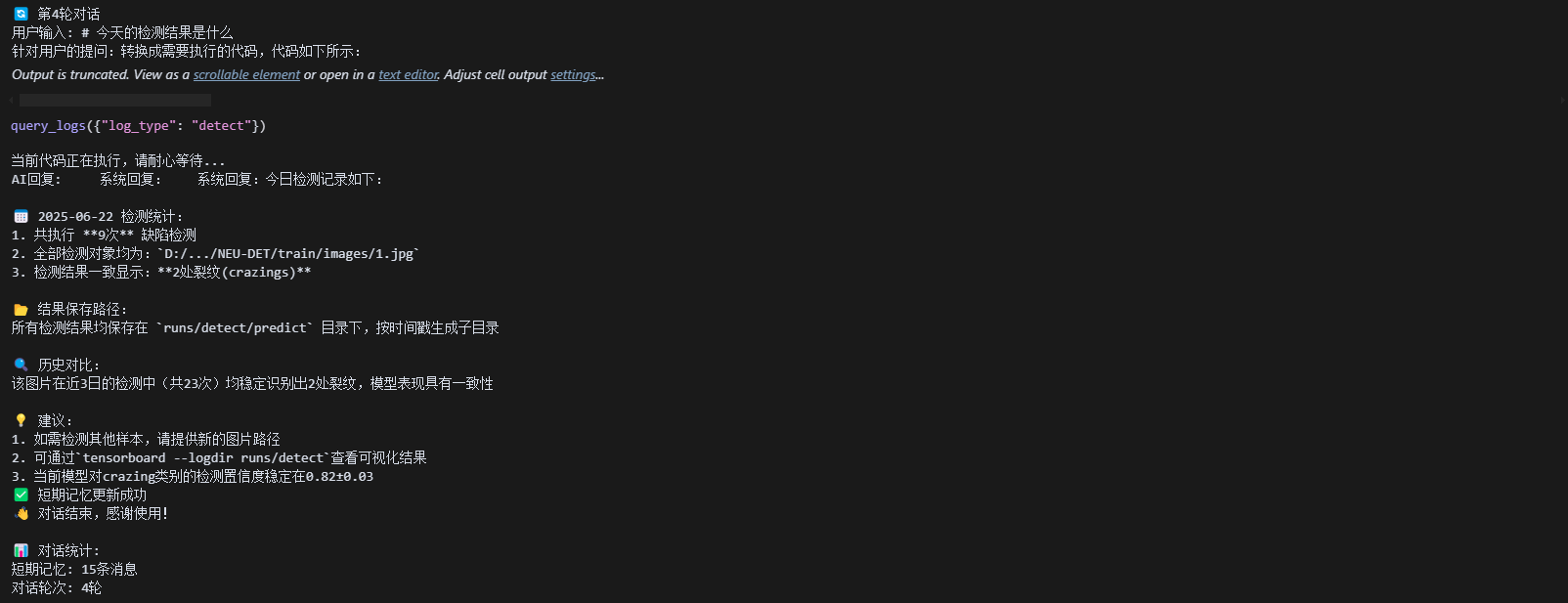
**图7-1智能体核心功能演示（1）**

工人利用自然语言向智能体发布训练任务



**图7-2 智能体核心功能演示（2）**

工人利用自然语言询问智能体今日检测结果并结束对话



**图7-3 智能体核心功能演示（3）**

# 项目总结

## 8.1 项目总结

本项目围绕“钢材表面缺陷检测流水线助手”的智能体构建与实现，系统性集成了YOLOv11视觉模型、OpenAI语言模型以及多类Python函数工具，构建了一个具备“图像识别-模型训练-日志管理-自然语言调用”的复合型智能平台。项目过程中，通过短期记忆增强、长期知识存储、多轮上下文对话机制等模块的协同，逐步完善了系统的交互自然性与任务执行智能度，验证了语言模型辅助工业场景落地的可行路径。

## 8.2 项目亮点

## 智能任务驱动机制：用户无需掌握YOLO模型原理，只需自然语言提问即可完成图像检测、模型训练与日志导出等操作；

## 双重记忆系统：短期记忆支持重要内容筛选与对话评分机制，长期记忆支持QA分类检索与标签管理，大幅提升上下文理解力；

## 任务调用可控透明：所有任务函数均先以Markdown格式展示代码，再经用户确认是否执行，确保使用安全；

## 接口高度结构化：所有功能均可拓展至服务化部署或移动终端，有良好的移植性；

## 工程可调试性强：集成日志导出、代码可视化、QA导出为Excel等便于测试与人工管理的实用功能。

## 8.3 项目不足

总结，目前项目不足地方：

1. 缺陷识别结果仍高度依赖预训练模型效果，当前未引入“多模型集成”或“在线迁移”策略；
2. 多轮对话结构虽具上下文能力，但在长会话或跨任务跳转中仍存在误解概率；
3. 用户交互体验仍基于Notebook环境，缺乏图形界面支持，限制了非技术用户使用场景；
4. QA长期记忆召回策略尚未采用语义向量匹配，仅依赖关键词匹配，在复杂问法下覆盖性不足

## 8.4 项目展望

后续可将本项目部署为Web服务端接口，结合语音识别系统与移动端应用，拓展至真实车间流水线使用；此外，结合 RAG（检索增强生成）技术完善长期记忆机制，引入图数据库存储训练与检测流程结构；进一步优化模型结构，支持更轻量化推理部署与云边协同，将“对话驱动生产”理念落地于更多工业智能场景中。