



## 基本信息

姓名：徐磊 年龄：20岁  
性别：男 籍贯：湖南益阳  
求职岗位：python 嵌入式 Alaganet 意向城市：长沙，广州，深圳  
电话：17763721622 邮箱：1457850671@qq.com

## 教育背景

2023-09 ~ 2027-06 湖南城市学院 (一本) 人工智能 (本科)

- 入选智能机器人实验室，拥有良好的项目开发经验
- 主修课程：深度学习、机器学习、智能优化算法、Python、数字图像处理与机器视觉、C++、C、数据结构、模拟数字信号处理等
- 参与了不同技术栈的学科竞赛，有丰富的团队协作经验

## 技能特长

### 编程语言：

C/C++：熟练使用 C 进行嵌入式及系统级开发，具备良好的数据结构基础。

Python：熟练使用 Python 进行自动化脚本编写、数据处理及测试工具开发，具备面向对象及函数式编程能力。有 Git、Docker 开发经验。

### 嵌入式：

掌握 STM32 开发，熟悉 STM32CubeMX、Makefile 等工具链。

熟悉 I2C、SPI、UART 等外设协议，具备裸机及 FreeRTOS 下的任务调度开发经验。

了解 Linux 基础命令及交叉编译环境，可完成简单驱动调试与应用移植。

### 运维管理：

熟悉 Linux (Ubuntu/CentOS) 系统管理，掌握 Shell 脚本、定时任务、日志分析、进程管理及网络配置。

具备 Docker 容器化部署基础，可编写 Dockerfile 及 docker-compose。

部署过个人网站 (xulei1112.cloud)。

### AI 工具使用：

熟练使用 Claude Code、OpenAI Codex、Cursor 等 AI 编程工具，用于代码生成、补全、理解与重构，提升开发效率。

具备基础的 Prompt Engineering 能力，能够利用大模型辅助调试、单元测试生成及技术文档撰写。

### 语言能力：

大学英语四级 (CET-4)，具备良好的技术文档阅读能力及基础英文报告撰写能力。

### Exelixi —— 可视化、可恢复、可审批的本地 Coding Agent

Python / LangGraph / FastAPI / WebSocket / LangChain / Tavily API

#### 项目描述：

为解决大模型编程 Agent 在实际工程中缺乏可视化、断点恢复、人工审批等边界能力的问题，基于 LangGraph 开发的本地 Agent。支持多轮会话、多 Agent 协作、Web UI / TUI / CLI 三端运行、上下文工程与 Harness 工程，目前已接近可用的本地 Agent 产品形态。

#### 核心工作：

- 多 Agent 协作架构：基于 LangGraph 构建有状态图 workflow，包含 intent\_router（意图路由）、planner（计划制定）、searchAgent（Tavily 搜索）、codeAgent（代码生成与调试）、verifier（只读验收）、context\_compressor（自动压缩）等多个专用 Agent 节点，任务按阶段自动流转。
- 三端运行界面：Web 端（FastAPI + WebSocket）实现在浏览器中输入任务、查看时间线、处理命令审批与 Agent 追问；Textual TUI 终端界面支持多轮本地编码会话；Rich CLI 适合脚本化运行与测试。
- 上下文工程（Context Engineering）：设计三层记忆体系（rules / working\_memory / history\_summary），结合 TODO.md / NOTEPAD.md / HISTORY\_SUMMARY.md 持久化。context\_compressor 在达到 token 阈值时自动压缩消息窗口并写入历史摘要，避免长会话超出上下文限制。
- Harness 工程：实现人工审批（inline/deny/auto 三种模式，高风险命令和文件写入可触发审批）、Checkpoint（light/strict 两级，支持从任意 checkpoint 恢复）、Trace（事件日志 JSONL + 摘要 JSON + 时间线 MD，用于调试和复盘）。
- Workspace 隔离：每个任务创建独立 workspace 目录，自动维护 TODO.md / NOTEPAD.md / 文件产物，避免任务间污染。

#### 难点与解决：

- 多 Agent 协作中状态管理复杂 → 使用 LangGraph 的 StateGraph 定义全局状态类型，每个 Node 只读写自己关注的状态片段，通过 Checkpoint 实现任意节点的断点恢复。
- 长会话上下文超限 → 设计三层记忆架构 + 自动压缩节点，在 token 接近限制时清除旧消息并生成结构化的摘要文本，保留关键决策和任务进度。
- Agent 工具调用存在安全风险 → 建立命令/文件写入两级审批机制，高风险操作（rm -rf、系统级命令、大范围文件修改）自动触发人工确认。

#### 项目成果：

- 完成 Web UI / Textual TUI / Rich CLI 三端运行，用户可在浏览器或终端中持续对话，让 Agent 搜索、写文件、运行命令、请求确认、记录记忆，并在失败后继续恢复。
- 支持从任意 checkpoint 恢复中断会话，保留 workspace 文件快照 + 状态 + 事件日志，适合长时间开发任务。
- Trace 系统记录完整运行事件，支持事后调试与复盘分析。
- 该项目作为个人 Coding Agent 持续使用和迭代，目前已接近可用的本地 Agent 产品壳，可在日常开发中辅助编码、搜索、调试与项目维护。

## 全国大学生统计建模大赛

Python / PyTorch / XGBoost / scikit-learn / pandas / NumPy / matplotlib

**项目描述:** 针对光伏电站 6 类故障（正常/短路/劣化/开路/部分阴影/完全阴影）分类准确率低、数据不平衡的问题，在约 1000 万条运行数据上设计并实现深度学习+集成学习混合诊断模型。

### 核心工作:

- 混合模型架构: 构建 CNN+LSTM+Attention 提取时序特征，再输入 XGBoost 分类，结合深度特征学习与树模型的小样本鲁棒性。
- 注意力机制: 在 LSTM 输出层后加入可学习的时间注意力（Temporal Attention），自动聚焦关键时间步，提升对局部故障的敏感度。
- 双策略处理不平衡: Focal Loss ( $\gamma=2.0$ ) + sklearn 类别权重计算，双重缓解多数类偏差，显著提升少数类召回率。
- 完整评估体系: 混淆矩阵、宏平均 F1/PR/ROC-AUC、注意力热力图、XGBoost 特征重要性排序。

### 难点与解决:

- 时序中短暂故障信号难聚焦 -> Temporal Attention 自动学习时间步权重，热力图验证模型关注故障发生时刻。
- 超大规模训练梯度震荡 -> AdamW + 双重裁剪 + 余弦退火+Warmup，60 个 epoch 稳定收敛。
- 成果: 完成 4 种递进模型对比实验，完整混合模型在宏平均 F1 上显著优于纯 CNN/LSTM 基线，具备工业级故障诊断潜力。

## 基于 MQTT 的智能仓储入柜管理系统

Python / Tkinter / pandas / matplotlib / MQTT / pyserial / 贪心算法

**项目描述:** 解决电商仓储中纸箱货位匹配效率低的问题，开发从尺寸输入->贪心匹配->可视化推荐->Excel 管理->硬件联动的桌面管理系统。

### 核心工作:

- 贪心推荐引擎: 遍历 50 个格子 + 6 种旋转方向，计算体积占用率并选取最优解，利用 set 去重减少冗余计算。
- 仓储分区策略: 可按尺寸自定义分区，动态支持区域扩展，提升空间利用率。
- 图形界面: Tkinter + matplotlib 实现仓库俯视图、分区彩色标识、历史记录面板。
- 数据管理: Excel 双向同步 (pandas)，支持导出/修改/导入; 历史记录自动记录最近 100 条操作。
- 硬件通信: MQTT (paho-mqtt) 实现格子状态实时发布/订阅; pyserial 串口监视器支持多波特率、十六进制显示、日志记录。

### 项目成果:

- 完成从输入->匹配->推荐->可视化->导出->硬件联动的全闭环。
- 支持随时扩展格子、6 维旋转适配、区域智能划分，提供串口/MQTT 双通道硬件通信能力，可直接用于小型仓储数字化管理。

## 基于 STM32 的多传感器姿态感知与无线控制系统 STM32F103ZET6 / C 语言 / STM32CubeMX / HAL 库 / PWM / ADC /

DMA / USART

**项目描述:** 为搭建无线遥控、姿态自适应的智能控制平台，基于 STM32F103 实现六轴姿态传感器、光敏传感、蓝牙遥控与多路舵机控制的嵌入式系统。

### 核心工作:

- 多外设架构: CubeMX 配置 TIM1 (4 路 PWM, 50Hz)、ADC1 (光敏, 12 位)、USART1/2/3 (115200, DMA 收发)，main 循环协调运行。

- 姿态传感器驱动：USART2 中断接收 JY61P 数据包（11 字节），状态机解析包头(0x55 0x53)->校验和->Roll/Pitch/Yaw ( $\times 180/32768$ )，Roll 角映射舵机实现姿态跟随。
- 舵机驱动库：封装 Servo\_SetAngle，PWM 占空比 50-250 $\mu$ s 对应 0-180°，支持 4 路独立控制。
- 蓝牙无线控制：USART3 + DMA + 空闲中断接收字符串，atof 解析指令控制舵机 2/4 角度，关闭半传输中断(DMA\_IT\_HT)防干扰。
- 光控继电器：ADC 采集光敏电压，阈值 150 判断光照，控制两路继电器通断。
- 成果：实现蓝牙遥控 4 路舵机、姿态自动角度跟随、光敏自动控制三大功能；3 个 USART 通道+DMA 协同工作互不干扰，可作为智能机械臂/机器人底盘的控制基础平台。

### 基于 51 单片机的红外循迹小车

STC89C52 / C 语言 / Keil / 红外传感器 (TCRT5000) / L298N 电机驱动 / PWM

**项目描述：**为实现低成本、高可靠的黑白线循迹功能，基于 51 单片机设计并制作了一辆红外循迹小车，支持自主循迹、差速转向及速度调节。

#### 核心工作：

- 硬件搭建：搭建 STC89C52 最小系统，配置 4 路红外传感器 (TCRT5000) 呈一字/弧形布局，L298N 驱动双直流电机。
- 循迹算法：根据 5 路传感器二进制状态查表决策，实现直行、左/右转弯、原地旋转（差速反转）及停止。
- PID 调速（可选）：编写简易增量式 PID，通过 PWM（定时器模拟）调节电机占空比，使循迹曲线更平滑。
- 软件抗干扰：连续多次采样取众数，消除传感器抖动导致的误判。
- 成果：小车可稳定循迹黑白跑道，实现自动循迹行驶。

### 基于 R-CNN 的棋盘识别与自动对弈

Python / OpenCV / Faster R-CNN / TensorFlow/Keras / PyAutoGUI

技术栈：Python、OpenCV、Faster R-CNN、TensorFlow/Keras、PyAutoGUI

#### 核心工作：

- 棋盘检测 (OpenCV + R-CNN)：基于 Faster R-CNN 训练棋盘定位模型，结合 OpenCV 的轮廓分析 (Canny+霍夫) 与 ORB 匹配，构建三级流水线，检测鲁棒性显著提升。
- 棋子分类：OpenCV 提取轮廓几何特征 (圆度/面积比) 作为降级逻辑，配合轻量级 CNN 识别棋子，模型识别失败仍有降级策略。
- 全闭环控制：PyAutoGUI 模拟点击，完成截图->解析->决策->落子自动化。

#### 难点与成果：

- 棋盘样式多变 -> OpenCV 几何检测与 R-CNN 互补，覆盖多数场景。
- 实现全自动人机对战，模块可独立替换，适用于任意屏幕井字棋。

## 自我评价

工作积极认真，细心负责，熟练运用办公自动化软件，善于在工作中提出问题、发现问题、解决问题，有较强的分析能力；勤奋好学，踏实肯干，动手能力强，认真负责，有很强的社会责任感；坚毅不拔，吃苦耐劳，喜欢迎接新挑战。