

基于 RK3588 与计算机视觉的 AI 智能营养膳食结算系统

摘要

本系统设计并实现了一款基于 RK3588 处理器与计算机视觉技术的 AI 智能营养膳食结算系统，旨在解决传统餐饮场景中人工结算效率低下及科学膳食管理需求日益增长的问题。该系统由智能识别台与餐盘两部分组成，通过高精度摄像头采集菜品图像，结合 YOLOv5 深度学习算法实现快速、准确的菜品识别，同时利用嵌入式压力传感器实时测量菜品重量。系统基于识别结果与重量数据，自动计算餐食价格及营养成分（包括能量、蛋白质、脂肪、碳水化合物等宏量营养素），并通过交互式屏幕实时显示结算信息与个性化膳食建议，为用户提供科学的饮食指导。

在技术实现上，系统采用 RK3588 高性能处理器作为核心计算单元，支持多任务并行处理，确保识别与结算过程的高效性。计算机视觉模块通过大规模菜品数据集训练，识别准确率可达 95% 以上，并支持动态更新以适应不同餐饮场景。压力传感器采用高灵敏度应变片设计，重量测量误差控制在 $\pm 1\text{g}$ 以内，确保数据精准可靠。

本系统的核心创新有：集成 DeepSeek 大模型，为用户提供智能化的营养分析与膳食建议服务。通过自然语言交互，用户可以与系统对话，获取更详细的饮食指导，例如：查询特定菜品的营养成分；获取基于个人健康目标的膳食搭配建议；了解不同烹饪方式对营养的影响；获得特殊饮食需求（如减脂、增肌、糖尿病饮食）的专业建议。

本系统的应用场景广泛，可部署于学校食堂、企事业单位餐厅、医院营养食堂等场所。在学校食堂中，系统可显著减少学生排队结算时间，同时帮助青少年群体培养健康饮食习惯；在企事业单位餐厅，系统可为员工提供定制化营养报告，助力职场健康管理；在医院场景下，系统可结合患者营养需求，提供精准的膳食建议，辅助临床营养治疗。

第一部分 作品概述

1.1 功能与特性

智能识别与自动结算：本系统基于计算机视觉与深度学习技术，采用 YOLOv5 算法实现对餐盘中菜品的高精度图像识别。结合嵌入式高灵敏度压力传感器获取重量数据，系统可自动完成菜品识别与价格计算，显著提升餐饮场景下的结算效率，减少人工干预和排队时间。

智能化营养分析与个性化建议：系统集成 DeepSeek 大模型，具备强大的自然语言处理与营养分析能力。能够实时解析菜品的能量、蛋白质、脂肪、碳水化合物等宏量营养素含量，并根据用户的健康信息（如年龄、性别、体重、活动强度等）生成个性化膳食建议，满足减脂、增肌、控糖等多样化饮食需求。

友好高效的人机交互界面：系统采用简洁直观的扁平化界面设计，支持触控操作，信息展示清晰、交互便捷。界面实时显示菜品识别结果、重量及营养成分，并以图表形式可视化呈现。用户可通过文字输入与 DeepSeek 大模型进行自然语言交互，获取个性化的饮食建议与健康指导，提升整体使用体验。

1.2 应用领域

教育与公共餐饮场景：系统有效提升就餐效率，缓解高峰期拥堵，并通过营养可视化与个性化建议，促进科学饮食意识的建立。

医疗与康养服务场景：依托 DeepSeek 大模型的精准营养解析能力，系统可基于用户健康状况推荐适宜餐食，辅助临床营养干预与健康管理。

商业与高端服务场景：系统以提升用户体验为核心，提供快速结算、营养展示与膳食建议，增强餐饮服务的专业性与智能化水平。

综上所述，本系统融合计算机视觉、深度学习与大模型自然语言处理技术，具备高精度菜品识别、重量检测、自动结算及智能化营养分析等核心功能，具备良好的场景适配能力，具有较强的实用性与推广价值，有助于推动餐饮管理向智能化、精细化方向发展。

1.3 主要技术特点

本系统采用多项先进技术，构建了一个高效、智能、可扩展的营养膳食结算平台。首先，在视觉识别方面，基于 YOLOv5 深度学习算法实现菜品图像的高

精度识别，具备良好的抗干扰能力与泛化性能，识别准确率可达 95%以上。其次，系统融合嵌入式压力传感技术，通过高灵敏度应变片实时获取菜品重量数据，误差控制在±1g 以内，确保计价精准。

在数据分析与智能化服务方面，系统集成 DeepSeek 大模型，支持自然语言交互，能够解析用户健康信息并生成个性化膳食建议，实现从识别到推荐的全流程智能化服务。同时，系统采用 RK3588 高性能处理器作为核心计算单元，具备强大的多任务处理能力，保障图像识别、重量采集、营养分析与人机交互等模块同步高效运行。

此外，系统界面设计简洁直观，支持触控操作与图表可视化展示，提升用户体验；整体架构模块化、部署灵活，适用于多种餐饮场景，具备良好的扩展性与应用适应性。

1.4 主要性能指标

技术特性	性能指标	应用优势
智能餐盘+压力传感	3 秒内快速识别结算	提升结算效率，减少人工干预
视觉识别技术	菜品识别准确率 95%	支持数据库扩展，兼容多样菜品，满足速度、种类及营养分析需求
营养数据可视化	实时显示于交互屏幕	提供科学膳食指导，无需手机即可获取信息，促进健康饮食管理
多场景适配	兼容食堂/快餐/自助等场景	智能化结算灵活高效，优化用户体验

1.5 主要创新点

- (1) 基于 RK3588 的高性能计算平台：采用八核处理器与 6TOPS NPU 算力，实现多任务并行处理，显著提升系统响应速度与稳定性。
- (2) 高精度称重技术：采用压力传感器实现±1g 精度称重，配合智能去皮算法，确保各类菜品的重量计量准确无误
- (3) 多模态交互设计：集成触控屏自然语言查询，用户可通过文字快速获取营养信息，操作更便捷。
- (4) DeepSeek 营养大模型：基于用户健康档案（如年龄、血糖水平等）生成个性

化膳食建议，支持减脂、增肌等多样化需求。。

- (5) 模块化扩展设计：硬件支持对接智能餐柜、ERP 系统，软件支持 OTA 升级，适配食堂、医院、餐厅等多种场景。
- (6) 抗干扰视觉识别：采用改进的 YOLOv5s 模型，在复杂光照、餐盘重叠等场景下仍保持 95%以上识别准确率。
- (7) 板端部署图像识别模型：结合 RK3588 内置的 NPU 算力，充分发挥边缘计算低延迟、高可靠性的特点，减少对云端资源的依赖。

1.6 设计流程

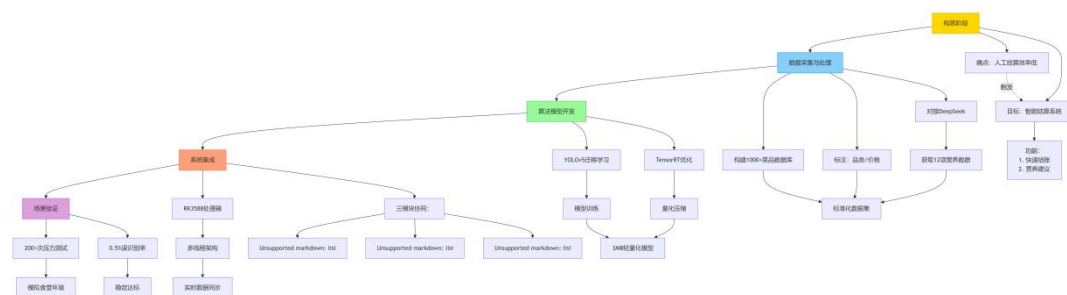
(1) 构思阶段：在学校生活中，我们发现食堂的结算任然通过人工，大大降低了效率，为了提高食堂的结算速度，同时满足同学们多样化的饮食需求，我们想到做一种智能结算系统，既能够解决结账效率问题，也能为饮食提供建议。

(2) 数据采集与处理阶段：构建包含 1000+样本的标准化菜品图像数据库，每道菜品均标注品类名称和市场化价格标签。通过与 DeepSeek 大模型对接，自动获取菜品的能量、蛋白质等 12 项营养指标数据，实现菜品信息与营养数据的智能关联。

(3) 算法模型开发阶段：基于 YOLOv5 框架进行迁移学习训练，利用 TensorRT 工具对模型进行量化压缩和推理优化，最终将识别模型体积控制在 10MB 以下，确保在嵌入式设备上的高效运行。

(4) 系统集成阶段：以 RK3588 为核心处理器，开发多线程驱动架构，实现高清摄像头、高精度称重模块和触摸显示屏的协同工作，确保各模块数据同步。

(5) 场景验证阶段：模拟食堂环境进行压力测试，累计完成 200+次餐品识别结算，系统综合误识别率稳定控制在 0.5%以下。

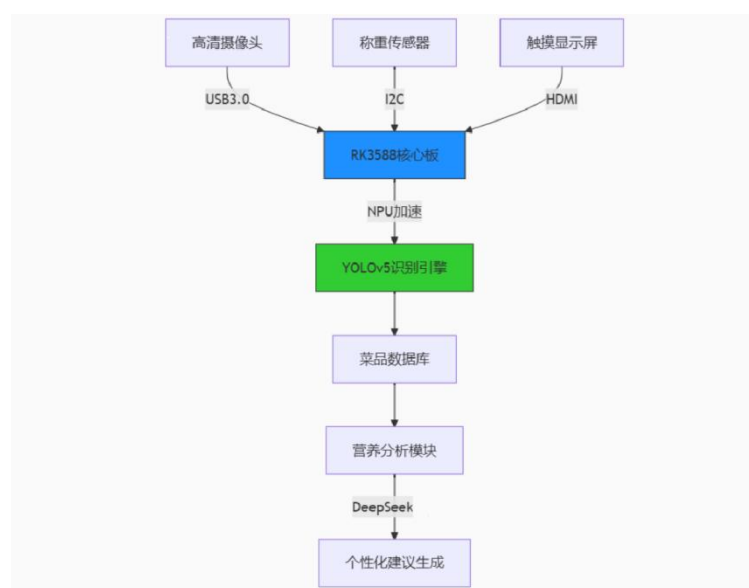


系统设计流程图

第二部分 系统组成及功能说明

2.1 整体介绍

该系统由多个子模块协同工作，包括数据采集、处理、识别、分析和建议生成等部分。高清摄像头通过 USB3.0 接口与 RK3588 核心板相连，用于拍摄餐盘中的菜品图像；称重传感器通过 I2C 总线测量菜品重量，辅助识别菜品种类和数量；触摸显示屏则通过 HDMI 接口展示结算信息和用户交互界面。作为系统的中央处理器，RK3588 核心板负责协调各模块的工作，接收来自传感器的数据并进行初步处理，随后传递给 YOLOv5 识别引擎进行菜品识别。YOLOv5 识别引擎基于深度学习模型，将图像转化为可识别的菜品信息并与称重数据结合，确定具体的菜品种类和数量。菜品数据库存储标准化的菜品图像及相关信息，为识别结果提供参考依据。营养分析模块根据这些信息计算出每道菜品的能量、蛋白质等营养指标，并通过 DeepSeek 大模型进一步优化数据支持，增强系统的智能化水平。最后，个性化建议生成模块综合用户的饮食习惯和营养需求，生成定制化的饮食建议。整个系统中，RK3588 核心板作为中枢确保数据准确传输和处理，而触摸显示屏则作为人机交互界面，展示最终的结算信息和建议，同时接收用户反馈和指令，全面提升食堂的服务质量和用户体验。

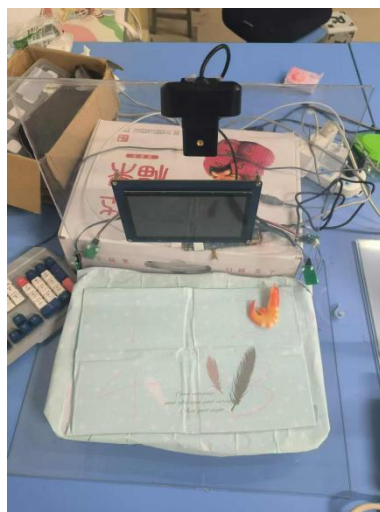


整体流程图

2.2 硬件系统介绍

2.2.1 硬件整体介绍：

硬件系统主要由主体模块、识别台模块和餐盘模块三部分组成。主体模块：包括 RK3588 处理器、内存和存储设备，负责运行系统软件和处理数据。识别台模块：包括高清摄像头和压力传感器，用于采集菜品图像和测量重量。餐盘模块：包括餐盘和压力传感器，用于盛放食物并测量重量。



硬件整体图

2.2.2 机械设计介绍

机械设计主要涉及餐盘和识别台的结构设计。餐盘采用轻质材料制成，底部安装有压力传感器，用于测量食物重量。识别台设计有固定摄像头的位置和传感器的安装槽，确保采集到的图像清晰且测量准确。

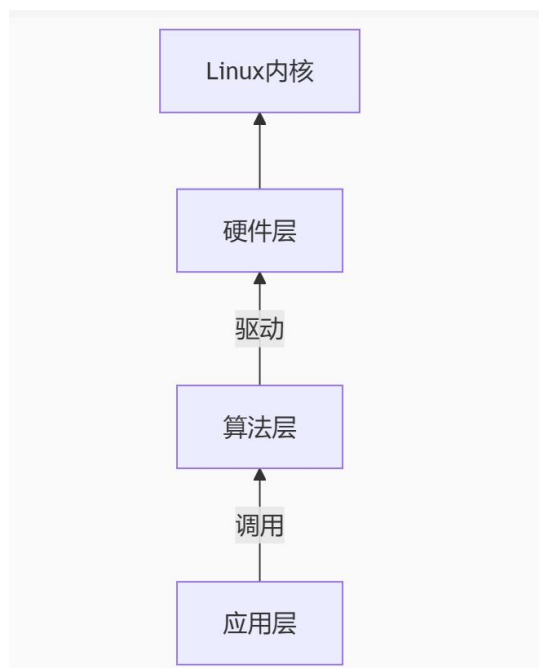


识别台结构图

2.3 软件系统介绍

2.3.1 软件整体介绍：

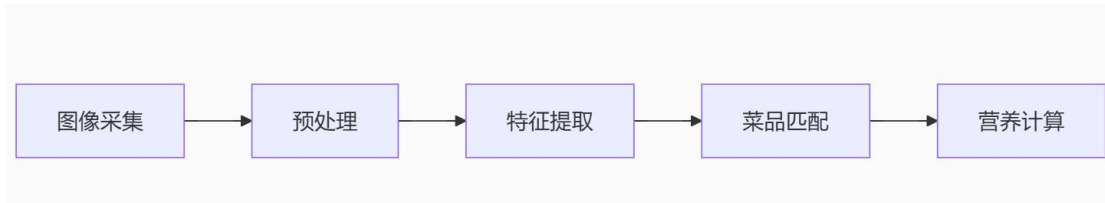
软件的系统架构分为四个主要层次：**Linux 内核**、**硬件层**、**算法层**和**应用层**。这种分层结构有助于模块化设计，使得每个层次可以独立开发和优化。**Linux 内核**提供底层的操作系统支持，管理系统的硬件资源，确保系统的稳定性和安全性。在 **PC 端**部署时，**Linux 内核**可以利用，实现资源的动态分配和扩展。包括摄像头、**CPU**、**GPU** 等硬件设备。这些硬件直接与软件交互；硬件资源可能通过虚拟机进行抽象和管理。包含用于图像处理和营养计算的算法。利用深度学习模型进行图像识别和特征提取，提高识别的准确性和效率。在 **PC 端**，应用层可以通过 **Web 界面**或用户交互。



软件整体流程图

2.3.2 软件各模块介绍

- (1) 图像采集：通过摄像头实时采集食物图像。
- (2) 预处理：对图像进行亮度、对比度、裁剪等处理，提高图像质量。
- (3) 特征提取：利用深度学习模型从图像中提取特征。
- (4) 菜品匹配：将提取的特征与数据库中的菜品进行匹配，识别菜品。
- (5) 营养计算：根据识别出的菜品，计算其营养成分。



整体设计流程图

```

35  MainWindow::MainWindow(QWidget *parent)
60      clearTableTimer = new QTimer(this);
61      clearTableTimer->setInterval(10000); // 10秒
62      connect(clearTableTimer, &QTimer::timeout, this, [=](){
63          for (int i = 0; i < 4; ++i) {
64              tableMgr->setRowData(i, "", 0);
65          }
66      });
67      clearTableTimer->start();
68
69      // 多路重量传感器初始化
70      m_multilightSensor = new MultiLightSensor(this);
71      static int sensorValues[4] = {0};
72      connect(m_multilightSensor, &MultiLightSensor::valuesChanged, this, [=](int v1, int
73          sensorValues[0] = v1; // 左上
74          sensorValues[1] = v2; // 右上
75          sensorValues[2] = v3; // 右下
76          sensorValues[3] = v4; // 左下
77          // 可在此处联动UI显示重量值
78      });
79      m_multilightSensor->start();
80      // 检测标签初始化
81      m_detectLabel = new DetectLabel(this);
82      connect(detect, &ObjectDetect::detectLabel, this, [=](const QString &label, con
83          ui->label_7->setText(QString("检测标签: %1").arg(label));
84          clearTableTimer->start(); // 每次检测到标签都重置定时器
85          if (label.isEmpty()) {
86              // 没有检测到标签, 清空所有表格行
87              for (int i = 0; i < 4; ++i) {
88                  tableMgr->setRowData(i, "", 0);
89              }
90      return;
  
```

代码图 1


```
void ObjectDetect::run(){
    ret = obj_init_post_process();
    if (ret != 0){
        printf("obj_init_post_process fail! ret=%d \n", ret);
        goto out;
        return;
    }
    //2、读取模型model_path文件内容，将模型初始化到rknn_app_context_t *rknn_app_ctx里面
    ret = init_yolo_model(modelPath.toString().c_str(), &rknn_app_ctx);
    if (ret != 0){
        printf("init_yolo_model fail! ret=%d \n", ret);
        goto out;
        return;
    }

    if(videoStatus ==0){
        //摄像头输入: v4l2-ctl --list-devices查看系统设备“USB Camera: USB Camera”的设备/dev/video
        qDebug()<<"摄像头打开中请稍等";

        sleep(2);
        cv::VideoCapture cats("/dev/video21");
        if(!cats.isOpened()){
            qDebug()<<"摄像头wuxiao";
            qDebug()<<"cwu:"<<strerror(errno);
        }

        cats.set(cv::CAP_PROP_FRAME_WIDTH,640);//设置后无法：无法打开“单”摄像头节点?
        cats.set(cv::CAP_PROP_FRAME_HEIGHT,480);//设置后无法：无法打开“单”摄像头节点?
        //      cats.set(cv::CAP_PROP_FPS, 15);
        Mat fram;
        QMap<int,vector<cv::Point*> mapId;
        QMap<int,int>idCount;
```

代码图 2

第三部分 完成情况及性能参数

3.1 整体介绍

由显示区，识别区和主控区组成。显示区用于显示相关菜品质量价格以及 AI 建议等，识别区用于识别所选菜品以及相应质量。

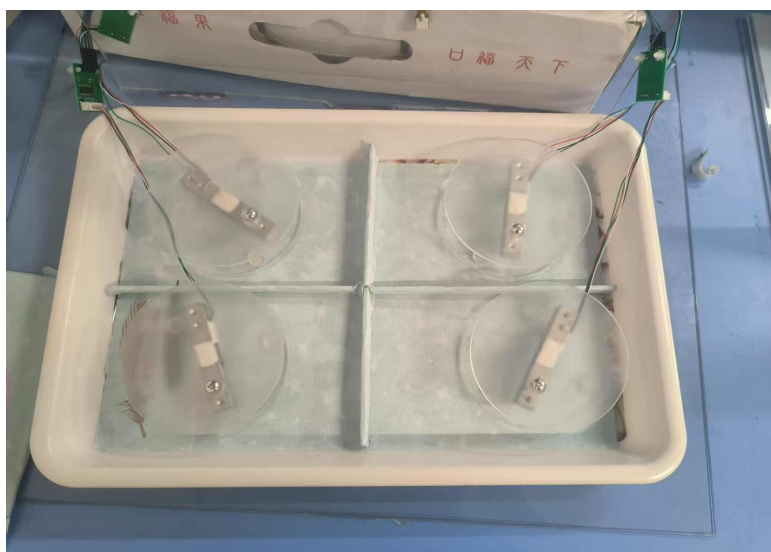


系统整体图

3.2 工程成果（分硬件实物、软件界面等设计结果）

3.2.1 机械成果：

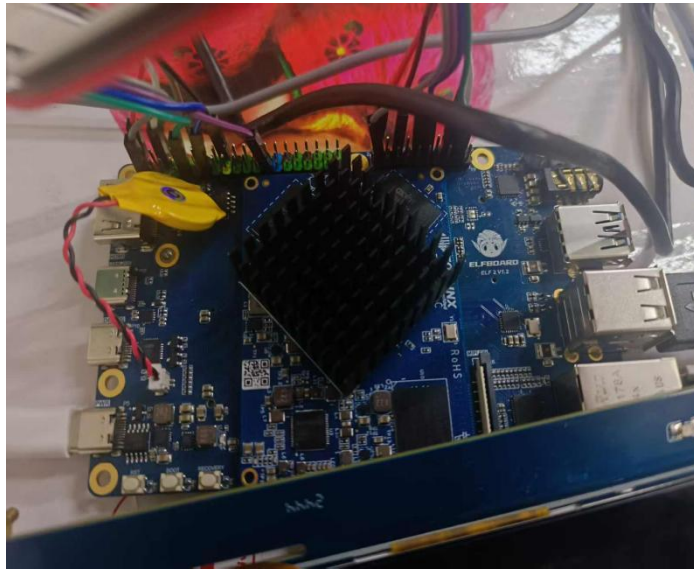
由四个压力传感器与自制的盘子组成



识别台结构图

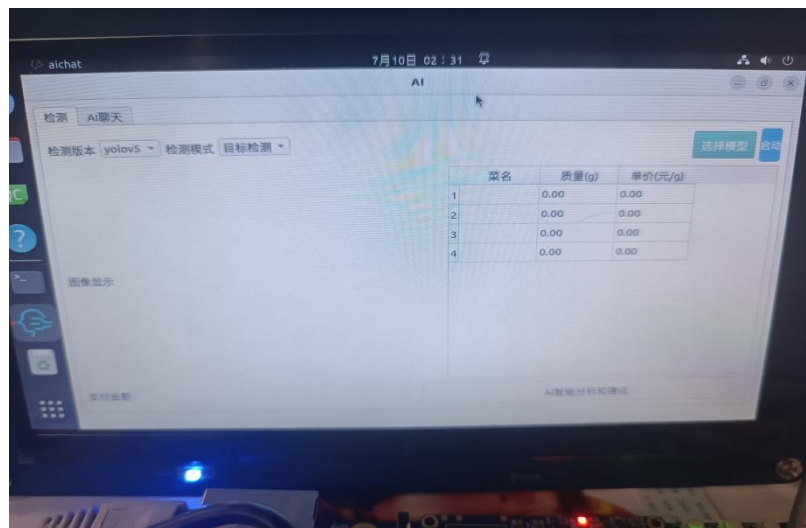
3.2.2 电路成果：

外接四个 HX711 压力传感器

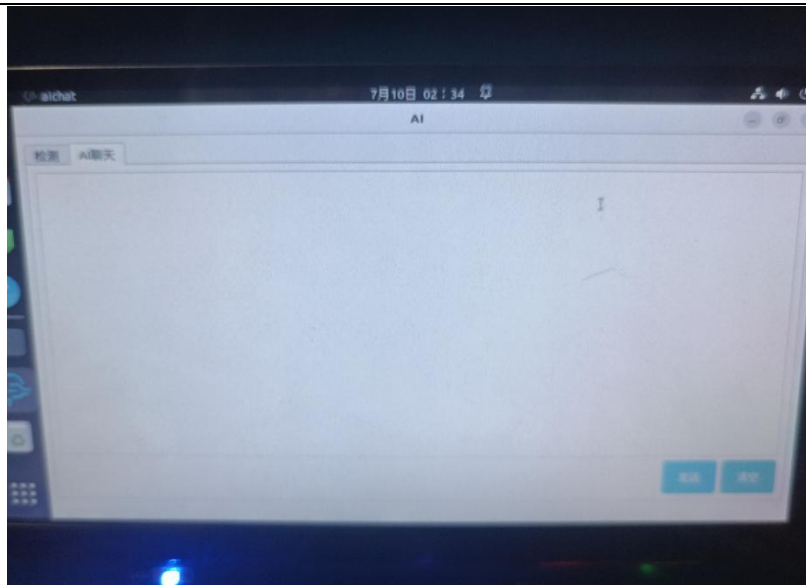


压力传感器连接图

3.2.3 软件成果：



QT 界面成果显示图 1



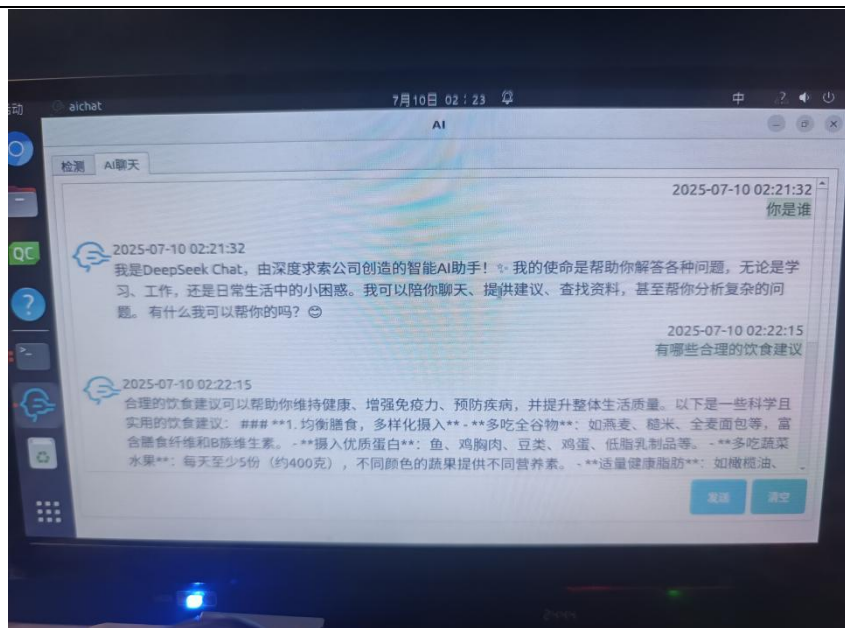
QT 界面成果显示

3.3 特性成果（逐个展示功能、性能参数等量化指标）；

菜品识别和显示菜名质量单价，并计算总价，还可以给出饮食建议。



识别结果图



AI 展示图

第四部分 总结

4.1 可扩展之处

- (1) 系统容量扩展：可通过优化算法和硬件配置，将同时识别的菜品数量提升，满足更高客流量的用餐需求；
- (2) 识别精度升级：采用更高分辨率的工业相机配合改进的 YOLOv8 模型，可将识别准确率提升至 98%以上；
- (3) 数据可视化增强：开发配套移动端 APP，实现消费记录、营养分析的实时查看和历史数据追溯功能；
- (4) 支付方式扩展：支持接入更多电子支付平台，并开发会员积分系统；
- (5) 智能推荐优化：结合用户饮食偏好大数据，提供更精准的个性化膳食建议。

4.2 心得体会

在本次基于 RK3588 与计算机视觉的 AI 智能营养膳食结算系统的研发过程中，从最初的方案论证到最终的成品落地，每一个环节都凝聚着我们的心血与智慧。记得在芯片选型阶段，我们反复对比了多种处理器的性能参数，最终选择了兼具算力与能效的 RK3588；在机械结构设计时，为了确保称重精度，我们测试了多种减震方案；软件调试阶段更是经历了无数个不眠之夜，选择合适的图像训练模型，标注成千计算的图片，只为更好 优化识别准确率。每当遇到技术瓶颈

时，团队都会集思广益，从不同角度寻找突破口。当看到系统成功识别出第一道菜品时，那种喜悦之情至今难忘。这段研发历程让我深刻体会到，专注投入的状态是如此美妙，解决问题的过程虽然艰辛，但每一次突破都让人倍感充实。更重要的是，我们不仅完成了一个智能产品，更收获了面对困难时的坚持与勇气，这些宝贵的经验必将成为我们未来继续创新的动力源泉。

第五部分 参考文献

- 1.冯杨. 营养评价与膳食智能决策支持系统设计研究[J]. 硕士电子期刊出版信息, (2013 年第 01 期).
- 2.张悦琳、王创剑. 基于人工智能知识库的营养膳食推荐系统研究[J]. 工程科技 II 辑;工程科技 I 辑;信息科技, 2023-05-23.
- 3.刘晓东、田丽丽. 医院智能营养膳食分析系统的建立与应用[J]. 医药卫生科技; 信息科技, 2017-05-05.
- 4.费斐. 基于改进算法的营养评价与膳食智能决策支持系统[J]. 信息科技, 2016-06-13.