

基于搭载 RK3588 芯片的飞凌嵌入式 ELF2 开发板的 肺癌初步诊断大模型医疗系统与医生诊断 AI 助理

摘要

为缓解基层医疗机构肺癌筛查资源短缺的问题，本团队基于飞凌 ELF2 开发板（搭载 RK3588 芯片）首创嵌入式轻量化 AI 诊断系统，借助 ELF2 开发板的高性能，利用 NPU 加速模型推理。团队突破性地采用边缘计算架构，通过专用双模型协同框架（分割模型 lung_rknn 与分类模型 lung_cls_rknn），在无需云端支持的环境下实现 CT 影像的实时解析。系统以 68 秒单次分析速度、≤1.2GB 内存占用的嵌入式算力，完成肺部结节精准定位与良恶性判断，较传统方案功耗可降低 60%，据研究表明，全国医院 CT 从拍片到诊断完成时间约为 7-8h，我们的模型可大大缩短诊断时间，生成的分析网页可辅助医生进行判断，预计可缩短整体时间为原来的 1/6 到 1/8。

作品创新性的人机交互设计通过动态 HTML 界面直观呈现三维肺部模型与可疑结节热力图及其性质判断和详细坐标位置等重要结果，支持医生实时旋转观察病灶细节。本作品可应用于多种医疗场景，例如，在基层医疗场景中，系统 2 分钟内生成结构化报告的特性可有效缓解短缺问题。团队历经构建模型，训练模型，转换模型，构建分析系统，自主建模构建外壳，最终使该设备实现落地应用，旨在为社会提供可复制的嵌入式 AI 医疗范本。

经过多轮实验室测试验证项目可用性后，我们特邀某市级医院影像诊断科医生进行实地测评考察。在访谈中，医生高度认可该项目的实用性与运行效率，并指出相较于传统方法，该项目在检测时间和结果准确度上均取得了突破性进展。

第一部分 作品概述

1.1 功能与特性

本项目旨在响应国家“分级诊疗”与“健康中国 2030”战略需求，针对基层医疗资源短缺问题，基于飞凌 ELF2 开发板（搭载 RK3588 芯片）构建轻量化肺癌 AI 辅助诊断系统。由于主流 YOLO 系列主要处理 2 维图像，3 维 CT 数据通过 YOLO 无法处理，所以我们对神经网络模型结构进行自主构建及创新，通过

创新性地融合边缘计算与医疗大模型技术，自主开发专注于肺癌 CT 影像分析的专用模型，实现了肺部结节自动分割与良恶性预测功能（结节分为良性、恶性两种，恶性结节作为肺癌判断标准）。系统充分发挥 RK3588 芯片的硬件优势，在性能、速度和功耗三个维度实现突破。为提升临床实用性，特别开发了交互式 HTML 界面，可直观展示可疑结节位置信息，辅助医生进行 3D 模型查看。

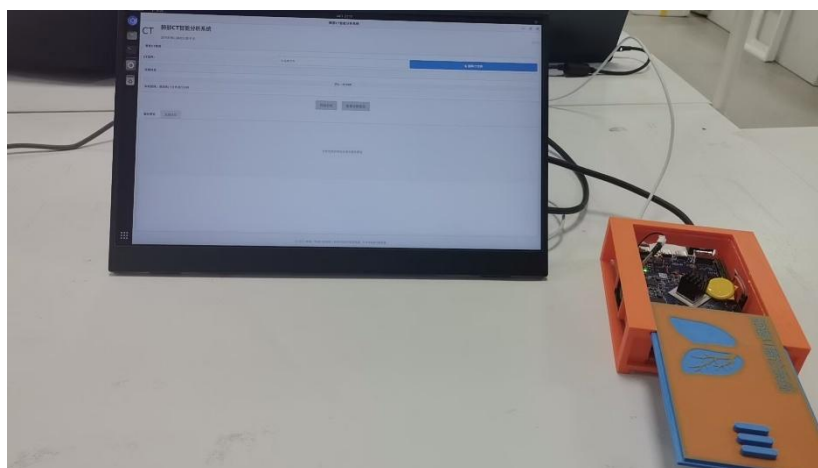


图 1.1 作品整体外观图

1.2 应用领域

（1）基层医疗：支持无网环境 CT 影像实时分析，2 分钟内生成结节定位报告，缓解基层影像诊断科医师不足问题，常见肺部疾病鉴别准确率 $>85\%$ 。

（2）体检初筛：高速处理能力支持单日多例检测，快速标记可疑结节，辅助医生筛查，从而缩短报告生成时间，患者不必多日后来获得报告。

（3）临床随诊：用于术后随访，对比术前术后 CT 影像，量化结节变化。

1.3 主要技术特点

（1）嵌入式 AI 部署：将 ONNX 模型转换为 RKNN 格式并优化，这样能在 RK3588 开发板上实现高性能边缘计算，兼顾了低功耗性与模型的高效推理。

（2）交互式可视化：生成 8 类肺部组织掩码，通过 HTML 网页动态展示结节位置、恶性概率及 3D 模型，支持鼠标旋转与缩放，辅助医生精准诊断。

（3）双模型协同：我们的模型结合深度学习与医学影像处理技术，可同步完成肺部组织分割与结节良恶性分类，从而提升诊断准确率。

1.4 主要性能指标

该模型在肺结节诊断中展现出卓越的综合性能，包括高诊断准确率（如恶性

结节召回率 99.6%、良性结节精确率 99.8%) 和高效性 (单次 CT 分析平均耗时 68.06 秒、峰值内存占用 $\leq 1.2\text{GB}$), 适合嵌入式设备部署。

表 1.1 模型综合性能总览表

评估维度	指标名称	性能值	临床/技术意义
诊断性能	AUC 值	0.911(最优模型)	整体分类能力强
	召回率	恶性结节: 99.6%	漏诊率仅 0.4%
	特异度	良性结节: 98.3%	误诊率仅 1.7%
分类精度	恶性结节精确率	96.7%	假阳性率低
	良性结节精确率	99.8%	误判风险低
	恶性结节 F1-score	0.982	综合性能优异
效率指标	单次 CT 分析平均耗时	68.06 秒	显著快于云端方案
	峰值内存占用	$\leq 1.2\text{GB}$	适配嵌入式设备

1.5 主要创新点

(1) 轻量化双模型架构: 我们的模型采用分割-分类解耦设计, 可以优化资源分配并提升诊断可靠性, 相比主流端到端模型显著降低计算量和内存消耗。

(2) 边缘计算部署: 将分割 (lung.rknn) 和分类 (lung_cls.rknn) 模型部署 ELF2 开发板, 从而实现低功耗、低成本的嵌入式 AI 诊断并减少云端依赖。

(3) 人机交互创新: 板端运行程序后, 可通过 HTML 网页提供可旋转、缩放的 3D 肺部模型, 并呈现多方面的其他关于可疑结节的详细信息, 辅助医生直观查看结节细节。

1.6 设计流程

该流程图 (图 1.2) 展示了系统的软硬件协同开发闭环: 软件设计 (大模型构建→训练→格式转换→结合数据构建分析系统→生成网页报告) 与硬件设计 (模型部署→机械结构→外壳设计→3D 打印) 并行推进。

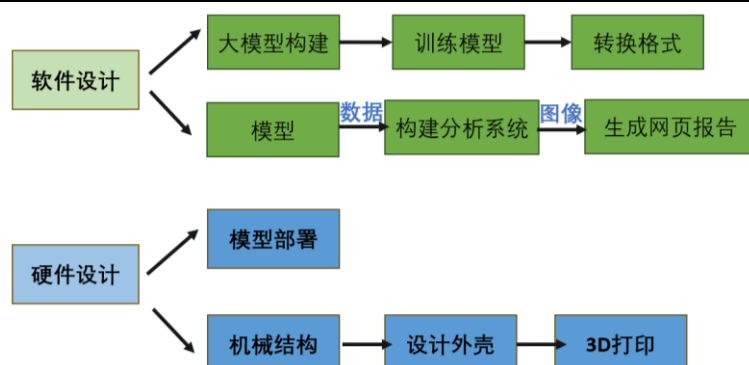


图 1.2 设计流程图

第二部分 系统组成及功能说明

2.1 整体介绍

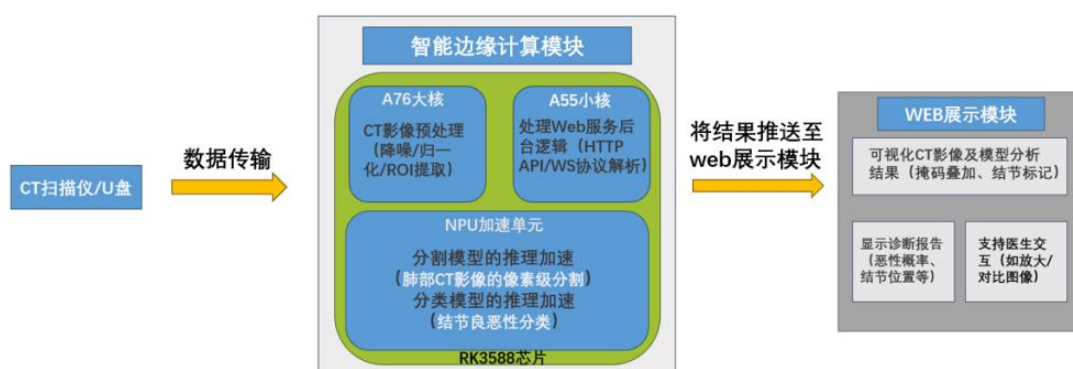


图 2.1 系统整体架构

我们的系统以飞凌嵌入式 ELF2 开发板为核心，搭载高性能 RK3588 芯片，构建了从影像采集到智能分析的完整闭环。系统通过 CT 扫描仪或 U 盘接入板端接口传输数据（[图 2.1 左侧部分](#)），开发板内置的 NPU 单元并行执行深度学习模型——先由分割模型精准提取肺部组织与结节区域，再经分类模型计算恶性概率（[图 2.1 中间部分](#)），最终通过 HDMI 接口将分析结果输出至医疗显示屏。医生可通过网页端交互界面实时查看 CT 影像与 AI 标注的叠加效果，动态调整窗宽窗位、旋转三维视图，并导出结构化诊断报告（[图 2.1 右侧部分](#)）。

2.2 硬件系统介绍

2.2.1 硬件整体介绍

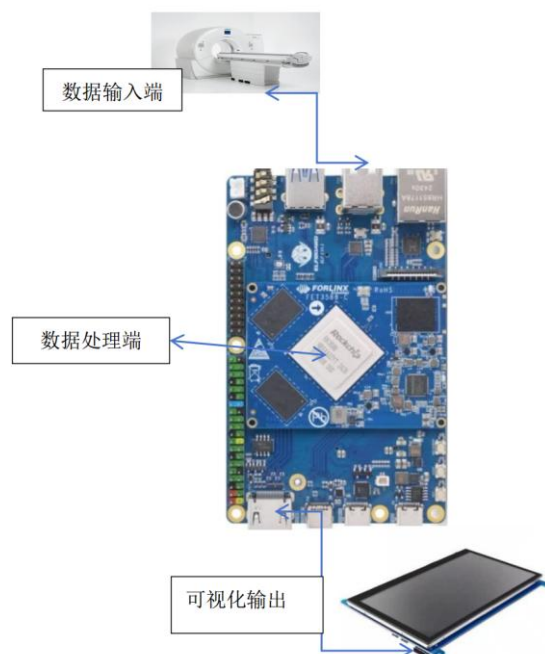


图 2.2 硬件部分整体介绍图

硬件部分以 RK3588 为核心，主要包括数据输入端（这里用 U 盘表示，数据为 CT 机输出的 mhd 格式的文件）、数据处理端和可视化输出三部分。输入的数据经板载 RKNN 模型识别，经板端 NPU 计算处理后进行可视化输出（数据处理端），如外接屏幕。

2.2.2 机械设计介绍

为了兼顾散热且保护开发板，我们采取 3D 打印的方式构建符合开发板形状的外壳，并且预留接口部分和风扇部分空隙。

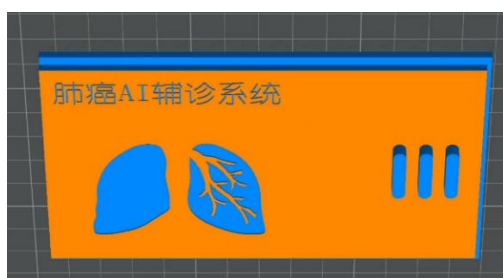


图 2.3 外壳盖设计图

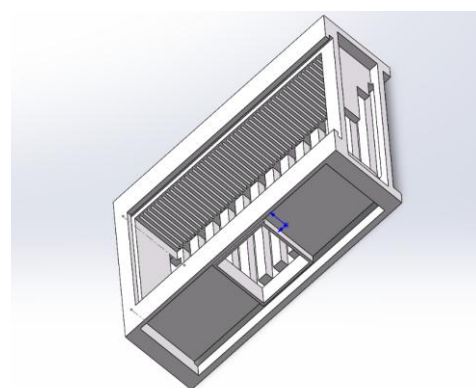


图 2.4 外壳盒设计图

2.2.3 电路各模块介绍

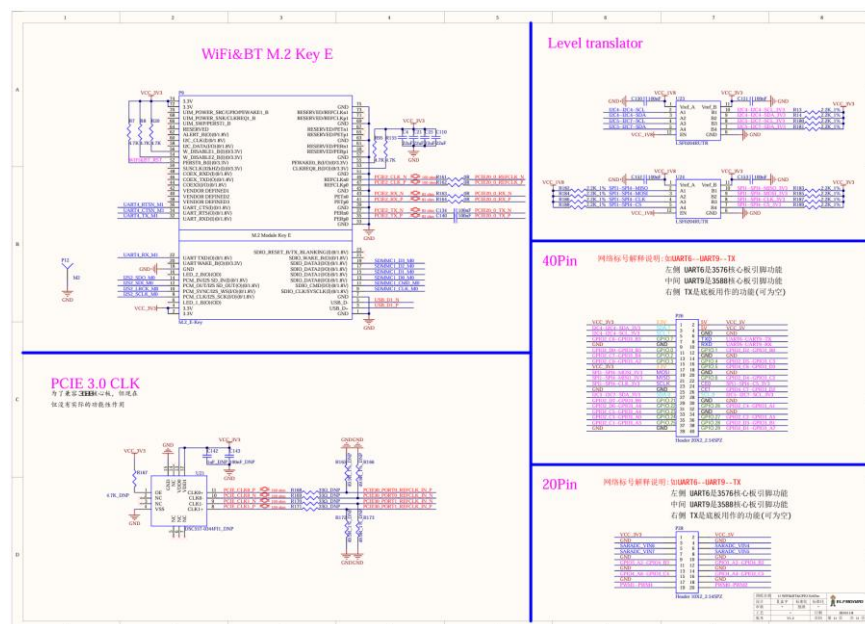


图 2.5 ELF2 开发板部分原理图

2.3 软件系统介绍

2.3.1 软件整体介绍

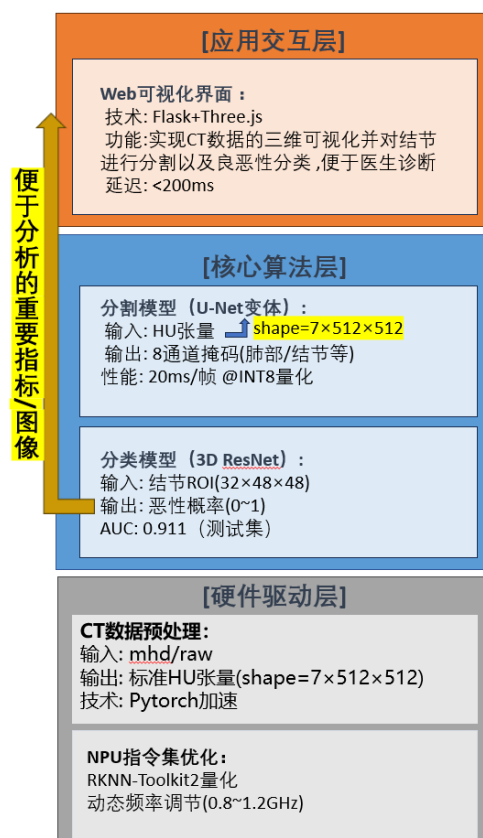


图 2.6 软件系统整体框图

本系统软件部分采用三层核心架构，包括边缘计算引擎、Web 可视化平台和硬件驱动层，基于 HTTP 协议实现模块间通信。系统通过飞凌 ELF2 开发板的嵌入式 Linux 环境直接处理 CT 影像，并采用 Flask+Three.js 构建交互式分析界面，实现医生终端与边缘计算设备的对接。所有计算均在本地完成，确保医疗数据隐私与实时性。

2.3.2 软件各模块介绍

我们项目的软件部分分为三个模块：Web 可视化平台模块（基于 Web 的交互式医疗看板，直观呈现肺癌 CT 的 AI 分析结果和重要影像）、边缘计算引擎模块（部署在 RK3588 板载的实时 AI 处理器，在设备端直接完成 CT 影像的肺癌结节检测与分析）和硬件驱动层模块（连接 CT 设备的底层桥梁，实现医疗影像的高速采集、预处理和安全传输至计算引擎）。

(1) Web 可视化平台模块

Web 可视化平台模块总体框图（图 2.7）。



图 2.7 Web 可视化平台模块框图

Web 可视化平台模块流程图及主要函数流程图（图 2.8）。

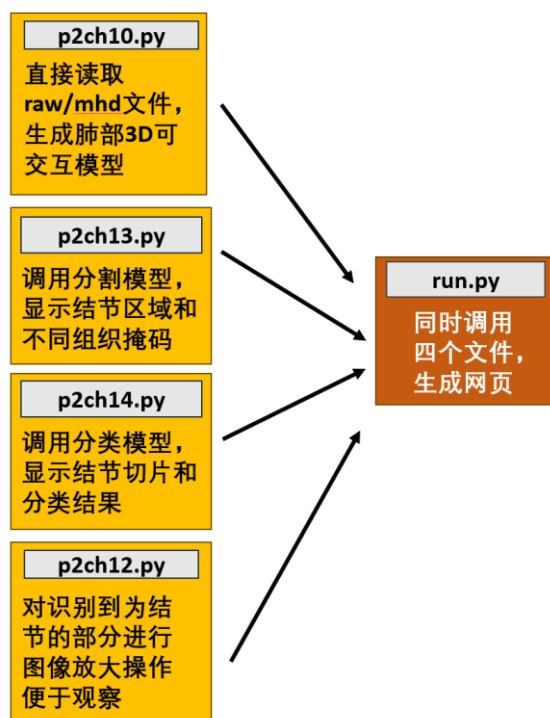


图 2.8 Web 可视化平台模块流程图

(2) 边缘计算引擎模块

边缘计算引擎模块总体框图（图 2.9）。

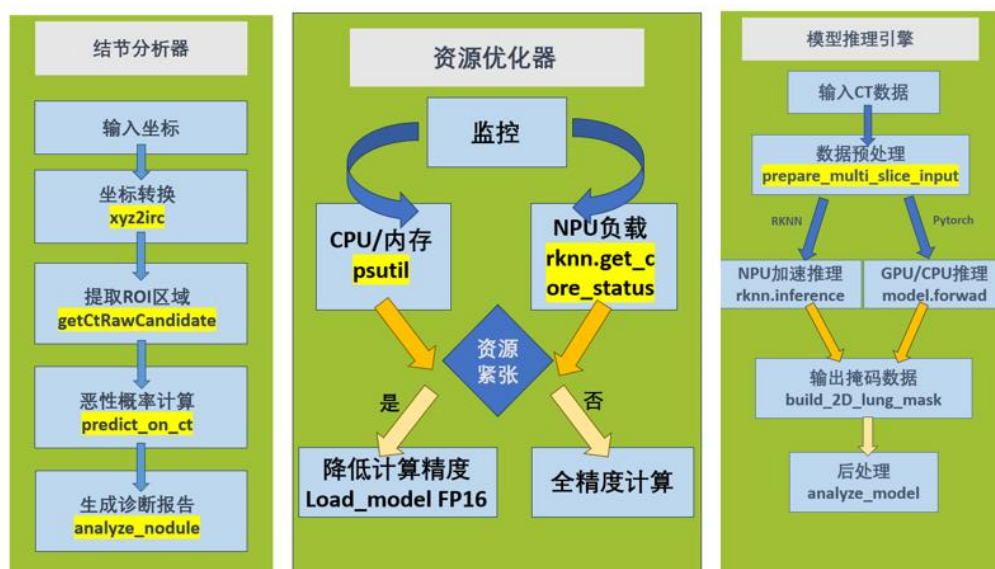


图 2.9 边缘计算引擎模块框图

边缘计算引擎模块流程图（图 2.10）。

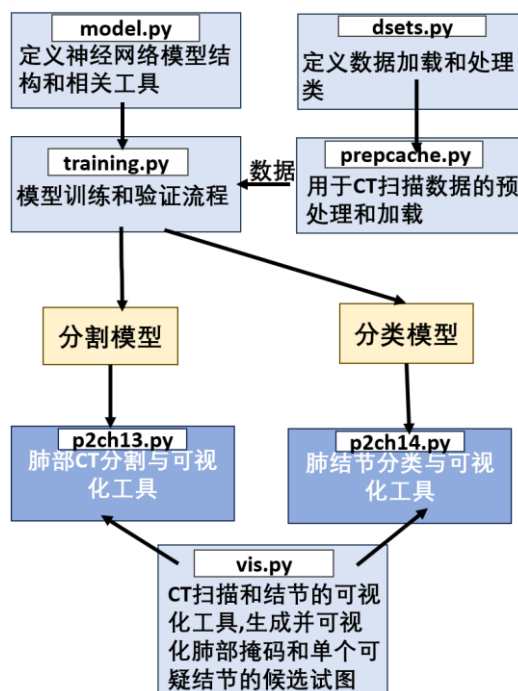


图 2.10 边缘计算引擎模块流程图

（3）硬件驱动层模块

硬件驱动层模块总体框图（图 2.11）。

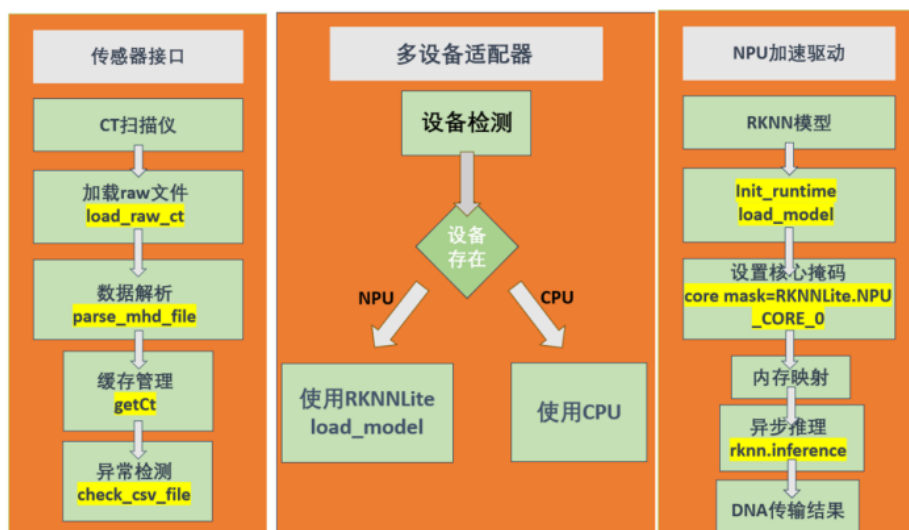


图 2.11 硬件驱动层模块框图

硬件驱动模块流程图及主要函数流程图（图 2.12）。

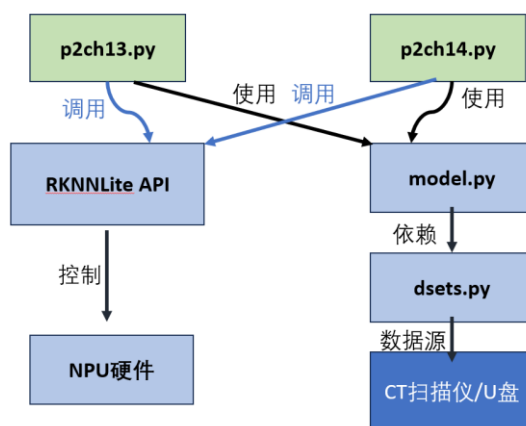


图 2.12 硬件驱动模块流程图

第三部分 完成情况及性能参数

3.1 整体介绍

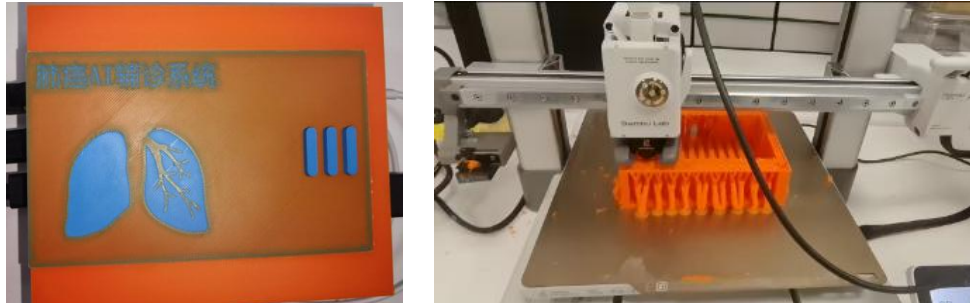
如图 3.1 所示，屏幕部分显示呈现给医生的 HTML 界面，无线鼠标便于医生滚动鼠标放大查看图片且旋转 3D 模型观察病灶细节，3D 打印的外壳设计契合开发板外端并兼顾了散热考虑。



图 3.1 作品整体外观图

3.2 工程成果

3.2.1 机械成果



3.2 机械成果图

3.2.2 电路成果

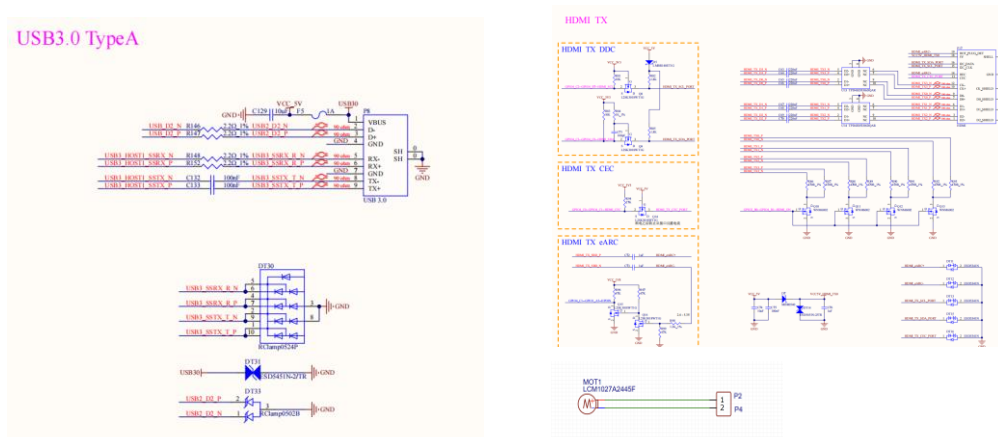


图 3.3 开发板接口原理图及外接电路图

3.2.3 软件成果

我们采用自主设计的神经网络，并且采用的流程是先进进行数据预处理-训练的训练流程，使得训练加速以及模型推理加速（报告仅展示部分代码）

```

@staticmethod
def train(self, model, self.args.model):
    model = model.cuda()
    if self.args.device:
        device = self.args.device
        torch.cuda.set_device(device)
    else:
        device = torch.device('cuda:0')
    torch.cuda.set_device(device)
    optimizer = optim.Adam(model.parameters())
    scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, 'min')
    for epoch in range(1, self.args.epochs + 1):
        train_loss, train_acc = self.train_epoch(model, optimizer, scheduler)
        val_loss, val_acc = self.val_epoch(model)
        scheduler.step(val_loss)
        if self.args.save_model:
            self.save_model(model, optimizer, scheduler, epoch)
    return model, optimizer, scheduler

def train_epoch(self, model, optimizer, scheduler):
    model.train()
    train_loss = 0
    train_acc = 0
    for i, data in enumerate(self.train_loader):
        optimizer.zero_grad()
        outputs = model(data)
        loss = self.criterion(outputs, data['label'])
        loss.backward()
        optimizer.step()
        train_loss += loss.item()
        train_acc += outputs.argmax(1).eq(data['label']).sum().item()
    train_loss /= len(self.train_loader)
    train_acc /= len(self.train_loader)
    return train_loss, train_acc

def val_epoch(self, model):
    model.eval()
    val_loss = 0
    val_acc = 0
    for i, data in enumerate(self.val_loader):
        outputs = model(data)
        loss = self.criterion(outputs, data['label'])
        val_loss += loss.item()
        val_acc += outputs.argmax(1).eq(data['label']).sum().item()
    val_loss /= len(self.val_loader)
    val_acc /= len(self.val_loader)
    return val_loss, val_acc

def save_model(self, model, optimizer, scheduler, epoch):
    torch.save(model.state_dict(), self.args.save_model_path)
    torch.save(optimizer.state_dict(), self.args.save_optimizer_path)
    torch.save(scheduler.state_dict(), self.args.save_scheduler_path)

```

图 3.4 部分主要函数代码图

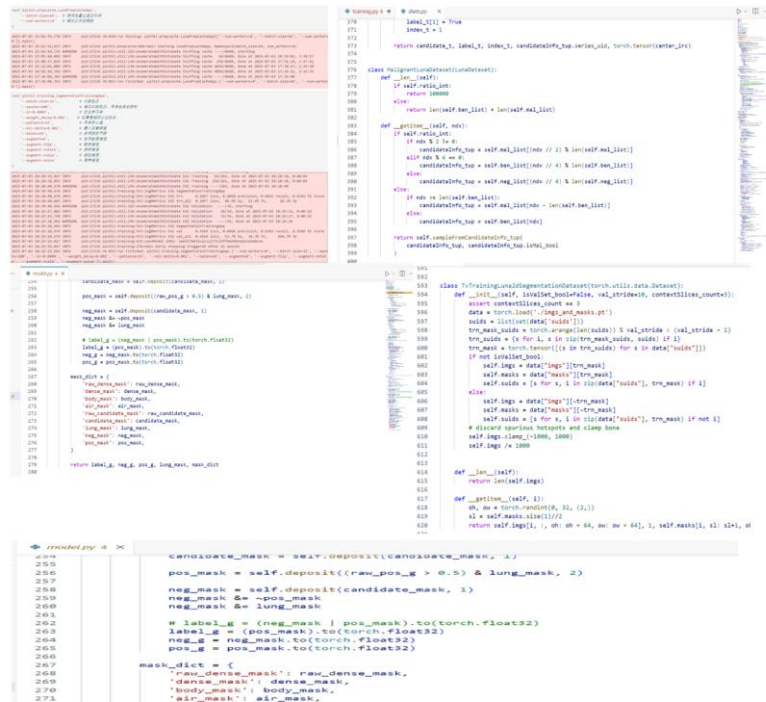


图 3.5 部分重要模型文件

3.3 特性成果

从图 3.6 至图 3.12，展示的是我们的模型在板端分析处理完图像后，通过设计的 HTML 网页展现出的可交互的可视化界面：其中包含 3D 肺部可视化、详细输出可疑结节坐标位置及其良恶性、数据增强处理可视化界面、肺部结节分割分析可视化界面、详细切片分析、结节分类与预测可视化界面等多种功能。



图 3.6 UI 界面

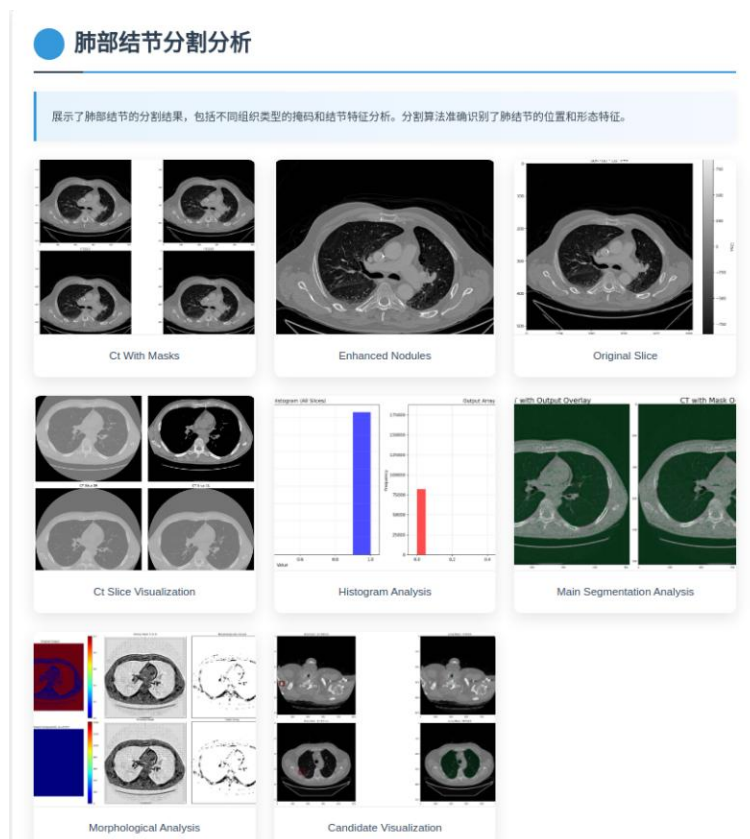


图 3.7 肺部结节分割分析可视化界面

详细切片分析

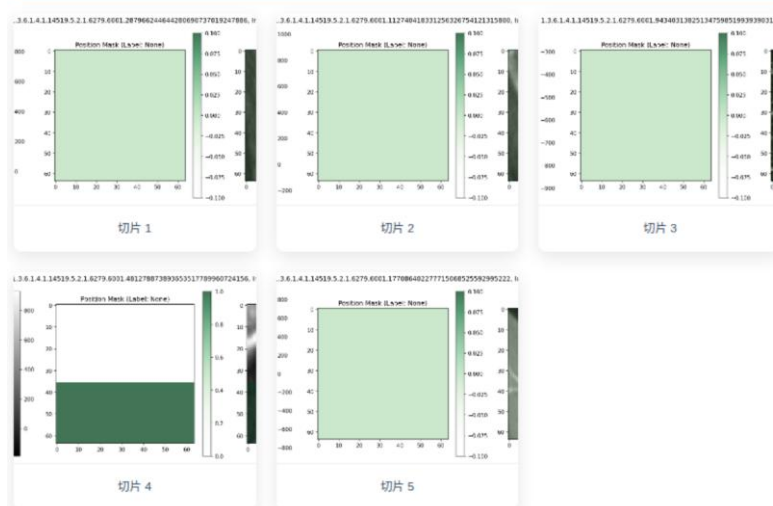


图 3.8 详细切片分析可视化界面

肺部结节详细信息

以下是检测到的肺部结节的详细信息：

结节ID:	1.3.6.1.4.1.14519.5.2.1.6279.6001.104562737760173137525888934217
X坐标:	-124.8342624 mm
Y坐标:	127.24715459999999 mm
Z坐标:	-473.0644785 mm
直径:	10.46585391 mm
性质:	恶性

图 3.9 肺部结节详细信息可视化界面

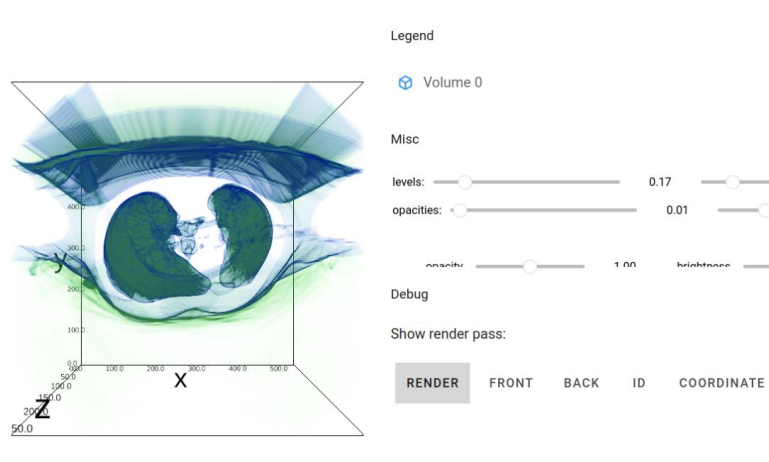
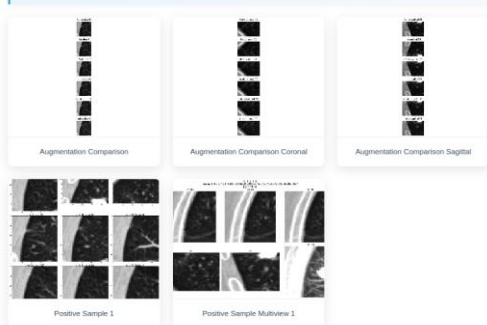


图 3.10 3D 肺部可视化界面

数据探索与增强处理

展示了不同数据增强技术对CT图像的影响，包括翻转、旋转、噪声添加等。这些技术用于提高模型的泛化能力，增强对肺部结节的检测准确性。



结节分类与预测

展示了结节分类模型的预测结果，包括良性恶性概率可视化。模型基于深度学习算法，对检测到的结节进行了分类评估。

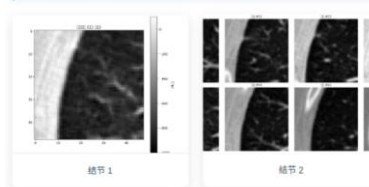


图 3.11 数据探索与增强处理可视化界面 图 3.12 结节分类与预测可视化界面

第四部分 总结

4.1 可扩展之处

- (1) 改动模型使其能分析多模态的数据，更具普适性。
- (2) 在此肺癌 AI 诊断系统上增加其他的肺部疾病识别功能。
- (3) 增加多屏输出功能，支持医生多视图对比（如原始 CT、分割掩码、热力图三屏同步）。
- (4) 进行联网，将医生认为准确无误的网页给到患者，得到更清楚的报告。

4.2 心得体会

三个大一新生，初出茅庐。这段旅程，从项目开始只有一个模糊的规划，再到后来的逐步落实和收尾，每个阶段对新手的我们来说，都是挑战。在模型构建和训练阶段，由于这个方向资料本身就少，再加上语言的限制，负责该部分的队友需要花费大量的时间阅读和学习、在一个多月的努力下终于调试成功。然而，由于 RK3588 芯片所支持的模型结构是规定好的 RKNN 格式，所以模型转换部分也是一大难题：我们是自主构建神经网络，并非 YOLO 系列那样更容易被转换的格式，所以只能在一次次试错中寻求真解。还记得团队花费了近一周的日夜，总共换代五十多个版本，终于将模型成功转换。接下来的收尾工作也困难重重：首先 3D 打印外壳阶段，由于雨季气候潮湿，材料受到的影响大，因此常有翘边的情况，而且由于是初学 3D 打印，经常遗漏一些实际问题，所以最后改了五六个版本。写报告和拍摄视频，对于初次参加比赛的我们来说，也只能一路摸索。走到现在，除了感慨便是对队友们的感激。“万人操弓，共射一招，招无不中。”正是因为我们之间的配合与默契，才能有惊无险的度过种种考验。也感谢瑞芯微官方开设的交流群里官方技术人员和其他参赛同袍的答疑解惑，正是因为这个社区的温暖和 ELF2 开发板的高性能支持，我们的项目才能顺利落地，也祝瑞芯微社区越来越好！

第五部分 参考文献

- [1] 史蒂文斯 E., 安蒂加 L., 菲曼 T. (2022). PyTorch 深度学习实战 (牟大恩, 译). 北京: 人民邮电出版社.
- [2] 邹瑜, 孔德香, 汪瑶, 等. 精益管理对改善 CT 和 MRI 检查服务流程的应用效果[J]. 医疗装备, 2023, 36(11): 38-45.
- [3] Mouli, C., Satish, M. (2023). Lung Cancer Detection using Machine Learning and Deep Learning Algorithms. *International Journal of Research Publication and Reviews*, 4(12): 4757-4763.
- [4] Idowu, E., Doris, L. (2024). Transfer Learning in Lung Cancer Detection: Leveraging Pre-trained Models for Improved Performance.
- [5] Chu, P. T. M., Ha, T. P. B., Vu, N. M., Ha, H., Doan, T. M. (2024). The application of deep learning in lung cancerous lesion detection. *medRxiv preprint*.
doi: <https://doi.org/10.1101/2024.04.12.24305708>