

---

## 基于 RK3588 芯片的智能竹蛭分练系统 与高速机器视觉优化

### 摘要

面向水产养殖和加工场景中的蛭子自动化分选需求，团队设计并实现了基于 RK3588 高性能 SoC 与高速机器视觉的智能竹蛭分练系统。系统以 SoC 为主控核心，结合多通道视频采集，硬件解码器，硬件放缩器，神经网络计算单元，使用 ResNet50v2 深度神经网络模型对传送带上的竹蛭进行三分类。识别结果通过 RS422 发送至下位机 MCU，MCU 通过 CAN 总线驱动步进电机进行实时分拣。同时，系统还通过 PID 算法调控传送带的运行速度，实现高速且稳定的物料输运及分拣。本系统集成嵌入式 AI 推理、实时控制与多机协作等技术，具有处理速度快、识别精度高、系统稳定性强等优势，可广泛应用于水产品分拣领域，提升分拣效率与产品质量。

## 第一部分 作品概述

### 1.1 功能与特性

#### 1. 高效硬件视频解码与放缩

利用 RK3588 的高性能 MPP+RGA，解码网络摄像头的 h264 流，并使用 RGA 将视频帧从硬件帧转换为软件帧，此外还可以零拷贝将图片进行放缩，以便进行更加复杂的操作，在解码 1080P 30 帧的 h264 流情况下，CPU 占用仅为 30%/800%，充分发挥 RK3588 硬件资源优势。

#### 2. 高性能图像处理

充分利用强大的 6TOP 三核 NPU，提前训练好 ResNet 残差网络模型，并对模型进行量化，无需云服务器与高算力，在端侧即可实现实时性的图片处理和识别，INT8 量化的 ResNet 推理速度可达 1s 中 60 次以上。

#### 3. 全自动蛭子筛选

通过 RK3588 实现蛭子大小，是否合格的全自动识别，无需人为操作，实现蛭子筛选行业的无人化，智能化，未来化。

#### 4. 稳定总线通信

使用 RS422 实现 RK3588 与下位机之间的高可靠、抗干扰通信，确保控制数据的稳定传输，同时使用 CAN 总线进行步进电机控制，确保通信稳定，及时。

#### 5. 图像识别光强自适应

通过自制包含不同光照情况的各种类蛭子，模型可训练出克服光照明亮不同导致识别出现偏差的困难，以便能够更好的满足实际工业应用场景的变化环境。

### 1.2 应用领域

#### (1) 海鲜智能分拣流水线

在大型水产批发市场、港口冷链加工企业中，水产品需在短时间内完成分级、打包、入库等操作，传统手工分拣已难以满足高速周转需求。本系统可与冷链输

送线集成，实现高速连续分拣，支持 7×24 小时长时间运行。图像识别可在不同光照环境下进行，适应冷库或露天加工车间。

## **（2）水产养殖加工**

本系统最核心的应用场景是水产行业中对贝类（如竹蛏、蛤蜊、花蛤、扇贝等）进行尺寸与品类分级筛选。在传统水产养殖及加工企业中，分选作业通常依靠人工进行，不仅效率低、误差大，而且长期人工操作劳动强度高、成本高。本系统通过摄像头+AI 模型对竹蛏进行高精度识别，结合挡板分拣结构，替代人工判断与操作，实现流水线上连续、稳定的自动分级处理。该方案可根据实际产品替换模型与机械参数，快速适应不同贝类或软体海鲜的分拣任务，有效提高产能与标准化水平。

## **（3）科研与教学实验平台**

作为一个集图像处理、AI 推理、嵌入式控制、通信协议、机电一体化为一体的综合性系统，该平台非常适合用于高校相关专业（如自动化、人工智能、嵌入式系统、计算机视觉等）的课程教学与实验研究。学生可以深入理解从数据采集、模型训练、边缘部署、通信控制到执行动作的完整流程，也可以在现有框架上开发新的识别模型或控制策略。平台的开源性与模块化结构也为科研课题提供了良好的拓展基础。

# **1.3 主要技术特点**

## **（1）基于深度学习的图像识别**

本项目的核心 AI 部分使用 ResNet50v2 模型进行图像分类。该模型具备较深的网络结构和残差连接机制，具有优秀的特征提取能力和分类准确性。模型通过大量竹蛏图片进行有监督训练，最终实现对“大”、“小”、“杂质”三类目标的准确识别。

图像识别由 RK3588 内置 NPU 在边缘本地完成，确保识别过程低延迟、高可靠。推理结果直接作为控制指令依据，驱动下位机进行分拣动作，构成完整的“视觉识别 + 物理控制”闭环。

## (2) 多机协同控制系统

系统采用双处理器架构,其中 RK3588 负责 AI 计算与高级调度,STM32F407 作为下位机专注于执行实时控制任务,包括:控制传送带电机的转速(通过 PWM 调速);控制步进电机驱动挡板实现快速分拣;执行主控下发的动作指令,响应快、实时性强。双处理器通过分工协作实现“软硬解耦”:RK3588 聚焦图像处理与 AI 推理,STM32 专注于物理层面电机控制,提高了系统可靠性和响应效率。

## (3) 传送带+挡板机构实时联动控制

系统根据 AI 识别结果预测目标竹蛭将到达分拣挡板的具体时间点,STM32 根据该延迟时间控制步进电机精确摆动挡板,实现高精度物理分拣。同时,为避免物料堆叠与识别延误,系统支持对传送带速度进行动态调整,使图像识别与分拣动作始终保持同步。这种“视觉-时序-机械”多级联动机制,极大提高了分拣准确率和整体处理速度。

## 1.4 主要性能指标

表 1 性能指标

指标	参数	说明
识别距离	20-40 厘米	取决于摄像头距传送带距离
识别速率	30 次/秒	现阶段以实现 30 次每秒
模型推理极限速度	60 次/秒	Int8 模型的极限推理速度
准确率	$\geq 90\%$	模型对蛭子分类正确率高

## 1.5 主要创新点

### (1) 图像识别与物理控制间的时间协同机制设计

在传统的机械控制系统中,图像识别结果难以精准映射到物理装置上的分拣动作,常出现延迟与识别失配问题。本系统创新性地引入“识别延迟补偿机制”,通过传送带速度检测与光电门检测物体到来时间,可做出延迟补偿,实现图像识别与挡板摆动之间的精确时序对齐,使分拣控制动作与目标物体位置保持一致,

极大提升分拣准确性与效率，确保即使在高速传送条件下也可稳定运行。

## (2) 异构计算架构创新

采用"RK3588+下位机"的异构计算架构，充分发挥 RK3588 的 AI 算力与下位机的实时控制优势。通过任务分级处理（视觉计算在上位机、实时控制在下位机）实现计算资源的最优分配

### 1.6 设计流程

1. 数据准备与模型训练：收集不同批次、不同光照、不同拍摄角度下的竹蛭图像，并通过随机裁剪、旋转、镜像、亮度/对比度扰动等，增强模型鲁棒性。使用 ResNet50v2 神经网络模型进行训练，采用 Adam 优化器，损失函数为交叉熵函数。训练完成后量化为 rknn 模型文件。

2. 视频解码与视频处理：调用 rk3588 板端的视频解码器与放缩器，为 QT 显示与模型推理打下铺垫。

3. 边缘部署与集成：NPU 加载训练好且量化后的 int8 rknn 模型，本地摄像头通过调用 RKMPP 的 mpi 库对视频作硬解码处理，传回图像经调用硬件图像处理库 RGA 进行零拷贝缩放后送入 NPU 推理。

4. 控制挡板分类：RK3588 得到 NPU 的推理结果后，根据光电门触发的时间信号和获取传送带的转动速度来预测蛭子到达分拣挡板的延迟时间，RK3588 使用 RS422 通信发送分拣指令给 STM32 后，STM32 根据延迟时间来择时通过 CAN 总线控制步进电机旋转挡板对蛭子进行分类分拣。

## 第二部分 系统组成及功能说明

### 2.1 整体介绍

本系统采用上下位机协同架构实现工业自动化场景中的智能检测与精准运动控制。系统以 RK3588 作为核心处理单元，负责摄像头采集的视觉数据分析和 AI 模型推理，并通过 RS422 串行通信协议与下位机进行数据交互。下位机作为实时控制单元，接收来自光电传感器的位置检测信号和工控按钮的指令输入，同时通过 PID 闭环算法精确控制直流电机转速，并利用 CAN 总线协议驱动步进电机完成高精度定位。系统工作流程中，RK3588 首先对摄像头输入的图像进行目标检测或缺陷识别等 AI 推理运算，然后将处理结果传输至下位机，由下位机根据传感器反馈实时调节电机运动，形成完整的感知-决策-执行闭环。该架构兼具视觉处理能力和实时控制特性，通过 RS422 和 CAN 总线等工业级通信协议确保系统可靠性，实现高效、精准的自动化蛭子筛选分类。

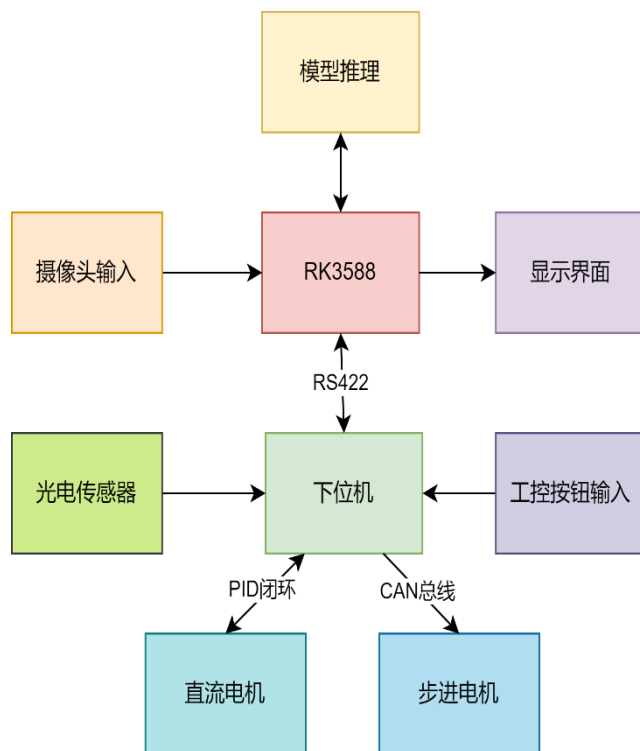


图 1

### 2.2 硬件系统介绍

#### 2.2.1 硬件整体介绍

本项目硬件系统采用模块化设计，基于铝型材与桦木底板构建稳定的传送带结构，集成步进电机、直流电机和可调节相机平台，具备良好的兼容性与减震性能。电源系统采用多级 Buck 降压方案，提供高效稳定的 12V 与 5V 输出，整体



结构紧凑、安装灵活，适用于工业自动化控制与视觉检测场景。

### 2.2.2 机械设计介绍

本项目面向工业自动化场景，设计了一套基于模块化理念的传送带控制与检测系统。系统结构以传送带为核心，采用侧板铝型材和桦木底板作为机械支撑平台，实现整体坚固可靠、装配灵活的特点。模块化设计使得系统各部分（包括驱动机构、相机平台、传感器等）均可快速拆卸与重新配置，便于维护与功能扩展。

在动力部分，系统集成了步进电机和直流电机，其中直流电机通过定制的梅花联轴器与传送带轴体连接。该联轴器具备良好的减震和缓冲能力，有效提升了传动系统的稳定性与耐用性。步进电机和相机平台均固定在铝型材框架上，平台高度和光电门安装位置均可调节，适应不同规格的检测需求，从而增强系统的适配性和兼容性。

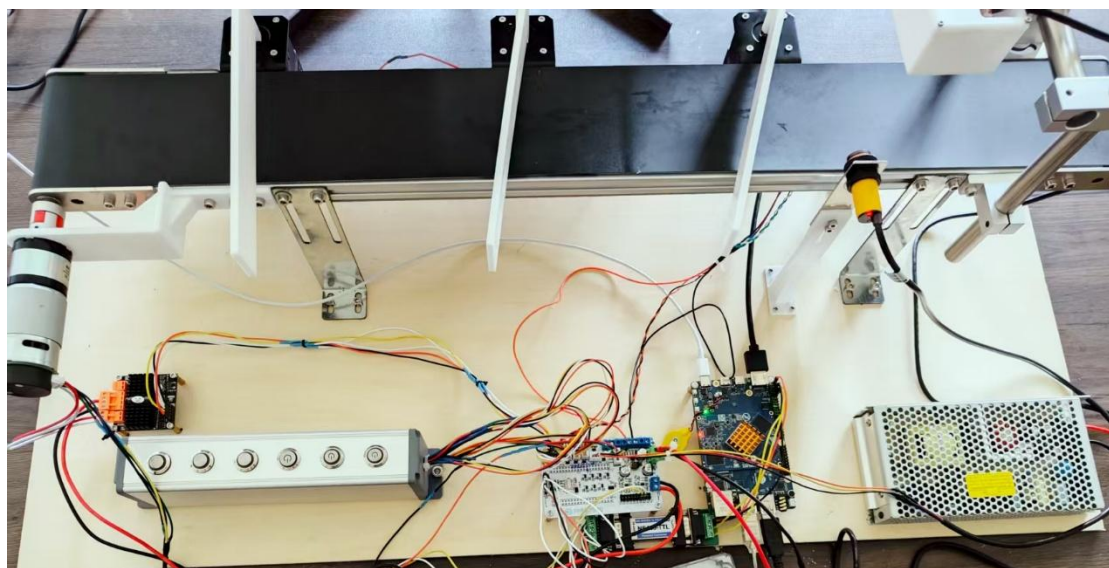


图 2

### 2.2.3 电路模块介绍

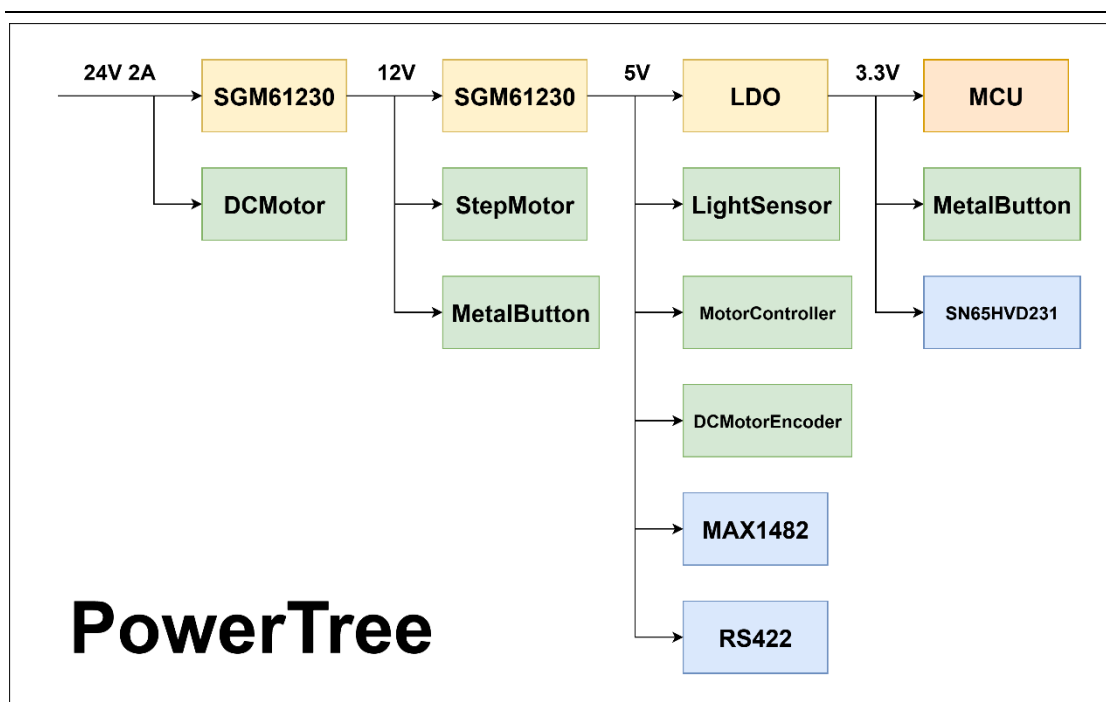


图 3

控制系统采用下位机集中供电方案，如电源树结构图所示。系统通过多级 Buck 降压电路将输入电压有效转换为 12V 与 5V 两级输出，分别用于驱动电机、传感器、相机等核心模块。电源设计确保每级输出电压保持在输入的一半左右，运行在芯片效率最高的“甜点区”，有效提升系统供电效率与稳定性。具体的电源管理与电路实现详见图 4、图 5



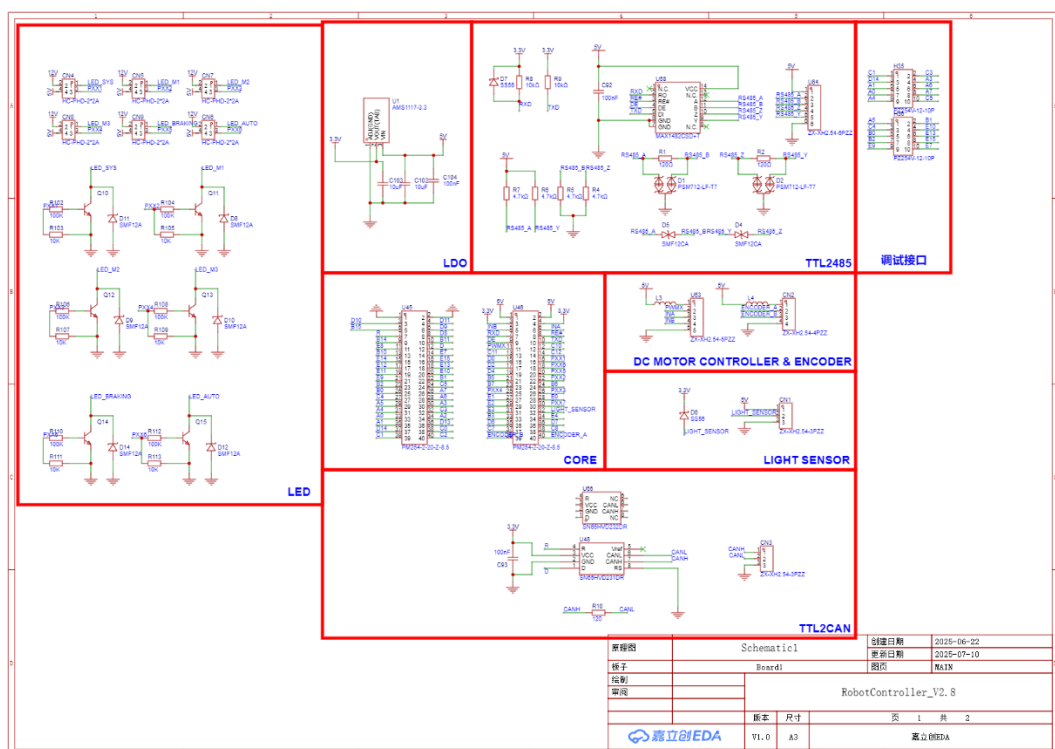


图 4

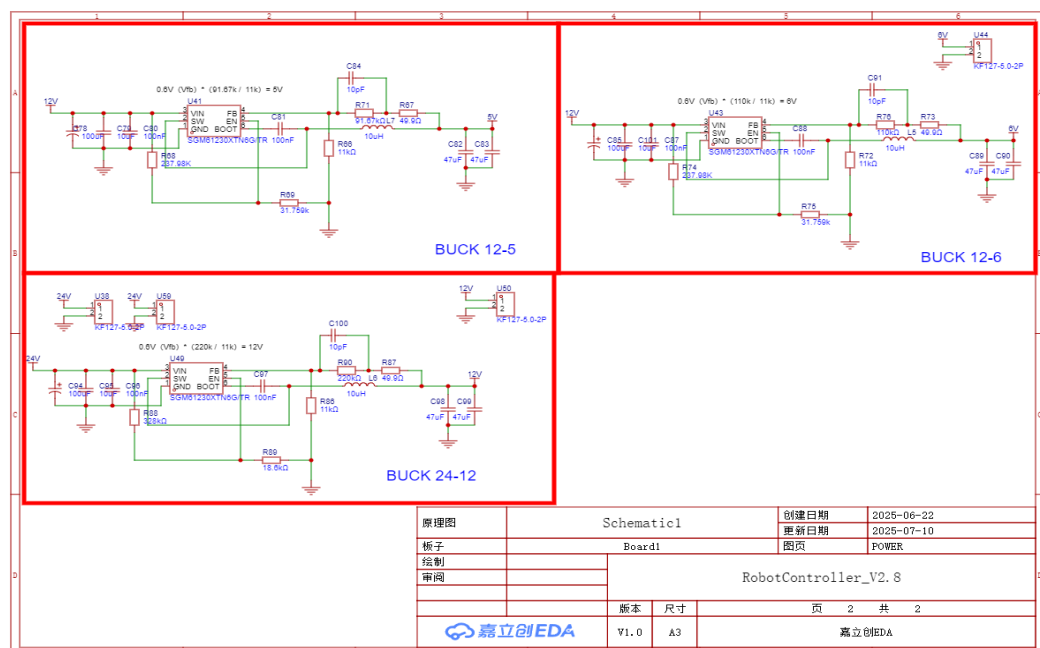
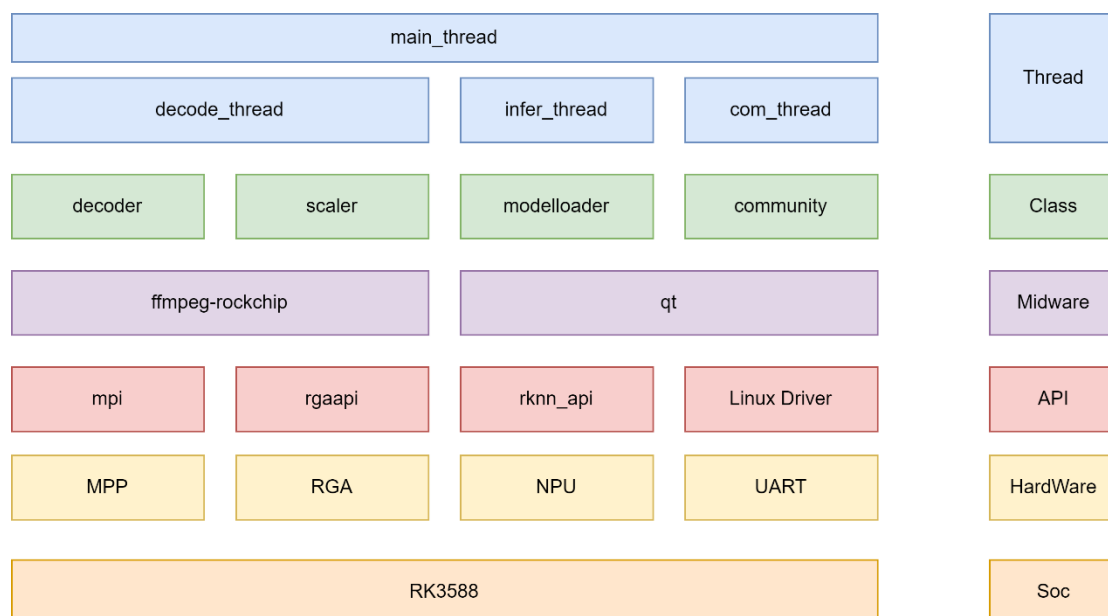


图 5

板级电路最终效果图如图 6 所示。

## 2.3 软件系统介绍

### 2.3.1 软件整体介绍



10

本项目使用 ffmpeg-rockchip（对 RK 芯片特殊支持的 ffmpeg，底层是直接操作 MPI 和 RGAapi），MPI 和 RGAapi 利用 RKMP 对摄像头视频进行硬解码，使用 RGA 对解码后的图片预处理，预处理后的图片输入 NPU 加载的 int8 量化后的 rknn 模型，将模型的输出进行整合，等待下位机请求，发送识别结果。

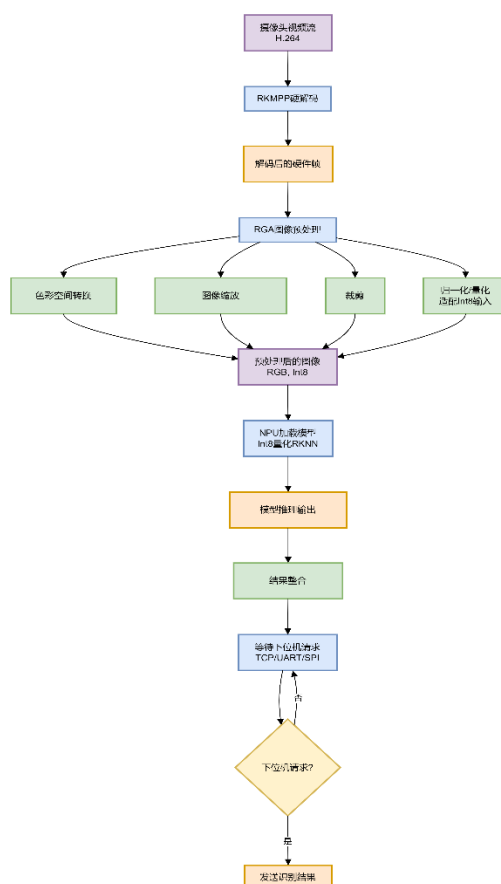


图 8

## 2.3.2 软件模块介绍

### （1）视频解码模块设计

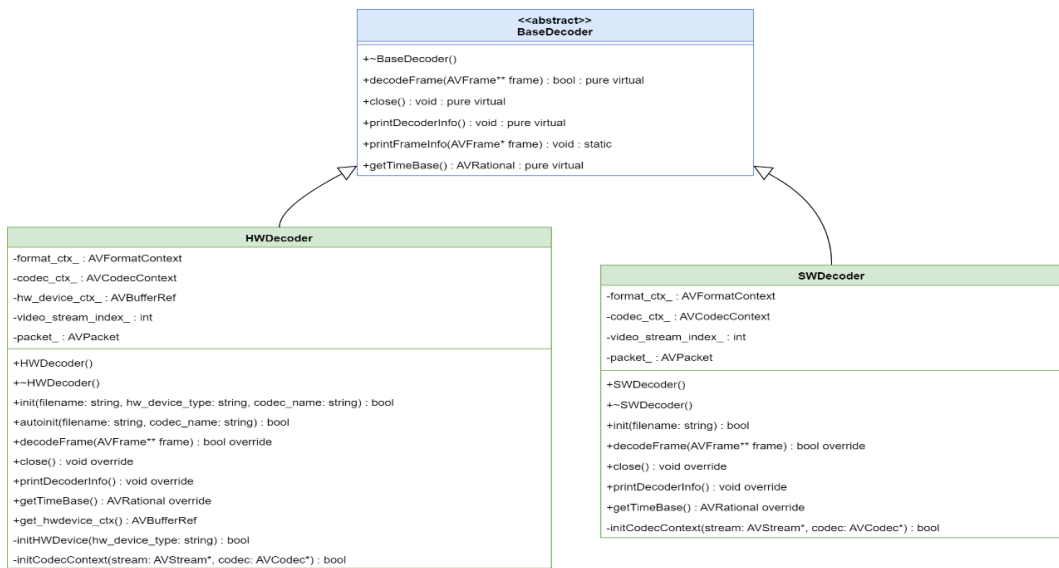


图 9

我们设计基类，并让软件解码器和硬件解码器继承这个类，他们的底层实现均为 ffmpeg，使用硬件解码器时，需要设置硬件解码器和硬件加速器，以 RK3588 为例，我们使用 mpp 作为加速器，使用 h264\_rkmpp 作为硬件解码器。

## (2) 视频放缩模块设计

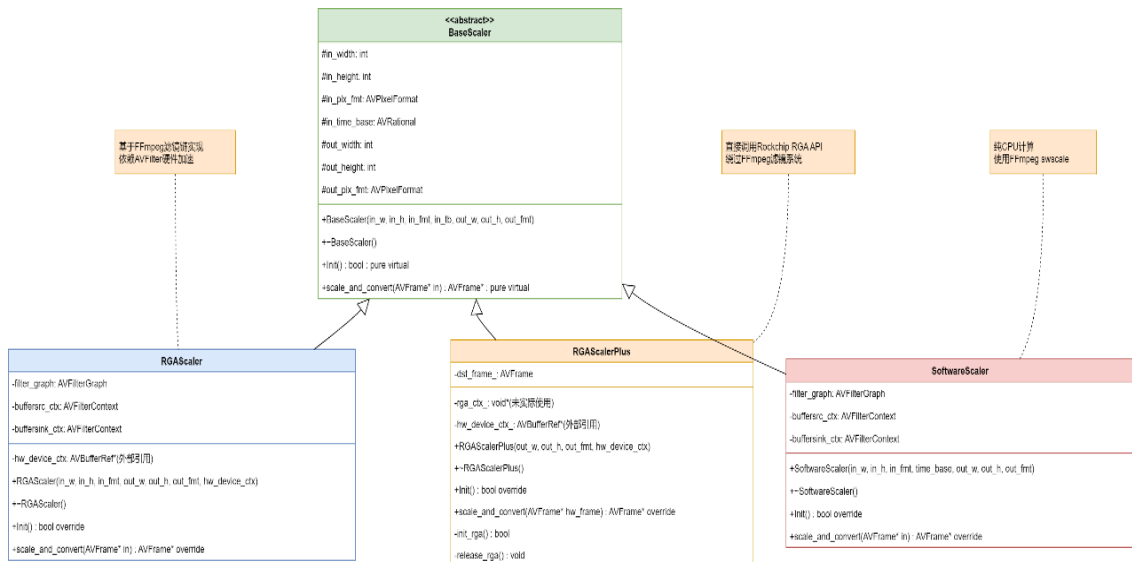


图 10

我们设计基类放缩器，并让软件放缩器和 RGA 放缩器继承这个类，设计软件放缩器保证其通用性，设计基于 ffmpeg-rockchip 接口的 RGAScaler，进行 RGA 硬件放缩，尽管对于 ffmpeg-rockchip 已经实现了命令行零拷贝，但是基于其 C

库，由于 ffmpeg 接口的限制，我们较难实现直接将显存中的数据直接进行格式转换+放缩，RGAScaler 的实现为将软件帧上传到 mpp，然后进行硬件放缩，而我们通过 mpp 解码后，帧已经在显存中了，这样就造成了两次无意义的拷贝，因此，团队便设计了 RGAScalerPlus，直接操作 RGAapi，减少这两次不必要的拷贝，实现格式转换+放缩，通过保留 ffmpeg 的输入接口。

### (3) 模型加载与推理模块设计

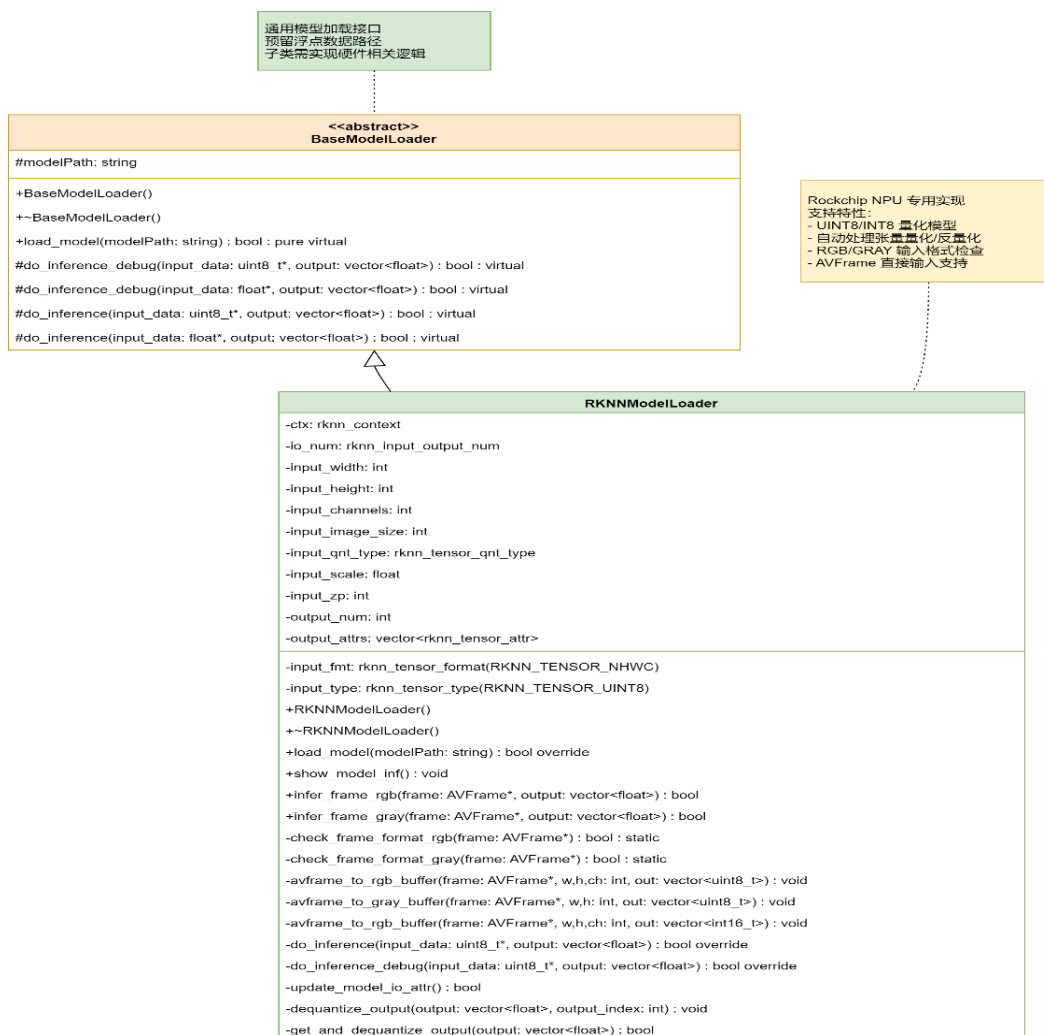


图 11

我们使用 rknnapi 的 c 接口，我们加载模型，查询模型的输入输出信息，包括输出张量形式，输入输出类型，输入输出尺寸，输入输出量化零点与放缩值等信息，将 ffmpeg 的 AVFrame 转换为复合输入格式的张量，将输入正向量化为输入类型，传递给模型，进行推理，获取模型输出的原始数据，进行反向量化，获取模型输出。

#### (4) 通信模块设计

SerialPort
-fd: int
+SerialPort(port: const char*, baudrate: int)
+~SerialPort()
+isOpen() : bool : const
+writeData(data: const char*, length: size_t) : int
+readData(buffer: char*, bufferSize: size_t) : int

图 12

我们使用串口进行下位机通信，将串口设置为阻塞读模式，实现读写接口，使用 /dev/ttyS9 串口。

#### (5) QT 线程设计与可视化模块设计

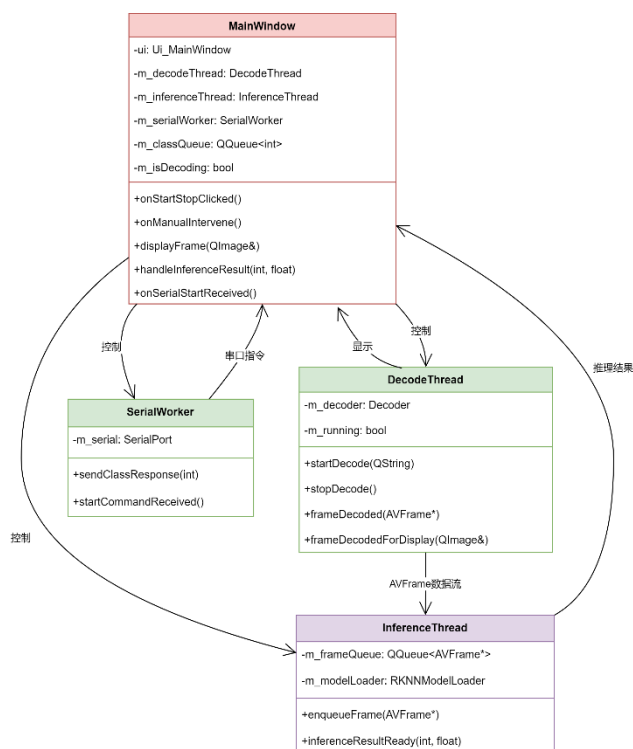


图 13

系统主要由 4 个 QT 线程实现

解码线程(DecodeThread)：通过 FFmpeg-Rockchip 获取视频流，完成硬件解码后产生 AVFrame，调用视频解码与图片放缩模块

推理线程(InferenceThread)：消费 AVFrame 数据，调用 RKNN 模型进行 NPU 加速推理，调用模型加载与推理模块

串口线程(SerialWorker)：独立处理串口通信，避免阻塞 UI 主线程，调用通信模块

主线程(MainWindow)：负责界面更新和全局状态管理

本系统采用生产者-消费者模型构建四级流水线架构,通过 Qt 信号槽机制实现线程间通信。数据流与线程通信如下图所示

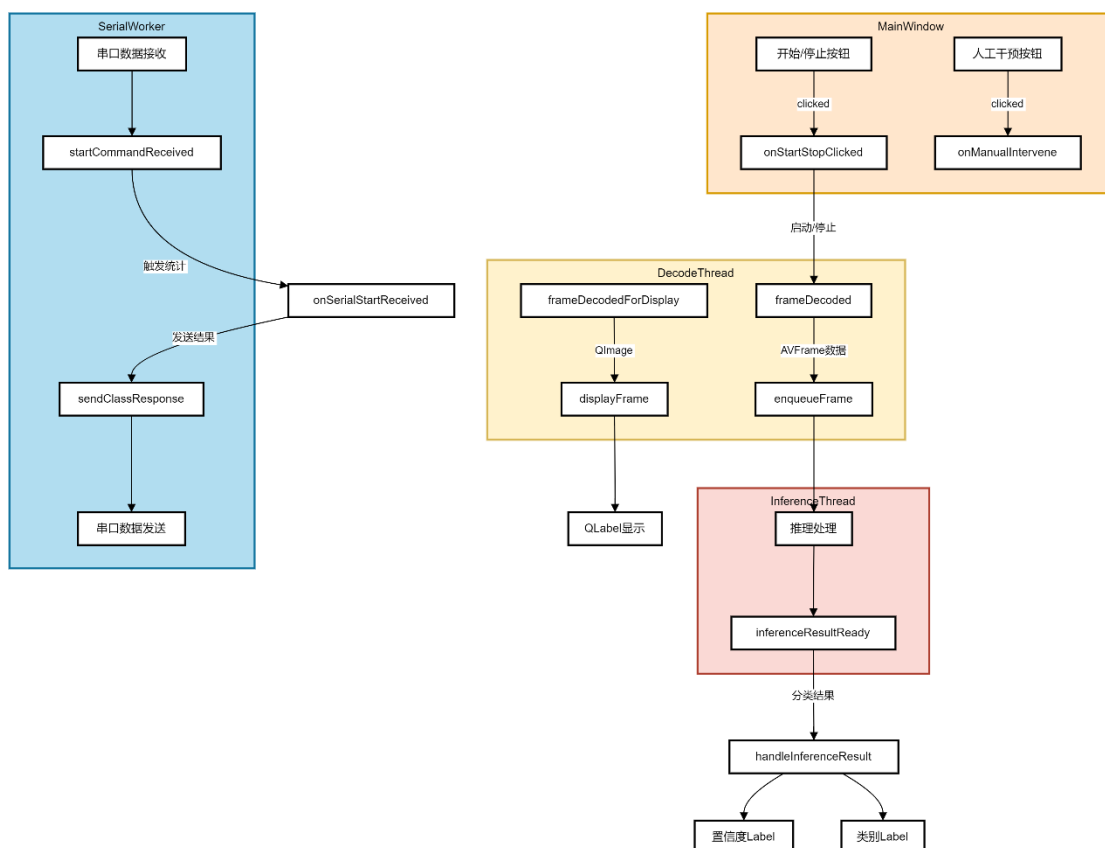


图 14

## 第三部分 完成情况及性能参数

### 3.1.1 机械成果



完成了零件定制、分拣系统装配，具体实现如图 15 所示

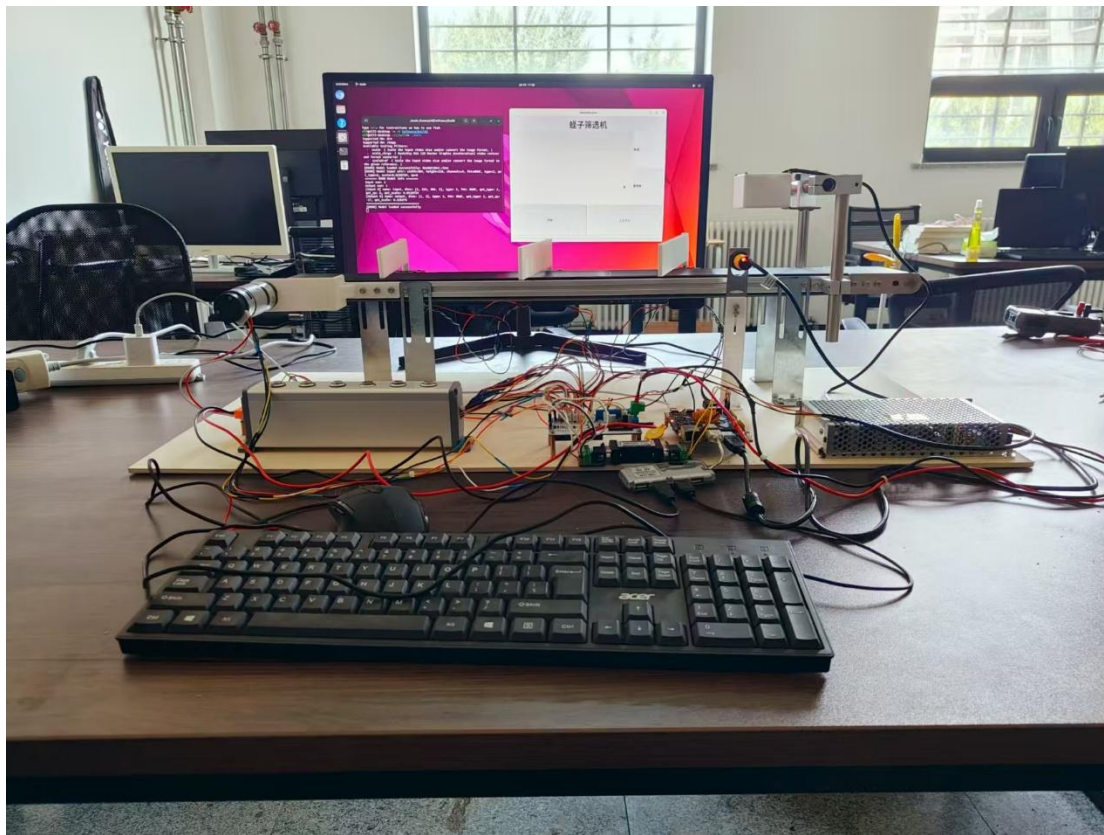


图 15

### 3.1.2 电路成果

完成了下位机控制电路，如图 16 所示

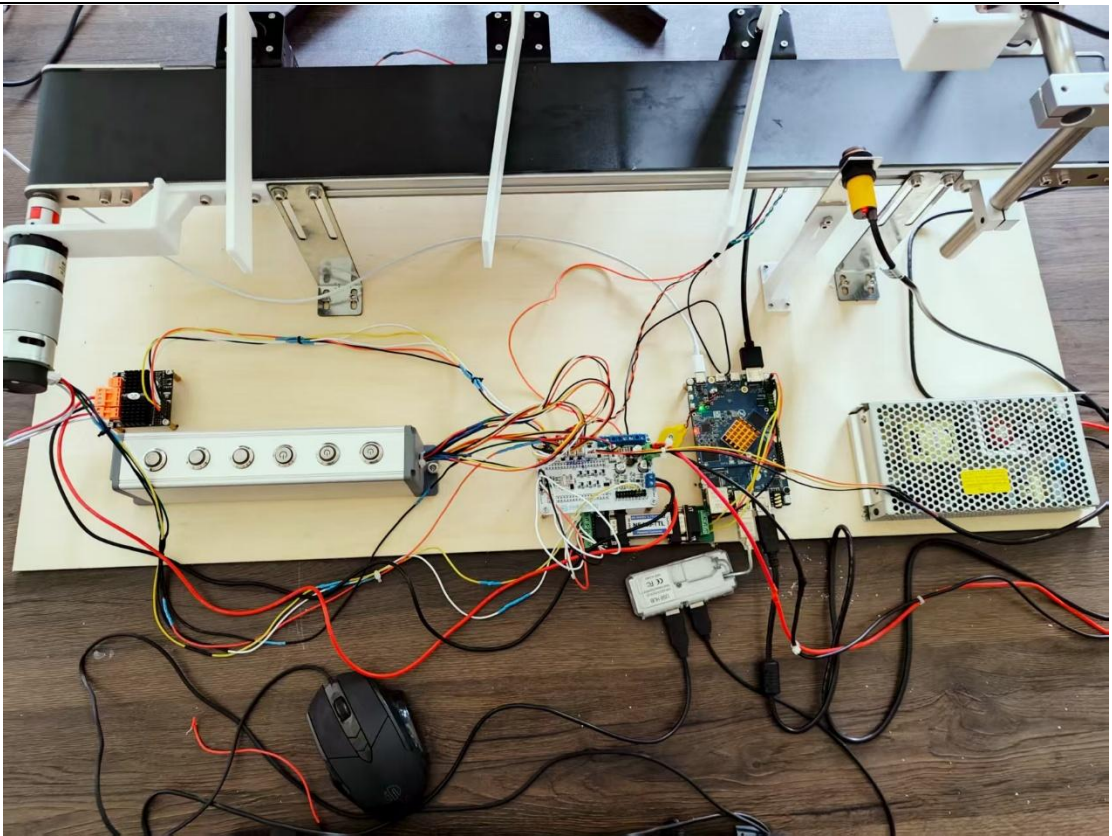


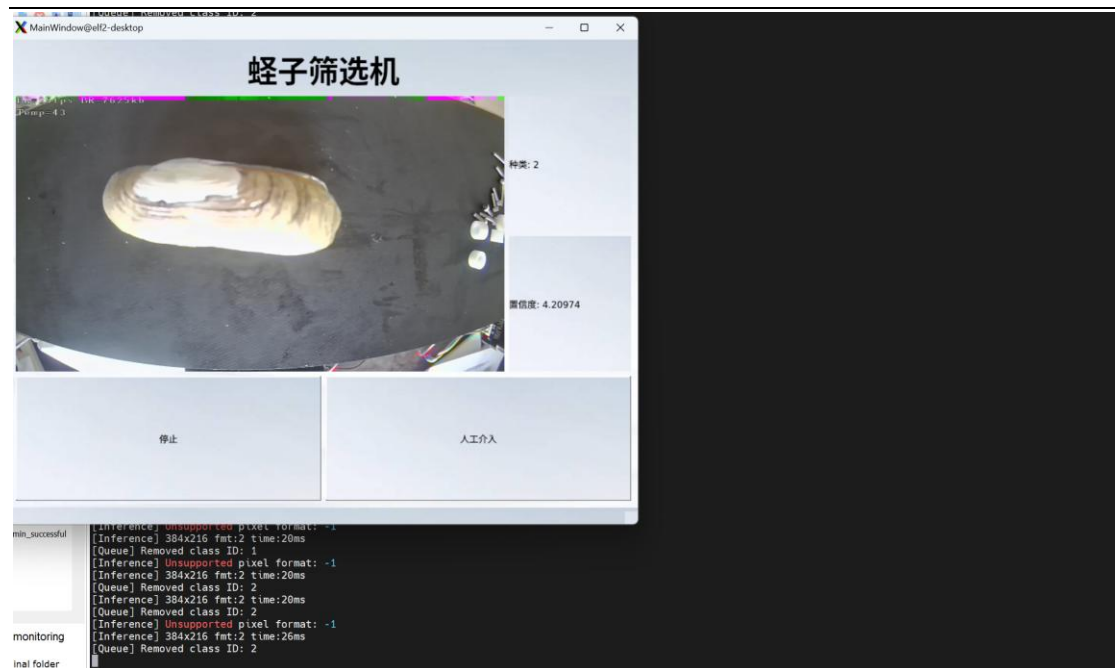
图 16

### 3.1.3 软件成果



```

elf2-desktop ~/5/build> ls
CMakeCache.txt  CMakeFiles  cmake_install.cmake  communication  decoder  main  main_autogen  Makefile  modelloader  output.mkv  ResNet50v2.rknn  scaler  tes
elf2-desktop ~/5/build> ./main
Supported HW: drm
Supported HW: rkmp
Available Scaling Filters:
- scale ( Scale the input video size and/or convert the image format. )
- scale_rkrga ( Rockchip RGA (2D Raster Graphic Acceleration) video resizer and format converter )
- scale2ref ( Scale the input video size and/or convert the image format to the given reference. )
[RKNN] Model loaded successfully: ResNet50v2.rknn
[RKNN] Model input attr: width=384, height=216, channels=3, fmt=NHWC, type=2, qnt_type=2, scale=0.0210724, zp=6
===== RKNN Model Info =====
Input num: 1
Output num: 1
[Input 0] name: input, dims: [1, 216, 384, 3], type: 2, fmt: NHWC, qnt_type: 2, qnt_zp: 6, qnt_scale: 0.0210724
[Output 0] name: output, dims: [1, 3], type: 2, fmt: NHWC, qnt_type: 2, qnt_zp: -17, qnt_scale: 0.120278
=====
[RKNN] Model loaded successfully
    
```



```
elf@elf2-desktop ~ [1]> sudo cat /sys/kernel/debug/rkrga/load
num of scheduler = 3
===== load =====
scheduler[0]: rga3
        load = 4%
-----
scheduler[1]: rga3
        load = 0%
-----
scheduler[2]: rga2
        load = 0%
-----
        process 1: pid = 1932, name: ./main
elf@elf2-desktop ~>

elf@elf2-desktop ~> sudo cat /sys/kernel/debug/rknpu/load
[sudo] password for elf:
NPU load:  Core0: 32%, Core1:  0%, Core2:  0%,
```

图 17

## 第四部分 总结

### (1) 零拷贝推理优化

当前系统虽已实现视频解码和图像处理的硬件加速，但在数据流传输过程中仍

存在少量内存拷贝操作。RKNN 提供了零拷贝推理 API，直接从硬件中读取输出，进一步降低延迟并提升系统吞吐量。

### **(2) 模型轻量化与剪枝**

现有 ResNet50v2 模型在保证精度的前提下，可通过通道剪枝、知识蒸馏等技术进行深度优化。结合 RKNN-Toolkit2 的混合量化功能，开发专用于竹蛭识别的轻量级网络，在保持 95%+ 识别率的同时，将模型体积压缩至现有 1/3 以下，使推理速度突破 100FPS。

### **(3) 数字孪生集成**

通过开发 Web 端管理平台，构建物理系统的数字孪生模型。实时可视化显示分拣数量、设备状态、质量统计等数据，并支持远程参数配置与 OTA 固件更新，实现智能化运维。

## **心得体会**

通过本项目的完整研发周期，团队成员在技术实践和工程管理方面获得了宝贵经验，主要体会有以下几点：

### **(1) 边缘计算落地的挑战与突破**

在实际部署 RK3588 芯片的过程中，我们深刻认识到边缘 AI 部署远比云端部署复杂。最初直接将实验室训练的浮点模型部署到 NPU 时，出现了严重的精度损失问题。通过反复试验，我们掌握了模型量化的关键技术：采用分层量化策略，对网络不同层采用不同的量化参数；设计校准数据集时充分考虑实际场景的光照变化；通过量化感知训练提升低比特模型的精度。这个过程让我们明白，边缘计算不是简单的模型移植，而是需要建立从数据采集、模型设计到部署优化的完整技术闭环。

### **(2) 从技术到产品的思维转变**

在项目推进过程中，我们经历了从追求技术先进性到注重产品可用性的转变。例如，最初设计的深度学习模型虽然达到较高的实验室识别率，但推理速度只



能支持低速的传送带速度。通过分析实际需求，我们降低训练测试，调整学习率，让模型获得不那么高的准确度，但是推理速度大幅提升。这个决策过程教会我们，优秀工程师不仅要会解决技术问题，更要懂得在多重约束下做出最优权衡。

### **（3）国产化技术栈的实践收获**

选择 RK3588 国产芯片作为核心处理器，使我们深入掌握了国产芯片的开发要领：需要更深入地理解硬件特性才能充分发挥其性能；开源社区的支持虽不如国际大厂完善，但技术响应更及时。这套技术栈的成功应用，增强了我们对国产化替代的信心，也积累了宝贵的适配经验。

这些心得体会将成为团队后续研发工作的宝贵财富。我们认识到，真正的技术创新不仅需要技术突破，更需要建立面向实际应用的系统工程思维，这正是本项目带给我们的最大收获。





## 第五部分 参考文献

- [1] 杨克义,赵康迪,李杲阳,等.适配 RK3588 的 YOLOv5 改进方法[J].物联网技术,2025,15(13):119-121+125.DOI:10.16667/j.issn.2095-1302.2025.13.028.
- [2] 曾烨林,林栎,赵亮.基于剪枝与量化的 ResNet 模型硬件加速方法[J].计算机工程与设计,2025,46(06):1601-1608.DOI:10.16208/j.issn1000-7024.2025.06.009.
- [3] 赵江涛,夏杰,秦翔.基于国产 NPU 的边缘端嵌入式智能计算系统设计与实现[J/OL]. 集成电路与嵌入式系统,1-7[2025-07-10].<https://doi.org/10.20193/j.ices2097-4191.2025.0011>.
- [4] 邢家旺.基于嵌入式 NPU 的目标检测系统研究[D].石家庄铁道大学,2024.DOI:10.27334/d.cnki.gstdy.2024.000233.
- [5] 涂静正,温晓婧,陈彩莲,等.基于边缘计算的工业视频网络智能感知:挑战与进展[J/OL].自动化学报,1-24[2025-07-10].<https://doi.org/10.16383/j.aas.c240668>.
- [6] 张媛荣,张轩雄,邓辰鑫,等.基于 YOLOv8 的消火栓状态在线监测与预警研究[J/OL].电子技术,1-9[2025-07-10].<https://doi.org/10.16180/j.cnki.issn1007-7820.2026.04.006>.