高速运动目标的识别与追踪设计文档

张　畅 2020020422

王凯森 2020090320

第一部分　设计概述

1.1 设计目的

运动目标检测与跟踪技术是计算机视觉领域中最重要的研究内容之一，随着数字电路的飞速发展，FPGA 的出现为数字图像处理在算法和结构上带来新的解决方法和设计思路，这是因为图像中的所有像素都可以执行相同的操作，非常适合于采用 FPGA 的并行架构通过硬件实现。基于此，我们决定设计这样一个作品，实现高速运动目标的识别与追踪。

本作品基于 GOWIN2A 使用 RISC-V 架构的 GOWIN IP 软核，在 GoAI 平台在板上部署 CNN 来运行深度训练过的模型，使用 TinyMaix 框架将学习后的目标二次推理后，输入到资源有限的板上，最终完成了目标识别跟踪的功能，并将其通过 HDMI 显示在外接屏幕。

1.2 应用领域

本作品应用领域广泛，可用于如下场景：

1. 智能视频监控：基于运动识别（基于步法的人类识别、自动物体检测等），自动化监测（监视一个场景以检测可疑行为）；交通监视（实时收集交通数据用来指挥交通流动）；

2. 人机交互：传统人机交互是通过计算机键盘和鼠标进行的，为了使计算机具有识别和理解人的姿态、动作、手势等能力，识别和跟踪技术是关键；机器人视觉导航：在智能机器人中，智能机器人视觉系统有着执行自主定位、环境识别、障碍物检测、目标跟踪等仿生功能。

3. 医学诊断：跟踪技术在超声波和核磁序列图像的自动分析中有广泛应用，由于超声波图像中的噪声经常会淹没单帧图像有用信息，使静态分析十分困难，而通过跟踪技术利用序列图像中目标在几何上的连续性和时间上的相关性，可以得到更准确的结果。

1.3 主要技术特点

传统的机器视觉系统存在着实时性差的问题，于是根据 FPGA 体积小、功耗低等优势，我们提出了基于 FPGA 的目标识别和跟踪。其中主要设计方案：使用 RISC-V 架构的 GOWIN 软核，在板上部署 CNN 来运行深度训练过的模型，部分神经网络模型可以直接引用，其他用 opencv 和 numpy 工具，传统方式是利用目标的特征点进行匹配，但是 opencv 已经引入了接口使用，用三个操作：特征提取(Feature Extraction)、特征匹配(Feature Matching)、目标识别(Object Detection)提升，特征提取(feature extraction)，用 SIFT 算法，Opencv SIFT 函数库，实现。第二步，使用 TinyMaix 框架将学习后的目标二次推理后，输入到资源有限的板上，最终完成了目标识别跟踪的功能。

1.4 主要创新点

- 使用连通域实现目标检测；
- 高命中率的 Cache 内存子系统；
- 支持图像裁切、缩放硬件加速；
- 支持并行加法、移位硬件加速；
- 支持垂直同步，防止画面撕裂；
- 计算资源复用，卷积与全连接共用 MAC；
- 支持卷积、全连接、最大池化硬件加速；
- 除 PicoRV32I 软核与 DDR3 等 IP 外均为 RTL 实现。

第二部分　系统组成及功能说明

2.1 整体介绍

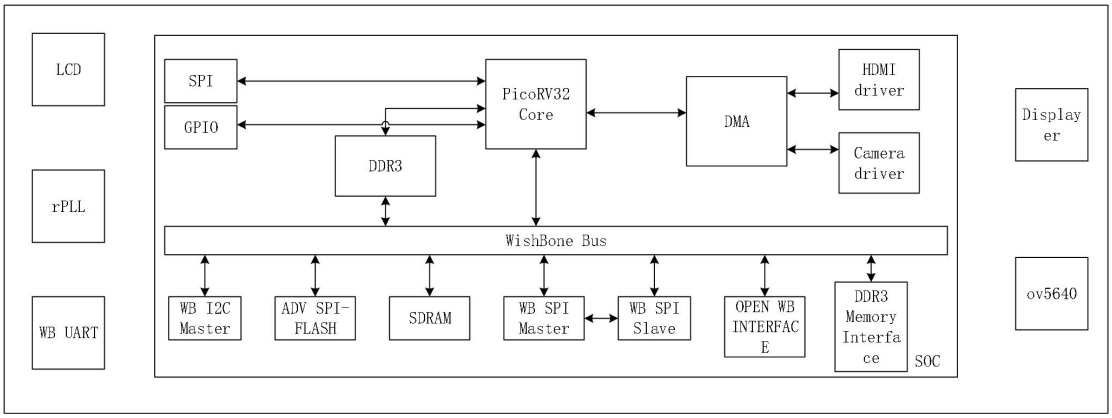


图 2.1 基于 PicoRV32I 系统设计架构图

1）MCU 端。此部分以 PicoRV32Core 为主，添加了 UART 外设，按键外设和 LCD 显示外设。用于负责系统控制、数据的传输以及与外界的交互。

2）存储模块。在设计功能时为了让 CPU 可对参数进行更新，并且能从 ram 中读取出结果，本项目采取 CPU 端进行了内存拓展。由于考虑到 DDR 无法初始化，本项目额外增加 BRAM 用于存储参数，使得其在 FPGA 下载后就存有参数。如果在设计智能算法数据量超出了 RAM 大小，就需使用大容量的 DDR，但是读取速度逊于片上 BRAM。

3）数据采集与驱动模块。此部分用于采集摄像头的输入数据以及 HDMI 显示。由于模块的带宽需求较高，所以采用硬件的方式来采集与驱动。

4）协处理器模块。此模块主要用于接收 RISC-V 拓展指令，并进行译码、执行、回写。而执行则主要分两种操作，对寄存器的配置和控制工作的开始。

5）硬件加速模块。此模块主要用于对卷积、全连接以及池化进行加速。通过配置寄存器中的信息进行操作。

2.2 各模块介绍

2.2.1 图像预处理模块

由于环境的复杂性和干扰物体的存在，所得到的图像序列通常包含大量影响因素。图像预处理即对传感器所得到的图像序列进行相关处理，使处理后的图像有利于进行待跟踪目标的识别和分割工作。通常，用于目标跟踪系统的图像预处理技术并不考虑图像质量下降的原因，只突显出图像中感兴趣的区域，削弱非目标区域的图像特征，即阁像增强技术。阁像预处理是目标识别与动态跟踪系统中的重要环节，特别是在图像质量较低的情况下。根据不同的成像方式，所用到的图像预处理方式也不尽相同。图像预处理技术主要分为两大类：空间域图像增强技术和频率域图像增强技术。空间域法是对图像直接在空间域内处理，包括图像增强、局部统计法、图像的平滑、图像的锐化等几个方面。频率域法针对图像在频率域中的变换值进行计算，包括图像的傅里叶变换、图像低通、高通滤波等方式。

2.2.2 运动目标检测模块

目标的检测与分割算法是目标识别与动态跟踪技术中的一个重点和难点。目标检测是在传感器拍摄的图像序列中检测出目标的位置，目标的分割是将目标从背景图像中提取出来。日前，常见的日标检测与分割ガ法有背景差分法、帧间差分法和光流场法"。不同的图像类型应选用不同的检测与分割方法。由于图像序列的质量容易受到光线、天气和物体阴影的影响，如何准确的进行目标的检测和

分割仍是目标识别与动态跟踪技术研究的一个难点。

目标特征提取与识别。特征提取的优劣将直接影响整个目标识别与动态跟踪系统的准确性和稳定性，如何准确的提取目标的特征对接下来的跟踪过程至关重要。一般而言、目标的特征必须满足唯一性、不变性和稳定性。特征的唯一性是指目标区别于图像序列中其他物体的特征：特征的不变性是指针对同类型目标，特征的提取方式应该相同：特征的稳定性是指特征不容易因为天气等外界因素而消失。通常，目标的形状、灰度、纹理、边缘信息都可以作为目标的特征，不同的特征对应不同的提取方法。对于目标的识别通常是建立在先验知识的基础上，常用的方法有模板匹配法和神经网络分类法。

2.2.3 基于模板匹配的目标跟踪算法模块

利用已经提取到的目标特征，对图像序列中的每一帧图像进行匹配，并标记最佳匹配区域就能实现对目标的跟踪，同时可以得到目标的运动轨迹，估算目标相对运动速度和方向等信息。目前常见跟踪方法有模板匹配法”、基于轮廓的匹配跟踪算法”、基于模型的匹配跟踪算法、基于特征的匹配跟踪算法″。这些算法都力求在保证目标跟踪系统跟踪精度的基础上，提高系统的实时性。在实际应用中，根据不阿的目标特征和系统要求，选用不同的跟踪算法。

第三部分 完成情况及性能参数

具体完成的情况，可图文结合，具体的性能参数等量化指标

3.1 软件设计

3.1.1 UART 连接上位机

在 TangPrimer 20K 原有的外设中便已自带了 UART 外设，且提供了驱动 API,可在 GPIO 配置为 UART。通过 UART 上位机可以很方便的对 TangPrimer 20K 进行数据传输与控制。在设计时，分别为两种不同用途，即上机测试与正常工作，设计了不同的驱动。

3.1.2 GPIO 配置按键中断

在实际应用中，按键是种简单、高效的控制手段。本项目利用开发板上的按键连接到 CPU 端的 GPIO 端口上来完成中断控制。实现的效果为，当我们按下按键，CPU 产生中断，中断函数指向完整的识别流程函数，此函数包括了图片数据加载、神经网络配置与启动、结果的查表判断以及打印识别信息。该流程如下图 3.1.2 所示：

```
Epoch [70], train_loss: 0.3306, train_acc: 0.8959,val_loss: 0.0100, val_acc: 0.9987
Epoch [71], train_loss: 0.3269, train_acc: 0.8963,val_loss: 0.0113, val_acc: 0.9977
Epoch [72], train_loss: 0.3172, train_acc: 0.8999,val_loss: 0.0151, val_acc: 0.9956
Epoch [73], train_loss: 0.3143, train_acc: 0.9005,val_loss: 0.0077, val_acc: 0.9988
```

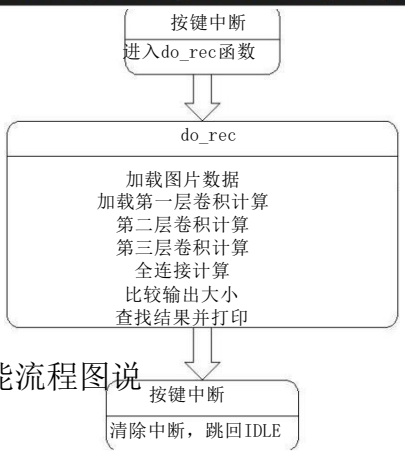


图 3.1.2 按键控制功能流程图说明

3.1.3 SPI 驱动 LCD 图

由于 TangPrimer 20K 原有的外设中已自带了 SPI，且提供了驱动 API，因此可将 GPIO 配置为 SPI。同时配合 labs1 中提供的相关 LCD 驱动函数，可实现 CPU 端对 LCD 屏幕的显示控制。显示主要内容为识别物品名称、物品速度等。

3.2 神经网络设计

1) 基于 Pytorch 编写设计面向应用的神经网络，并进行训练和测试。将所编写的神经网络作为总对照母本，来获得训练好的权重参数，并利用 hook 钩子函数获得神经网络运行时相应的输入和输出来进行调试。

2) 基于 Python 建模：利用软硬协同的思想，通过用 Python 语言来模拟构造卷积和池化两个硬件通用功能模块，并通过传入不同层神经网络对应的相关参数以及反复调用这两个模块来模拟搭建硬件可实现的网络模型。

3) 软件优化：数据预处理增强，数据集混合，模型融合，参数量化。

3.2.1 模型结构

整体网络架构由基本块和全连接层共同搭建而成。该神经网络训练后车牌识别准确率可达到 99%，如图 4.2.2-2 所示，数据增强后经历 70 多次循环训练可以看到识别准确率收敛到 90%左右，其中 train_acc 为训练集准确率，val_acc 为测试集准确率，loss 为各自对应的梯度下降值。将该训练好的神经网络作为母本，用于测试 python 搭建的硬件网络模型。

图 3.2.1 数据增强后神经网络识别率

3.2.2 基于 python 建模

为了验证所构造的模型能否成功移植到硬件中实现，本项目利用软硬协同的方法，采用 Python 进行硬件建模，来描述卷积和池化两个网络基本块的逻辑功能。

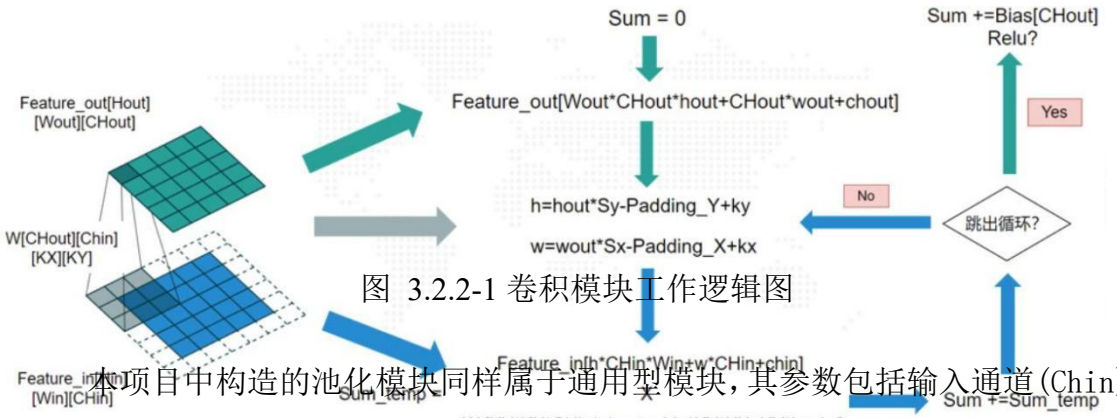


图 3.2.2-1 卷积模块工作逻辑图

本项目中构造的池化模块同样属于通用型模块，其参数包括输入通道(Chin)、输入特征尺寸(Hin, Win)、池化区域长宽(KX, KY)、Mode（选择池化种类）、输入特征(feature_in)。其中输入特征的维度顺序如下：

feature_in[Hin][Win][CHin]，数组的访问形式依旧采用一维数组。

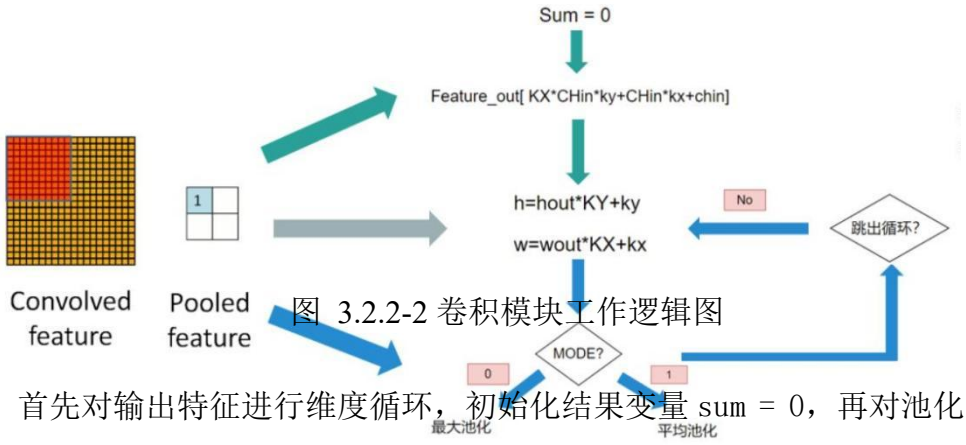
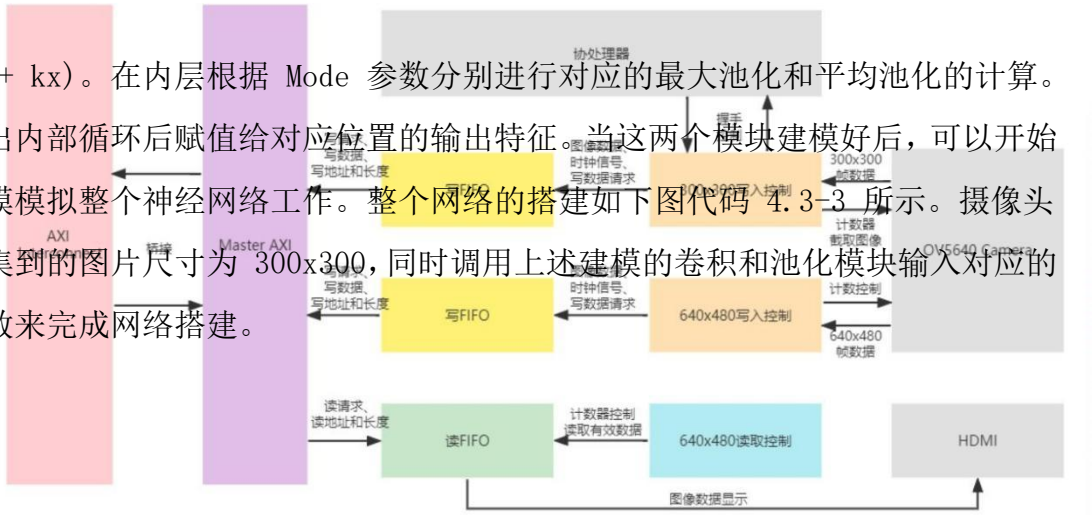


图 3.2.2-2 卷积模块工作逻辑图

首先对输出特征进行维度循环，初始化结果变量 sum = 0，再对池化区域进行循环，计算出对应于输入特征的坐标映射(h = hout × KY + ky, w =wout ×

KX + kx)。在内层根据 Mode 参数分别进行对应的最大池化和平均池化的计算。跳出内部循环后赋值给对应位置的输出特征。当这两个模块建模好后，可以开始建模模拟整个神经网络工作。整个网络的搭建如下图代码 4.3-3 所示。摄像头采集到的图片尺寸为 300x300，同时调用上述建模的卷积和池化模块输入对应的参数来完成网络搭建。




```
image_init= pool(3,300,300,3,3,0,image)
conv1_out = conv(3, 16, 100, 100, 3, 3, 1, 1, 1, 1, image_init, W1, bias1, scalew=1)
if 127 in conv1_out:
    print('conv1 over')
pool1_out = pool(16, 100, 100, 2, 2, 0, conv1_out)
conv2_out = conv(16, 32, 50, 50, 3, 3, 1, 1, 1, 1, pool1_out, W2, bias2, scalew=0.03125)
if 127 in conv2_out:
    print('conv2 over')
pool2_out = pool(32, 50, 50, 2, 2, 0, conv2_out)
conv3_out = conv(32, 64, 25, 25, 3, 3, 1, 1, 1, 1, pool2_out, W3, bias3, scalew=1.0)
if 127 in conv3_out:
    print('conv3 over')
pool3_out = pool(64, 25, 25, 5, 5, 0, conv3_out)
fc_out = conv(64, 131, 5, 5, 5, 5, 1, 1, 0, 0, pool3_out, FC_weight, FC_bias, scalew=2.0)
```

图 5.2.4-调用卷积池化模块搭建网络关键代码

3.3 图像采集控制

图像采集控制功能主要通过摄像头图像采集与协处理器进行协同工作来实现控制摄像头 OV5640 采集帧数据并利用 FIFO 进行缓存，通过 AXI4 总线写到 BRAM 中，以及通过 HDMI 进行帧数据从 BRAM 读出同理利用 FIFO 缓存以及显示。

为了对图像进行初步的裁剪工作，即将分辨率为 640*480 的图像裁剪为 300*300 的图像用于神经网络识别，将摄像头采集的图像数据同时进行两种处理，一种以 I2C 设置的 640*480 分辨率正常采集并存入 BRAM 中，用于给 HDMI 进行显示；一种按照 300*300 的分辨率来采集图像数据，即采集原始图像的中间 300*300 部分数据以同样的方式存入 BRAM 中，但需要与正常模式采集的数据的地址间隔一定大小来保证数据没有冲突。

图 3.3 数据采集控制整体框架

3.3.1 OV5640 摄像头数据采集模块：

OV5640 摄像头中传感器在 HREF（行同步信号）为高电平时候输出有效图像数据，当 HREF 变高时，每一个 cam_pclk 时钟输出一个 8bit 像素数据，对于 HDMI 显示我们采用 RGB565 转 RGB888 格式输出，其转换是通过量化补偿方法进行填充；对于 300x300 进行图像识别的帧数据，我们采用 RGB565 转 RGB161616 格式输出，即先按照上述方式转换成 RGB888 格式，然后再给每个通道高 8bit 进行补零操作（选择 RGB161616 为了提高图像识别精度）。因此每两个 cam_pclk 时钟组成一个像素数据，将原像素数据进行拼接与扩充并将数据有效信号拉高一并传给两个写缓存队列进行数据缓存。

对于 OV5640 摄像头的配置，我们采用 I2C 协议进行配置，其主要配置部分为 ISP 输入窗口设置、预缩放窗口设置和输出大小窗口设置、输出像素格式设置。

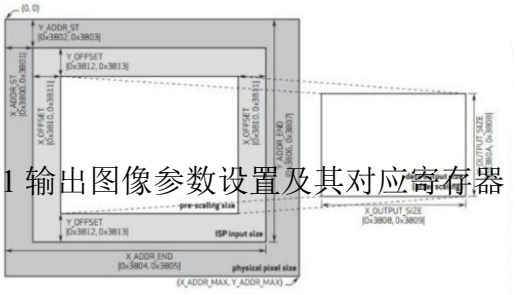


图 3.3.1 输出图像参数设置及其对应寄存器

第四部分　总结

本次设计主要实现了在一定程度上通用的神经网络加速器，并采用协处理器加自定义指令的方式来驱动，致力于满足多种智能算法应用场景。本次设计的卷积与池化计算模块均达到了较高的通用性，可以通过指令配置以及参数更新来实现不同智能识别算法的切换，满足不同的应用需求。

目前大致功能已经得到实现，即硬件加速、系统工作和外设驱动。但是本次的项目还有诸多不完善的地方，如硬件加速模块虽进行了部分优化，但是计算速度却不尽人意，后续还要大力提升其计算速度，可通过增加缓存、展开以及流水来提升性能。相较于初赛以及既定的功能，本项目已经得到大体的功能实现，后续就是进一步硬件优化以及软件功能的增加。

本次设计还让我们体会到很多其他东西，比如：

1) 实际工作远比看起来的多，除了可见的设计工作，我们还编写了很多脚本进行数据转换，代码编写等工作。

2) 设计系统需要学习多方面知识，为了完成本次设计工作，我们使用了 Verilog，C，Python 语言。

3) 需要对任务进行科学的分工安排，最大效率的安排人员。

4) 培养良好的资料搜索方法，避免遇到问题没有地方参考。

6) 要学会科学的 DEBUG 方法，避免盲目调试导致消耗过多时间。

7) 需要有良好的文档以及文件管理能力，避免项目文件出现版本混乱、文件混乱等问题。

第五部分 参考文献

[1].杨戈，刘宏，视觉跟踪算法综述]．智能系统学报，2010,05:95-105.

[2].候志强，韩崇昭,视觉跟踪技术综述]，自动化学报，2006,32:603-617.

[3]. Gang Yang , Fuzhou Duan , Wenhui Zhao , et . Building extraction in towns and villages based on Digital Aerial Image by texture enhancing [C]. International Conference in Geoinformation , Beijing ,2010:1-6.

[4].Viola PJones M . Rapid object detection using a boosted cascade of simple features[C].Proceeding of the 2001 IEEE computer Vision Pattern Recognition ,2001,1:511-518.

[5]. Monnet A , Mital A , Paragios N , et . Background modeling and subtraction of dynamic scenes[C]. IEEE international Conference on computer Vision ,2003:1305-1312.

[6].熊英,基于背景和赖间差分法的运动目标提取[J]计算机时代.2014,3:38-41

[7].Viola PJones M . Robust real time object detection []. International Journal of computer Vision ,2004,57:137-154.

[8].党晓军，尹俊文，一种基于模板匹配的运动目标跟踪方法期计算机工程与应用，2010,46:173-176.

[9].Weiming Hu , Xue Zhou , Wei li , et . Active Contour - Based Visual Tracking by Integrating Colors,ShapesandMotions IEEETransactionsonImage Processing ,2013,22:1778-1792.

[10]. Kyrki . Integration of Model - based and Model - free Cues for Visual Object Tracking in 3D[C].IEEE International Conference on Robotics and Automation ,2005:1554-1560.