

DOI: 10.13652/j.issn.1003-5788.2020.08.023

基于 FPGA 的快速樱桃缺陷检测与识别系统设计

Design of fast cherry defect detection and recognition system based on FPGA

裴悦琨^{1,2} 谷 宇^{1,2} 连明月^{1,2}PEI Yue-kun^{1,2} GU Yu^{1,2} LIAN Ming-yue^{1,2}

(1. 大连大学辽宁省北斗高精度位置服务技术工程实验室, 辽宁 大连 116622;

2. 大连大学大连市环境感知与智能控制重点实验室, 辽宁 大连 116622)

(1. *Beidou High Precision Positioning Service Technology Engineering Laboratory of Liaoning**Province, Dalian University, Dalian, Liaoning 116622, China*; 2. *Environment Sensing and**Intelligent Control Key Laboratory of Dalian, Dalian University, Dalian, Liaoning 116622, China*)

摘要:为使樱桃缺陷检测与识别系统满足实时性的要求,提出以卷积神经网络模型为基础,使用 SDSoC 开发平台,完成 FPGA 对樱桃缺陷进行快速检测与识别系统的设计。通过优化数据传输,复用网络模型中通用矩阵乘法函数(GEMM)和对卷积操作进行并行化设计,实现 PL 端硬件加速。利用 SDSoC 平台,在 PS 端使用高级语言映射卷积神经网络模型,在实现所需性能的同时大量节省了开发时间。结果表明,与纯软件方式相比,基于 Zynq7020 硬件开发平台,速度提高了 2.19 倍以上,与 CPU 平台相当。

关键词:缺陷检测;樱桃分级;卷积神经网络;硬件加速;现场可编程门阵列;SDSoC

Abstract: To make the cherry defect detection and identification system meet the real-time requirements, it was proposed to use the convolutional neural network model as the basis and used the SDSoC development platform to complete the design of FPGA for rapid detection and identification of cherry defects. By optimizing data transmission, multiplexing the general matrix multiplication function (GEMM) in the network model and parallelizing the design of the convolution operation, PL-side hardware acceleration was realized. Using the SDSoC platform, a high-level language mapping convolutional neural network model was used on the PS-side, which saved a lot of development time while achieving the required performance. The results showed that, compared with the pure software method, the speed based on the Zynq7020 hardware development platform had increased by more than 2.19

times. Compared with the CPU platform, the speed was almost the same.

Keywords: defect detection; cherry grading; convolutional neural network; hardware acceleration; field programmable gate array; SDSoC

卷积神经网络(convolutional neural networks,CNN)是目前性能较好的深度学习算法之一。通过其网络模型和足够的训练数据集,CNN 可以为某些任务生成复杂的功能,胜过传统的人工算法。目前,CNN 已成功应用于手写体数字识别及交通标志识别^[1-2]等,并且取得了较好的效果。

基于 CNN 的水果品质检测大多是在软件平台上实现。刘云等^[3]基于 CNN 对苹果进行分块缺陷检测,并且在软件平台上进行训练和测试,检测速度达到 5 个/s,且正确率高达 97.3%;裴悦琨等^[4]对樱桃的缺陷进行检测和识别,利用 CNN 对采集的樱桃图片进行测试,基于 CPU 的软件平台上识别速度可达 25 个/s;伍锡如等^[5]构建一个多层卷积神经网络,在英特尔 i5 平台上对水果采摘机器人视觉识别系统进行测试,单张水果图像的识别速度只需 0.2 s。这些软件平台都是基于通用的处理器,然而 CNN 具有很高的并行度,通用处理器主要用来控制指令调度、执行和逻辑判断,并不适合用来大量的并行计算^[6]。因此基于软件方式的 CNN 在实时性和能耗方面都不能适应实际水果检测中场外作业的需求。目前大部分对于 CNN 的研究主要还是采用 GPU,但是对于 GPU 功耗大的问题一直存在,所以在需要电池供电的嵌入式设备中也是很难得到应用。

现场可编程门阵列(field programmable gate array,

基金项目:国家自然科学基金项目(编号:61601076)

作者简介:裴悦琨(1985—),男,大连大学讲师,研究生导师,博士。E-mail:peiyuekun@dlu.edu.cn

收稿日期:2020-05-29

FPGA)具有强大的并行处理能力、灵活的可配置特性和超低功耗,使其成为 CNN 实现平台的理想选择。王巍等^[7]充分利用 CNN 的并行计算特征,进行了 CNN 算法的 FPGA 并行结构设计,提高了资源利用率且计算效率也得到大幅提升;Ma 等^[8]使用 RTL 编译器完成 CNN 网络的硬件加速,使用 FPGA 在 100 MHz 的工作频率下得到 2 倍的性能提升,然而上述基于硬件描述语言(HDL)或者逻辑图的设计方法难度较大,周期较长并且在数学运算上存在许多弊端,极大地阻碍了 FPGA 中进行人工智能的快速开发^[9]。Danopoulos 等^[10]的研究充分地展现了 FPGA 快速开发的优势,将网络框架移植到 FPGA 的 ARM(基于 Zynq-7000)处理器中,在 Xilinx 的 SDSoC 开发环境中设计硬件加速器,并利用硬件加速器显著提高神经网络执行速度。

试验拟采用美国 Xilinx 公司发布的 SDSoC 开发环境,通过使用 C/C++ 在目标平台上完成完整的硬件/软件系统的编译、实现、调试执行等全过程^[11]。利用卷积网络并行结构的特点,采用高级语言映射卷积神经网络模型并进行优化设计,通过处理器系统(Processing System, PS)和可编程逻辑(Programmable Logic, PL)的协同处理实现一个完整的软硬件系统,为樱桃的快速、准确地检测分级提供新的方法和策略。

1 卷积神经网络

相比普通的图像处理算法,卷积神经网络仅需少量的预处理,识别范围广,能容许图像的畸变,对几何变形具有很好的鲁棒性,其经典网络模型包含卷积层、池化层、全连接层等结构。其中卷积层即特征提取,针对图像上存在的特征进行局部感知,紧接着进行多层卷积获取全局信息。

池化层也叫下采样层。主要用作降低特征维度,减小过拟合的发生,同时增强系统的容错性。

全连接层即输出层,主要对数据进行归一化处理,避免数据值大的节点对分类造成影响,最终进行正确的图片分类。

卷积过程如图 1 所示:输入层的图像首先通过多个卷积层、修正线性单元(rectified linear unit, ReLU)以及最大池化层运算,之后进入全连接层,最后通过输出层得到最终的分类结果。

针对樱桃缺陷的特点,设计 CNN 模型。该模型包括卷积层、修正线性单元、最大池化层以及全连接层。模型设计如表 1 所示。

2 樱桃缺陷检测与识别系统设计

2.1 系统结构

樱桃缺陷检测与识别系统的结构设计如图 2 所示。主要包括图像采集、HDMI 接口、SD 卡数据读取、图像预处理、PL 端加速和终端显示等模块。其中系统全局控制在 ARM 处理器完成,主要负责 HDMI 接口调用、数据传输,对图像进行预处理,并完成加速函数对 PL 端的写入工作,最终 PS 端将分类结果显示在终端显示器上。

由于卷积模块和硬件接口模块计算密集且模块化较强,因此将其模块放入 FPGA 的 PL 端实现。该系统优点是在 Xilinx 的 SDSoC 开发环境中采用软硬件结合的形式,复用卷积模块以及使用优化指令对卷积层和数据传输进行优化,节省资源,提高识别效率。

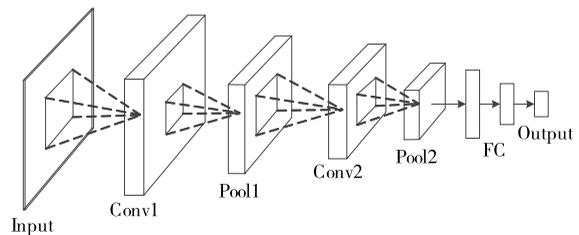


图 1 卷积神经网络结构

Figure 1 Convolutional neural network structure

表 1 CNN 模型设计

Table 1 CNN model design

网络架构	具体层	s 参数	步长	输出
Input	数据层			32×32×3
Conv1	卷积层 1	5×5×32×3+32	1	28×28×32
Relu1	激活层			
Pool1	2×2MAX 池化层 1		2	14×14×32
Conv2	卷积层 2	5×5×32×32+32	1	10×10×32
Relu2	激活层			
Pool2	2×2MAX 池化层 2		2	5×5×32
FC1	全连接层 1	5×5×32×128+128		128
Relu3	激活层			
FC2	全连接层 2	128×2+128		3
Softmax	损失函数层			

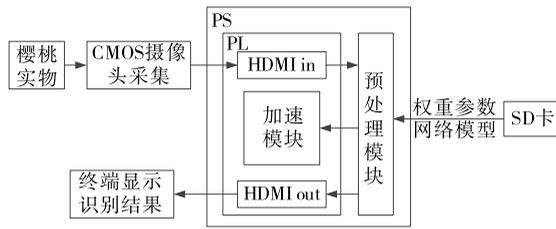


图2 系统结构设计框图

Figure 2 Block diagram of system structure design

2.2 网络映射

整个网络模型移植到 Zynq 7020 SoC 以便在 ARM 内核上运行。为了使网络在 Zynq 上运行,必须使用 SDSoC 环境中包含的 ARM 交叉编译器对整个模型进行交叉编译。SDSoC 是 Xilinx 提供的基于 IDE 的框架,该框架允许为 Zynq 平台开发硬件/软件嵌入式系统。该平台提供了定义,集成和验证硬件加速器的能力,该硬件加速器将加速特定功能,稍后将对其进行描述,生成 ARM 软件和 FPGA 比特流,同时生成 SD 映像,以便脱机进行 SD 卡读取^[12]。SDSoC 网络映射开发流程如图 3 所示。

2.3 硬件设计

该系统通过 HDMI 接收、输出图片,采用输入输出级优化、通用矩阵乘法函数复用优化以及卷积操作并行化设计,实现对输入输出级设计和 CNN 计算密集的卷积层加速。

对于 HDMI 接收设计,可调用相关 IP 进行驱动。因此硬件设计的重点即数据传输优化、通用矩阵乘法函数复用和卷积操作并行化设计和优化。

2.3.1 数据传输优化 数据传输优化主要包括 PS 和 PL 接口优化和数据传输方式优化。如图 4 所示,PortA 和 PortB 接口分别为 PS 端和 PL 端接口,Data Mover 为 PS

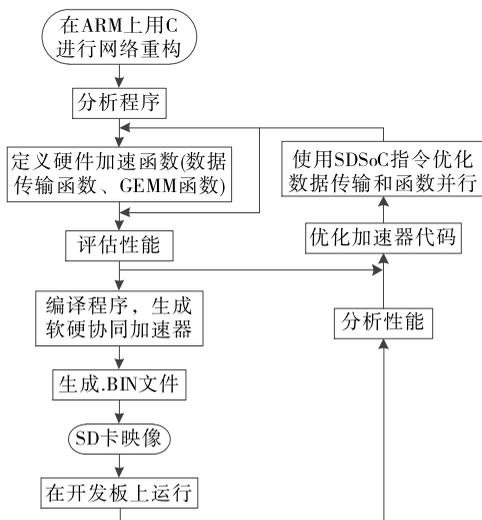


图3 SDSoC 网络映射开发流程

Figure 3 SDSoC network mapping development process

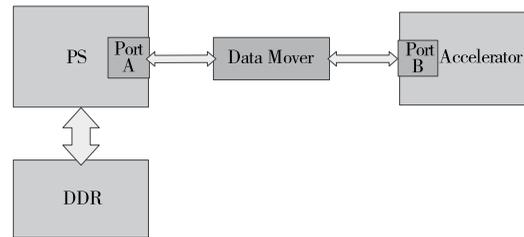


图4 数据传输接口

Figure 4 Data transmission interface

和 PL 之间进行数据传输的方式。因此,数据传输优化即对 PortA、PortB 和 Data Mover 进行优化设计。

通常大部分的权重参数和算法模型存在 DDR 中,PS 和 PL 则通过 AXI 标准总线接口进行数据的交互,AXI 接口包括通用 AXI 接口 (AXI_GP)、加速器一致性接口 (AXI_ACP) 和高性能接口 (AXI_HP)^[13]。

鉴于访问 DDR 延迟会导致读写速度缓慢,在 PS 端与 PL 端对数据的传输进行优化。传输接口及传输方式的相关优化指令:

```
//system port
#pragma SDS datasys_port(input:AXI_HP,output:
AXI_HP)

//data mover
#pragma SDS datamem _ attribute (input:
PHYSICAL_CONTIGUOUS|
NON_CACHEABLE,output: PHYSICAL_CON-
TIGUOUS|NON_CACHEABLE)

#pragma SDS data copy (input [0: WIDTH *
HEIGHT],output[0:WIDTH * HEIGHT])

//accelerator
#pragma SDS dataaccess_pattern(input:SEQUEN-
TIAL,output:SEQUENTIAL)
```

SDS datasys_port 指令用于直接约束 PS 端总线接口类型,试验选择 HP 类型。AXI_HP 接口具有高性能和高带宽的特性,其内部配置异步 FIFO 作为高速数据读写的缓冲。相反 AXI_GP 则是没有配置缓冲的中低速接口,AXI_ACP 主要实现 PS 中 cache 和 PL 单元之间的一致性接口,因此都不适合。

SDS data mem_attribute 用于约束矩阵数据存放地址的连续性。硬件综合时,SDSoC 平台会选择传输连续内存更快的 AXI_DMA_Simple,而不是 AXI_DMA_SG。

SDS data copy 指令约束内存大小,意味着 SDSoC 会将硬件加速器的接口通过 AXI 总线直接连接到 PS 的存储器,对其进行一定内存大小的数据传输。

SDS dataaccess_pattern 指令约束访问数据的方式,在 CNN 中都是按照图像每行像素的顺序进行卷积操作,所以将 SEQUENTIAL 设置为数据访问的方式,那么综

合时自动生成顺序访问的接口协议(例如 ap_fifo),而不是生成随机访问的接口,大大提高了数据访问速度。

采用综合 4 条优化指令,可以在连续的物理空间中按照数据流的方式访问数据,降低数据传输延迟,增大数据吞吐率。试验中,传输图像及权重偏置信息均采用此方式加载。

2.3.2 通用矩阵乘法函数复用 CNN 中最为密集的计算是卷积,每个卷积层都要进行线性矩阵乘法和加法运算即通用矩阵乘法(GEMM)函数,利用 FPGA 来实现可以有效地提高卷积计算效率,并降低功耗,区别在于特征图的尺寸不同。对于这一结构上的差异,应将其同构化达到复用的目的。

在不造成卷积运算误差的前提下,对所要进行卷积操作的特征图进行补零。如图 5 所示,除了输入特征图,之后的卷积操作的特征图的宽和高均小于输入特征图尺寸,因此对其进行填补零值,使其尺寸与输入特征图尺寸相同。因此,与权重的矩阵乘法操作使每一层的卷积运算具有了同构性。在 SDSoC 中可以专门定义通用矩阵乘法(GEMM)函数并且每一次卷积都可以调用该函数,卷积运算可以复用相同的控制电路,从而节省硬件资源的开销。

2.3.3 卷积操作并行化

(1) 矩阵分块优化:对于卷积层中较大的数组拆分为多个较小的数组,用于增加数据访问的并行性。通过使用 ARRAY_PARTITION 指令对数组进行分块。在文中优化卷积操作使用下述代码结构。对二维数组进行降维,将二维数组分解为若干个小寄存器,提高并行度。相关指令:

```
# pragma HLSarray_partition variable = in_
Bcomplete dim=2
```

(2) 流水线优化:流水线优化能够将一个延时较大的操作切割成多个小操作并行执行,这样就大大增加了总体的运行速度。采用 PIPELINING 约束指令来使结构执行流水线操作。为了不消耗太多的硬件资源并且不浪费太多的数据延迟,综合考虑选择在第 2 层循环内添加优化命令,可以使得工作频率提高将近 1 倍。流水线优化

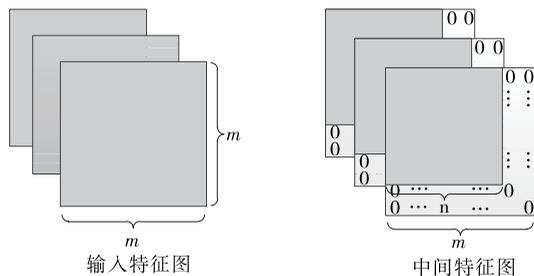


图 5 特征图同构化

Figure 5 Feature map isomorphism

使用 PIPELINING 指令:

```
for(index_a = 0; index_a < A_NROWS; index_a++){
    for(index_b = 0; index_b < A_NCOLS; index_b++){
        #pragma HLS PIPELINE
        //next operations
    }
}
```

(3) 循环展开优化:默认情况下,卷积的所有嵌套循环都是按顺序执行的。循环展开优化可以提高循环迭代之间的并行性,增加 FPGA 计算资源的利用率。如图 6 所示,将特征图和权重通过移位寄存器分别展开,并且进行并行乘加运算。

在卷积过程中,通过使用 UNROLL 优化指令对 for 循环进行展开。展开最内层 for 循环,每一层权重的乘积作为一个独立的处理单元(PE),使得 PE 核最小化。将第 3 层 for 循环展开作为 PE 的并行数量,用来一次获得多个输出层的部分和,部分代码:

```
//multiply accumulate broken into individual operators
# pragma HLS UNROLL
for(index_d = 0; index_d < B_NCOLS; index_d++){
    floatresult += weights[index_c][index_d] *
input_fm[index_c+i][index_d+j];
}
```

output_fm[i * A_NROWS + j]=result;

(4) 函数内联优化:此外,试验还使用了函数内联优化指令 INLINE,去除子函数层次结构,通过减少函数调

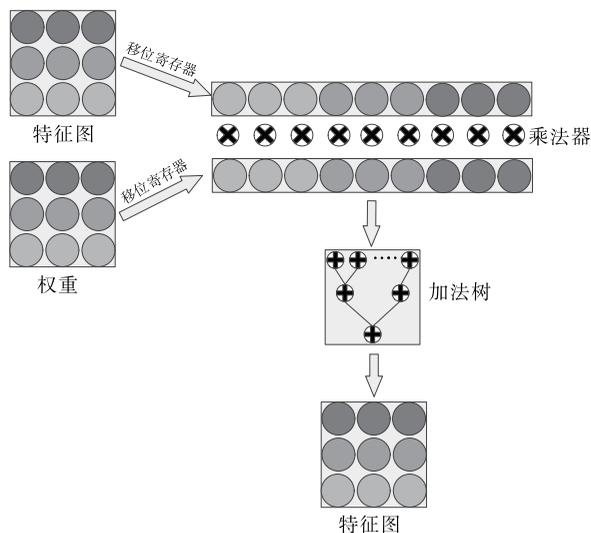


图 6 循环展开优化

Figure 6 Loop unrolling optimization

用开销来改善延迟。指令:

```
# pragma HLS INLINE self
```

3 实现

3.1 硬件实现

在硬件设备上,基于 Xilinx Zynq7020 开发板实现网络的加速,开发板为异构芯片,该芯片有 2 个 ARM A9 处理器作为处理系统(PS),还有一个可编程逻辑(PL)XC7Z020 用于硬件加速,开发板如图 7 所示;此外,开发板通过 SONY IMX222 CMOS 相机对采集的樱桃图片进行测试,试验还配有一台显示器用于测试结果的终端显示,如图 8 所示。

在硬件模块化设计上,试验基于 Xilinx Vivado 2016.4 软件,将数据传输模块,通用矩阵乘法模块以及接

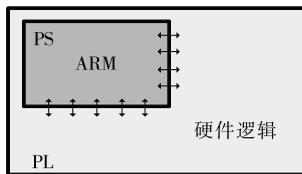
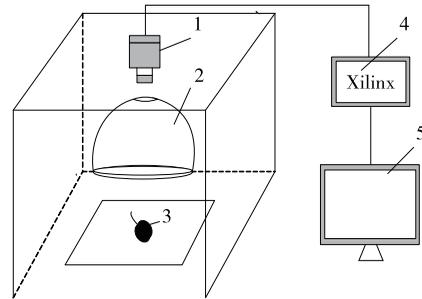


图 7 Xilinx Zynq7020 开发板结构图

Figure 7 Xilinx Zynq7020 development board structure diagram

口模块分别生成各个硬件 IP,完成硬件模块化设计。数据的内部传输采用 AXI Interconnect 模块,即 AXI 互联矩阵,能够实现将主设备和从设备的互联;外部数据传输采用 HDMI 模块实现图像数据传输,并利用异步 FIFO 模块作为帧缓冲区,缓存数据。内部 PS 模块包含一个 Zynq 处理器硬核,实现各 IP 初始化与配置。定义的 Mmult 模块实现卷积操作,其内部通用矩阵乘法 IP 在卷积层进行复用。

分别在 PC 机、纯 ARM 软件和试验加速器 3 种平台进行试验,结果如表 2 所示。



1. SONY IMX222 CMOS 相机 2. 圆顶光源 3. 樱桃样本
4. Xilinx Zynq7020 开发板 5. 终端显示器

图 8 硬件设备实现

Figure 8 Hardware device implementation

表 2 PC 机、纯 ARM 和试验加速器 3 种平台比较

Table 2 Comparison of three platforms: PC, pure ARM and test accelerator

试验平台	FPGA	通用 CPU	实现方式	时钟频率/MHz
纯 ARM 软件	—	Cortex-A9	C/C++	677
试验加速器	Zynq XC7Z020	Cortex-A9	高层次综合	300
PC 机	—	Core(TM) i5-4200	Python	2 500

3.2 制作并训练数据集

在制作数据集的过程中,在实验室的环境下对采摘来的樱桃进行图像采集。为了能覆盖每个樱桃的完整表面,试验使用 Basler acA2000-50gc 工业相机对樱桃的 6 个不同角度进行采集;利用 Matlab 对图像进行裁剪、旋转、缩放等操作来扩大样本数据;最后将所有的待训练图像统一成 32×32 规格大小。部分样本示例如图 9 所示。

训练在 Caffe 平台下进行,操作系统为 Ubuntu16.04,



(a) 裂口樱桃 (b) 完好樱桃 (c) 畸形樱桃

图 9 样本示例

Figure 9 Sample example

使用 Python 搭建 CNN 模型,采用监督式学习的方法对樱桃样本进行训练。最后的全连接层的输出通道即分类数,将樱桃分为 3 类,分别为:裂口、完好和畸形。用于训练的樱桃图片 18 000 张,采用人工标记的方法对训练样本进行标记。

4 结果与分析

4.1 数据集测试

试验选取 600 张樱桃原图进行测试,每个分类各 200 张,将裂口和畸形的樱桃归为缺陷樱桃。试验先在 PC 机上利用 Python 进行测试,再将测试图片放入 FPGA 进行测试,如表 3 所示,可以发现在 FPGA 上,准确率几乎没有下降。

4.2 实际图片测试

工程编译完成之后,.BIN 文件会自动在 release 目录下的 sd_card 文件夹下生成。将.BIN 文件拷贝到 SD 卡中,设置开发板为 SD 卡启动模式。连接好终端显示器,

表 3 测试集结果
Table 3 Test set results

类别	FPGA 实现		PC 实现	
	正确率/%	错误数	正确率/%	错误数
裂口缺陷	97.25	7	97.50	6
畸形缺陷	97.25	4	97.50	4
完好	97.00	6	98.00	4

显示器左上角为识别出樱桃结果的黄色标签,试验运行结果部分展示如图 10 所示。

Zynq XC7Z020 资源包括:18 K 大小的 Block RAM 140 个,查找表(LUT) 53 200 个,触发器(FF)106 400 个和数字信号处理器(DSP)220 个,文中 2 个硬件加速函数在硬件中的资源占用使用情况和运行时长如表 4 所示。

输入同样尺寸大小的图片在不同平台下分别进行测

表 4 硬件加速函数资源占用和运行时长

Table 4 Hardware acceleration function resource occupation and running time

硬件加速函数	BRAM_18K/%	DSP48E/%	FF/%	LUT/%	时长/ms
Copy_buffer	0.0	0.0	0.0	0.1	5.8
Mmult	22.0	20.0	15.0	58.0	0.9

表 5 纯 ARM 软件与试验加速器对樱桃图片处理时长比较

Table 5 Comparison of the processing time of pure ARM software and this accelerator for cherry pictures

试验平台	Conv1/ms	Pool1/ms	Conv2/ms	Pool2/ms	FC/ms	缺陷果识别率/%	处理总时长/ms
纯 ARM 软件	5.33	12.81	24.80	13.25	1.70	97.50	57.89
试验加速器	3.72	7.90	9.30	5.18	0.32	97.50	26.42

表 6 CPU 平台与试验加速器对樱桃图片处理时长比较

Table 6 Comparison of the processing time of the cherry platform between the CPU platform and this accelerator (n=64)

试验平台	Conv1/ms	Pool1/ms	Conv2/ms	Pool2/ms	FC/ms	缺陷果识别率/%	处理总时长/ms
文献[4]平台	—	—	—	—	—	97.99	40.00
试验加速器	5.19	9.76	12.80	9.75	5.09	96.25	42.59

由表 6 可以看出,当输入图像尺寸扩大一倍时,一幅图片的处理时长为 42.59 ms,对比原来图像尺寸延长了 16.17 ms,与文献[4]的 CPU 平台处理时长相当,但是试验系统所使用的时钟频率与 CPU 平台相差甚远,若使用性能更好的 FPGA,速度应超过 CPU 平台。而且试验系统具有小型化、手持化的特点,与 CPU 平台相比具有很强的实际应用价值。

5 结论

试验利用 SDSoC 快速开发平台,协同处理 PS 端与 PL 端,实现一个快速樱桃检测的软硬件系统。通过对系统硬件设计及优化,大大提高了系统的计算性能,与纯 ARM 软件和 CPU 实现的算法相比,使用软硬件协同处理对算法的加速效果明显,满足实时性与低成本的要求。



(a) 裂口樱桃 (b) 完好樱桃 (c) 畸形樱桃
图 10 试验运行结果部分展示

Figure 10 Partial display of test run results

试,如表 5 所示。结果表明在纯 ARM 软件中对单张樱桃图片检测处理需要 57.89 ms,而在硬件平台上需 26.42 ms,其检测处理速度是纯 ARM 软件的 2.19 倍。

裴悦琨等^[4]选用 CPU 型号为 intel(R)Core(TM) i5-6500 CPU @ 3.20GHz HP LV2011 的 PC 机对樱桃进行检测,测试实际图片时在网络中将输入图片的尺寸缩放为 64×64,因此将 FPGA 中网络的缩放尺寸改为 64×64,并与文献[4]的处理时间进行比较,如表 6 所示。

试验系统可以不使用 PC 机,便携性大大提高,同时可拓展性好,具有很大的市场空间。

试验系统还有进一步优化的空间,比如增加学习样本种类、建立合理的分类分级逻辑和改进神经网络模型结构等。同时笔者正在尝试其他水果数据应用于此系统,以使其能检测更多水果缺陷,最终形成一个实用的快速自动水果缺陷检测与识别系统。

参考文献

[1] LAUER F, SUEN C Y, BLOCH G. A trainable feature extractor for handwritten digit recognition[J]. Pattern Recognition, 2007, 40(6): 1 816-1 824.

(下转第 165 页)

124 min,此条件下的金花茶叶皂苷得率为4.82%;金花茶叶提取的皂苷对胰脂肪酶的抑制试验表明,皂苷提取物在37℃、质量浓度在6 mg/mL、pH 7.5、反应30 min下,脂肪酶抑制率达到84.76%;并且金花茶叶皂苷浓度与胰脂肪酶的抑制作用呈剂量效应,浓度越高,抑制效果越好,说明金花茶叶皂苷是一种良好的胰脂肪酶抑制剂,具有较好的降血脂应用前景。后续可以利用现代分析方法深入研究金花茶叶皂苷对胰脂肪酶的抑制机理。

参考文献

- [1] 曾秋文. 金花茶皂甙的分离纯化及化学结构研究[D]. 湛江: 广东海洋大学, 2010: 1.
- [2] 张武君, 黄颖桢, 陈青瑛, 等. 综合评分法优化金花茶叶提取工艺及抗氧化活性分析[J]. 中药材, 2020(2): 408-414.
- [3] 朱成豪, 唐辉, 柴胜丰, 等. 嫁接与实生金花茶的叶片和花朵营养成分分析与评价[J]. 食品工业科技, 2019, 40(20): 329-333, 347.
- [4] 秦小明, 宁恩创, 李建强. 金花茶食品新资源的开发利用[J]. 广西热带农业, 2005(2): 20-22.
- [5] 韦英亮, 陈秋虹, 盘杨桂, 等. 金花茶叶皂甙 A 对照品的制备[J]. 食品科技, 2018, 43(1): 227-230.
- [6] 贺栋业, 李晓宇, 王丽丽, 等. 金花茶化学成分及药理作用研究进展[J]. 中国实验方剂学杂志, 2016, 22(3): 231-234.
- [7] 黄永林, 文永新, 陈月圆, 等. 金花茶叶中总黄酮提取及动态变化研究[J]. 食品科学, 2009, 30(6): 72-75.
- [8] 颜栋美, 李仁菊, 丘华, 等. 金花茶多酚提取工艺的研究[J]. 现代食品科技, 2007(9): 45-49.
- [9] 许子竞, 廖敏富, 陈海燕, 等. 金花茶叶多糖超声波辅助提取工艺优化和含量测定[J]. 食品科学, 2010, 31(4): 53-58.
- [10] 冯印, 高超, 于钧宇, 等. 款冬总皂苷提取优化及其降脂抗氧化作用研究[J]. 食品工业, 2018, 39(6): 169-174.
- [11] 龙妍, 张佳玉, 刘红梅, 等. 人参饮品中人参皂甙含量的测定[J]. 现代食品, 2018(2): 77-80.
- [12] 赵瑜, 周家春, 张靖伟, 等. 紫娟茶提取物对血管紧张素转换酶、 α -淀粉酶和胰脂肪酶的体外抑制作用[J]. 食品工业科技, 2017, 38(19): 11-20.
- [13] 张丹, 姚正颖, 侯北伟, 等. 香水莲花提取物对胰脂肪酶活性的抑制作用[J]. 食品科技, 2017, 42(3): 227-231.
- [14] 赵瑜. 紫娟茶中多酚类物质的功能性探究[D]. 上海: 华东理工大学, 2017: 42-43.
- [15] 付盼娟. 无患子皂苷的分离纯化及性能研究[D]. 上海: 上海应用技术大学, 2016.
- [16] 张正茂, 张振伟, 阚玲. 湖北利川山药零余子皂苷的提取工艺研究[J]. 中国食品添加剂, 2018(9): 71-76.
- [17] 李霄, 薛成虎, 弓莹, 等. 响应面分析法优化黑豆中皂苷的提取工艺及其抗氧化性研究[J]. 食品工业科技, 2017, 38(9): 235-241.
- [18] 邵青. 山丹鳞茎总皂甙提取纯化及其抗氧化、抑菌活性研究[D]. 延安: 延安大学, 2016: 18-19.
- [19] 耿杰, 张洪梅, 周泉城. 文冠果壳皂苷抑制胰脂肪酶活性的研究[J]. 现代食品科技, 2014, 30(11): 89-92, 132.
- [20] 冉琳, 何钢, 梁立, 等. 黑木耳多糖对胰脂肪酶活性的抑制作用[J]. 食品工业科技, 2017, 38(22): 56-60.
- [21] 褚盼盼, 门玉倩, 孔令悦, 等. 黑豆红色素抑制胰脂肪酶催化反应动力学研究[J]. 食品科技, 2018, 43(5): 302-307.
- (上接第 134 页)
- [2] SHUSTANOV A, YAKIMOV P. CNN design for real-time traffic sign recognition[J]. Procedia Engineering, 2017, 201: 718-725.
- [3] 刘云, 杨建滨, 王传旭. 基于卷积神经网络的苹果缺陷检测算法[J]. 电子测量技术, 2017, 40(3): 108-112.
- [4] 裴悦琨, 连明月, 姜艳超, 等. 基于机器视觉的樱桃缺陷检测与识别[J]. 食品与机械, 2019, 35(12): 137-140, 226.
- [5] 伍锡如, 雪刚刚, 刘英璇. 基于深度学习的水果采摘机器人视觉识别系统设计[J]. 农机化研究, 2020, 42(2): 177-182, 188.
- [6] 秦东辉, 周辉, 赵雄波, 等. 基于卷积神经网络图像识别算法的加速实现方法[J]. 航天控制, 2019, 37(1): 22-27.
- [7] 王巍, 周凯利, 王伊昌, 等. 卷积神经网络(CNN)算法的FPGA并行结构设计[J]. 微电子学与计算机, 2019, 36(4): 57-62, 66.
- [8] MA Yu-fei, SUDA N, CAO Yu, et al. ALAMO: FPGA acceleration of deep learning algorithms with a modularized RTL compiler[J]. Integration, 2018, 62(6): 14-23.
- [9] 瞿晶, 张旭欣, 金婕. 基于FPGA动态重构的快速车牌识别系统[J]. 传感器与微系统, 2019, 38(12): 69-72.
- [10] DANOPOULOS D, KACHRIS C, SOUDRIS D. Acceleration of image classification with Caffe framework using FPGA[C]// 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCASST). Thessaloniki; IEEE, 2018: 1-4.
- [11] RETTKOWSKI J, BOUTROS A, GHRINGER D. HW/SW Co-Design of the HOG algorithm on a Xilinx Zynq SoC[J]. Journal of Parallel and Distributed Computing, 2017, 109(11): 50-62.
- [12] 吴进, 赵隽, 李聪, 等. 机器视觉中边缘检测算法的SDSoC加速实现[J]. 计算机工程与应用, 2019, 55(12): 208-214.
- [13] Vivado design suite user guide high-level synthesis UG902 (v2017.3) [EB/OL]. [2017-10-04]. <http://china.xilinx.com/support/documentation>.