基于 HLS 工具的 CNN 加速器的设计与优化方法研究*

程佳风，王红亮
（中北大学 电子测量技术国家重点实验室，山西 太原 030051）

摘 要： 基于软硬件协同设计的思想，利用 HLS 工具，在 PYNQ-Z2 平台上设计并实现了一个卷积神经网络加速器，对卷积运算采用矩阵切割的优化方法，均衡了资源消耗和计算资源，使得加速器的性能达到了最优。利用 MNIST 数据集对加速器 IP 核进行性能测试，实验结果表明：对单张图片的测试，该加速器相对于 ARM 平台实现了 5.785 的加速效果，对于 1 000 张图片的测试则可达到 9.72 的加速效果，随着测试图片数量的不断增加，加速器的性能也将越来越优。

关键词： 卷积神经网络；PYNQ-Z2；HLS 工具；加速器
中图分类号： TN108.1　　**文献标识码：** A　　DOI：10.16157/j.issn.0258-7998.200841

中文引用格式： 程佳风，王红亮. 基于 HLS 工具的 CNN 加速器的设计与优化方法研究[J]. 电子技术应用，2021，47(3)：18-21，26.
英文引用格式： Cheng Jiafeng，Wang Hongliang. Research on the design and optimization method of CNN accelerator based on HLS tools[J]. Application of Electronic Technique，2021，47(3)：18-21，26.

Research on the design and optimization method of CNN accelerator based on HLS tools

Cheng Jiafeng，Wang Hongliang
(National Key Laboratory for Electronic Measurement Technology，North University of China，Taiyuan 030051，China)

Abstract： Based on the idea of software and hardware co-design，this article uses HLS tools to design and implement a convolutional neural network accelerator on the PYNQ-Z2 platform，and uses the matrix cutting optimization method for convolution operations to balance resource consumption and computing resources，so that the performance of the accelerator is optimized. This article uses the MNIST data set to test the performance of the accelerator IP core. The experimental results show that：for a single image test，the accelerator achieves an acceleration effect of 5.785 compared with the ARM platform，and an acceleration of 9.72 for a 1000 image test. As a result，as the number of test images continues to increase，the performance of the accelerator will become better and better.

Key words： convolutional neural network(CNN)；PYNQ-Z2；HLS tool；accelerator

0 引言

近年来，卷积神经网络的应用范围越来越广泛，其应用场景也日益复杂，卷积神经网络的计算密集和存储密集特征日益凸显，成为快速高效实现卷积神经网络的限制。于是基于 GPU[1]、ASIC[2]、FPGA[3]的不同的加速器平台被相继提出以提升 CNN 的设计性能。GPU 的电力消耗巨大，硬件结构固定，限制了卷积神经网络在嵌入式设备的应用；ASIC 开发成本极高，灵活性低，不适合搭载复杂多变的卷积神经网络；FPGA 具有功耗低、性能高、灵活性好的特点，因此更加适用于卷积神经网络硬件加速的开发研究，但由于 Verilog HDL 开发门槛高，开发周期相对较长，影响了 FPGA 在卷积神经网络应用的普及[4-5]。

本文基于软硬件协同的思想，利用 HLS 工具，在 PYNQ-Z2 上实现了一个卷积神经网络加速器，并采用矩阵切割的设计方法对卷积核运算进行优化。

1 PYNQ-Z2 和卷积神经网络

本设计采用 Xilinx 公司推出的 PYNQ-Z2 开发板作为实验平台。PYNQ-Z2[6-9]是基于 Xilinx ZYNQ-7000 FPGA 的平台，除继承了传统 ZYNQ 平台的强大处理性能外，还兼容 Arduino 接口与标准树莓派接口，这使得 PYNQ-Z2 具有极大的可拓展性与开源性。PYNQ 是一个新的开源框架，使嵌入式编程人员无需设计可编程逻辑电路即可充分发挥 Xilinx Zynq All Programmable SoC

* 基金项目：山西省“1331 工程”重点学科建设计划项目(1331KSC)

(APSoC)的功能。与常规方式不同的是,通过 PYNQ-Z2,用户可以使用 Python 进行 APSoC 编程,并且代码可直接在 PYNQ2 上进行开发和测试。通过 PYNQ-Z2,可编程逻辑电路将作为硬件库导入并通过其 API 进行编程,其方式与导入和编程软件库基本相同。

卷积神经网络[10-13]是一种复杂的多层神经网络,擅长处理目标检测、目标识别等相关的深度学习问题。卷积神经网络通过其特有的网络结构,对数据量庞大的图像识别问题不断地进行图像特征提取,最终使其能够被训练。一个最典型的卷积神经网络由卷积层、池化层、全连接层组成。其中卷积层与池化层配合,组成多个卷积组,逐层提取特征,最终通过全连接层完成图像的分类任务。卷积层完成的操作可以认为是受局部感受野概念的启发,而池化层主要是为了降低数据维度。综合起来,CNN 通过卷积来模拟特征区分,并且通过卷积的权值共享及池化来降低网络参数的数量级,最后通过传统神经网络完成分类等任务。

本文采用一种典型的手写数字识别网络 CNN LeNet5 模型[14-16]对系统进行测试,模型结构如图 1 所示,总共包含 6 层网络结构:两个卷积层、两个池化层、两个全连接层。网络的输入为 28×28×1 像素大小图片,输入图像依次经过 conv1、pool1、conv2、pool2、inner1、relu1、inner2 层后,得到 10 个特征值,然后在 softmax 分类层中将 10 个特征值概率归一化得出最大概率值即为分类结果。网络中的具体参数设置如表 1 所示。

由表 1 可以计算出,该 CNN 网络总共的权重参数量为 260+5 020+16 050+510=21 840 个变量。若将这 21 840 个变量都采用 ap_int(16)来存储,将大约消耗 43 KB 的存储资源,本文采用的 PYNQ-Z2 有足够的存储空间用于存放这些变量。

2 系统设计与实现

本文设计并实现基于 PYNQ-Z2 的 CNN 通用加速器,采用 PYNQ-Z2 的 PS 部分做逻辑控制,PL 部分执行卷积神经网络运算。由于全连接运算是特殊的卷积运算,

表 1 网络参数表

网络层	输出尺寸	卷积核大小\步进	补零	权重参数量
input	28×28×1			
conv1	10×24×24	5×5\1	0	260
pool1	10×12×12	2×2\2		
conv2	20×8×8	5×5\1	0	5 020
pool2	20×4×4	2×2\2		
interl	50×1×1	4×4		16 050
inter2	10×1×1	1×1		510
softmax	1			

因此依据卷积神经网络的特性设计了两个通用的运算模块,即通用的卷积运算模块和通用的池化运算模块,如图 2 所示。

由图 2 可以看出,这种加速器框架实现了两种通用的加速电路(即通用的卷积运算电路和通用的池化运算电路),CPU 通过 axi_lite 总线对卷积池化电路的参数进行配置,卷积池化电路通过 axi_hp 总线对 CPU 中存储的特征权重参数进行读取。当存储器中输入一组数据的时,CPU 就会进行参数配置并调用卷积运算模块进行运算,卷积 ReLU 后的结果保存在存储器中再进行参数配置并调用池化运算模块进行运算,可以通过这种循环运算的方式实现卷积神经网络的运算。

2.1 CNN LeNet5 模型训练

本文在 TensorFlow 中搭建 CNN LeNet5 网络模型并进行训练,训练过程如图 3 所示。其中横坐标轴代表训练次数,纵坐标轴表示每次训练的误差。设置训练速率为 50,训练 20 000 次,随着训练次数的不断增多,误差逐渐减小,最后的模型错误率仅为 1.58%。

2.2 CNN 加速器的 IP 核的设计与实现

Xilinx 推出的 HLS[17]工具是基于 FPGA 的设计与开发,用户可以选择多种不同的高级语言(如 C、C++、System C)来进行 FPGA 的设计,在代码生成时可以快速优化 FPGA 硬件结构,提高执行效率,降低开发难度。

本文通过 C 语言描述了两个加速电路,利用 HLS 工

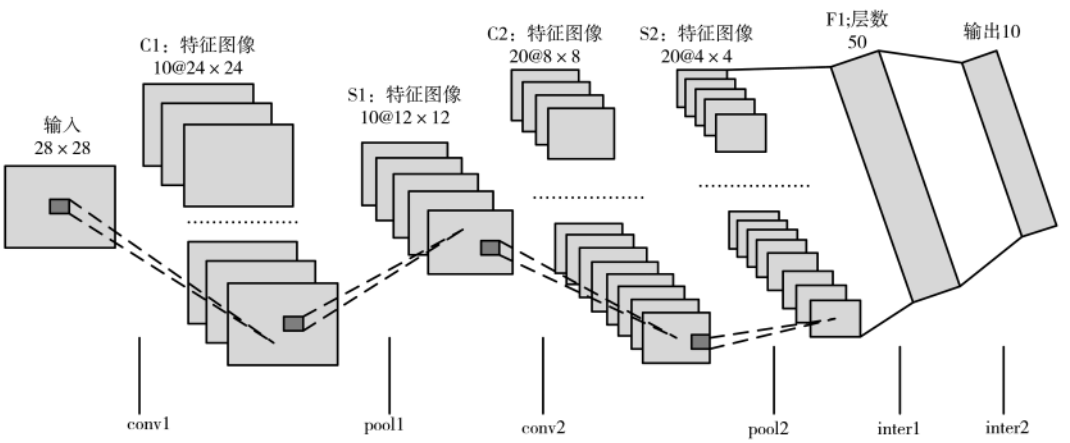

图 1 典型的手写数字识别网络结构

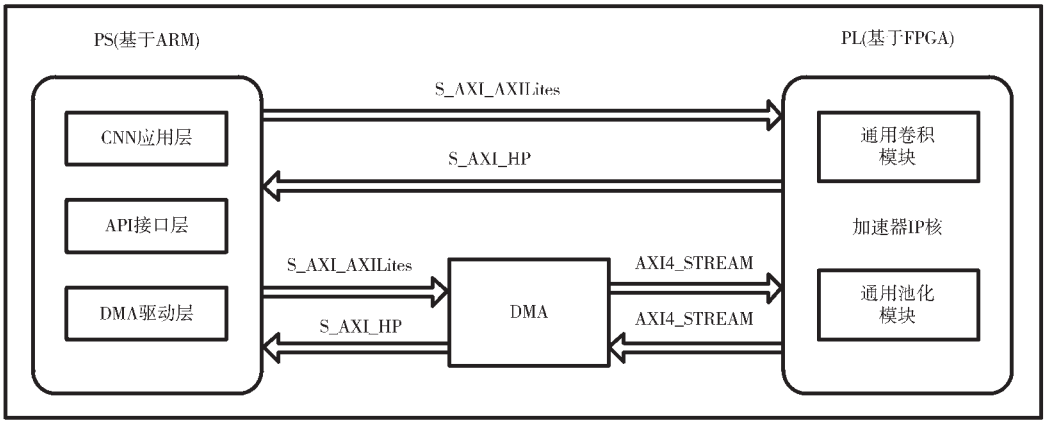

图 2　系统硬件原理框图

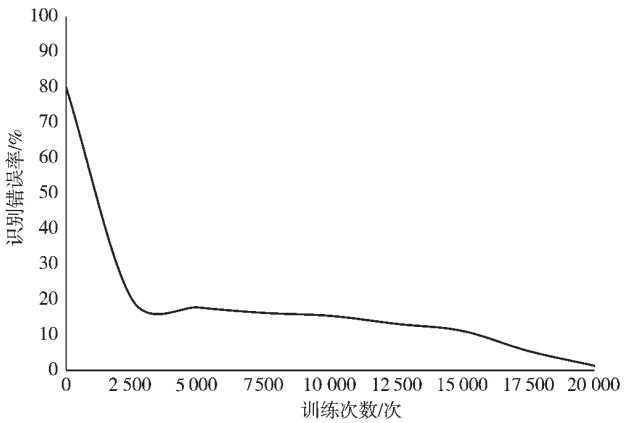

图 3　CNN LeNet5 误差与训练次数的关系

具生成加速器的 IP 核。系统通过 CPU 配置 IP 核的参数，采用 AXI 的通信方式进行数据传输，输入的数据通过 IP 核进行 CNN 运算，运算的结果通过 AXI 总线输出。图 4 是加速器 IP 核的原理图。

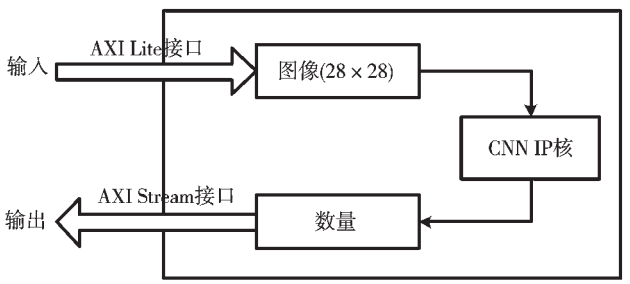

图 4　加速器 IP 核原理图

2.3 CNN 加速器的 IP 核优化

由于特征、权重参数都是多维的空间变量，无法在计算机中读取，因此需要将其展开为一维变量。如图 5 所示，对于特征参数，它在空间中的排布方式为三维变量，因此需要将其展开为一维变量，考虑到 FPGA 的并行计算能力优秀，所以在空间中沿输入特征的通道 C 将其切割为 C/k 通道，每一个通道可以实现 k 路并行的计算且需要的特征存储空间减少，大大提高了加速电路的运算效

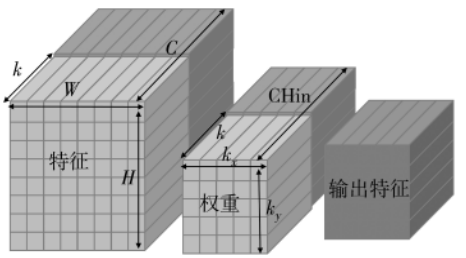

图 5　卷积运算的矩阵切割

率，节约了 FPGA 的存储资源。特征参数经过切割后，它在内存中的排布方式变为了一维变量：$[C/k][H][W][k]$。

权重参数在空间中的排布方式为四维变量，要将它展开为一维变量，也是对其输入通道 CHin 切割为 CHin/k 通道，实现每一个通道的 k 路并行，它在内存中的排布方式变为一维变量：$[\text{CHout}][k_y][k_x][\text{CHin}/k][k]$。

其中 k 的取值会对系统性能造成极大的影响，一个合适的 k 值可以使得存储资源、计算资源、带宽资源三者达到平衡。通常 k 的取值有 8、16、32、64，其不同的取值对应的资源消耗如表 2 所示，不同取值对应的计算图片的时间如表 3 所示。

通过表 2 和表 3 的数据对比可以得到，在 $k=32$ 时的资源占用较为合理，可以使得系统的性能达到最优，同时权衡了计算时间和数据存储时间，达到了比较好的均衡效果。

表 2　不同 k 的取值对应的资源消耗

k 的取值	BRAM 占用块数	BRAM 占用比率/%
8	42	15
16	79	28
32	148	53
64	275	98

3 实验与结果

3.1 实验测试平台

本实验采用 PYNQ-Z2 开发板，其主芯片是 XC7Z020，主要由 PS 和 PL 两部分组成，PS 端是 650 MHz 双核

《电子技术应用》http://www.chinaaet.com

表 3 不同 k 的取值对应的计算图片的时间

运行方式	k 的取值	消耗的时钟周期数	实际消耗的时间/ms
ARM	默认	4 713 245	47
加速器 IP 核	8	2 900 456	29
加速器 IP 核	16	1 743 567	17
加速器 IP 核	32	811 378	8
加速器 IP 核	64	811 378	8

Cortex-A9 处理器，PL 端的时钟频率为 100 MHz。通过表 3 可以看出，在进行单张图片测试时，CNN IP 核在 k=32 时计算图片的时间比 PS 端减少了将近 39 ms，达到了近 6 倍的加速效果。接下来进行多张图片测试来记录加速效果。

3.2 图片流测试

在 MNIST 数据集中选取 1 000 张图片，分成 10 组，每组 100 张测试图片，组成图片流，分别送入 ARM 层和硬件层的加速器 IP 核进行 CNN 运算，并且记录各自所用的时间，从而得到加速器 IP 核对图片流的加速效果。实现结果如图 6 所示。

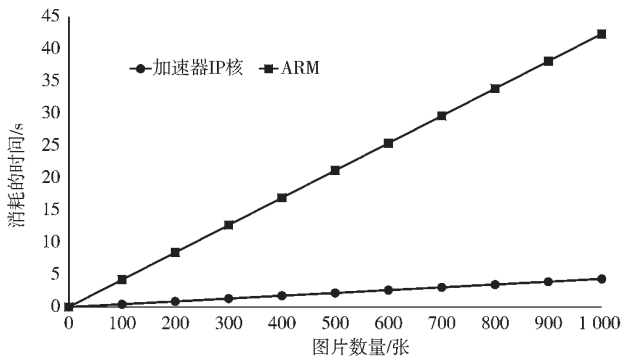

图 6 图片流的测试结果

由图 6 可以看出，两种不同平台的测试结果都成线性关系，说明每张图片的运算时间都是固定不变的，加速器 IP 核处理单张图片的平均时间为 4.3 ms，而 ARM 平台处理单张图片的平均时间约为 42 ms，由此可见，当运算相同数量的图片时，CNN IP 核可将运算速度提高到近 10 倍，远远超过了单张图片的加速效果。当处理 1 000 张图片时，加速器 IP 核比 ARM 端快了 38 s 左右，并且随着图片的数量越来越多，加速器 IP 核的性能也将越来越好，加速效果也将越来越显著。

3.3 实验结果比较

本文采用的 HLS 工具实现的加速器 IP 核与 FPGA 实现卷积神经网络[18]相比较，比较结果如表 4 和表 5 所示。

通过表 5 可以看出，文献[18]中提出的利用 FPGA 实现 CNN 加速器与传统的 CPU 相比有很大的加速效果，在处理单张图片时的加速比为 4.17，在处理 10 000 张图片时的加速比为 3.43，可见随着处理图片的数量逐渐增加，加速器的效果在不断降低。本文提出的利用

表 4 文献[18]与本文实验平台的对比

	FPGA	通用 CPU	实现方式	时钟频率
文献[18]		Corei5 2500K		3.3 GHz
	Virtex-5 XC5VLX110T		Verilog HDL	75 MHz
本文		Cortex-A9		650 MHz
	Zynq XC7Z020		HLS 工具	100 MHz

表 5 文献[18]与本文对不同数量的图片处理的运算速度的对比

	项目	图片数量/张		
		1	1 000	10 000
文献[18]	CPU 耗时/ms	1.13		$1.210\ 3\times10^4$
	FPGA 耗时/ms	0.272		3.522×10^3
	加速比	4.17		3.43
本文	CPU 耗时/ms	47	$4.232\ 2\times10^4$	
	FPGA 耗时/ms	8	4.352×10^3	
	加速比	5.875	9.72	

HLS 工具生成的 CNN 加速器在处理单张图片时所耗时间为 8 ms，加速比为 5.875，与传统 CPU 相比有很好的加速效果；在处理 1 000 张图片时，加速器 IP 核耗时 4.352 s，通用 CPU 耗时 42.322 s，此时的加速比为 9.72，加速效果越来越明显，并且随着处理的图片数量越来越多，加速效果会越来越好，具有很好的参考意义。

4 结束语

本文在 PYNQ-Z2 平台上利用 HLS 工具设计了加速器 IP 核来进行卷积神经网络运算，并通过矩阵切割的方法对加速器 IP 核进行优化，充分利用了 FPGA 的并行计算能力。通过实验证明在 k=32 时，均衡了存储资源和计算资源，使得加速器 IP 核的性能达到最优，运算速度得到明显的提升。由于本实验采用的开发平台资源有限，若采用资源更多的 FPGA 平台进行加速运算，加速器的性能将得到更大的提升。

参考文献

[1] GONG L, WANG C, LI X, et al. Maloc: a fully pipelined fpga accelerator for convolutional neural networks with all layers mapped on chip[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2018, 37(11): 2601-2612.

[2] MARTINEZ B, VALSTAR M F. Advances, challenges, and opportunities in automatic facial expression recognition[M]. Advances in Face Detection and Facial Image Analysis. Springer, 2016.

[3] VENIERIS S I, BOUGANIS C. FpgaConvNet: mapping regular and irregular convolutional neural networks on FPGA[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(2): 326-342.

[4] DAI Y, LIU D, WU F. A convolutional neural network

(下转第 26 页)

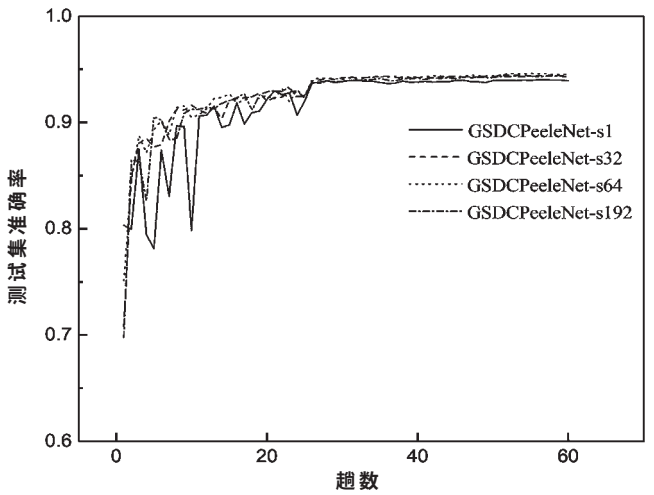

图 5 GSDCPeleeNet 各模型在 Fashion-MNIST 的准确率曲线

别效果差距不大甚至更佳。但由于 PeleeNet 网络结构的限制，GSDCPeleeNet 的网络参数和计算量还是较大。在接下去的工作中，希望通过结合其他轻量卷积神经网络结构和压缩方法，继续改进该网络。在进一步的工作中，还将会研究该网络在其他计算机视觉领域的应用，如目标检测和图像识别等。

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]. International Conference on Neural Information Processing Systems. Curran Associates Inc., 2012: 1097-1105.

[2] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge[J]Nnternational Journal of Computer Vision, 2015, 115(3): 211-252.

[3] SZEGEDY C, LIU W, JIA Y, et al. Going deeperwith convoutions[C]. CVPR 2015.7298594, 2015.

[4] HUANG G, LIU Z, LAURENS V D M, et al. Densely connected convolutional networks[C]. CVPR 2017, 2017: 4700-4708.

[5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]. Las Vegas: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[6] ZHANG X, ZHOU X, LIN M, et al. Shufflent: an extremely efficient convolutional neural network for mobile devices[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

(下转第 30 页)

(上接第 21 页)

approach for post-processing in HEVC intra coding[C]. International Conference on Multimedia Modeling, 2017.

[5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[6] AIMAR A, MOSTAFA H, CALABRESE E, et al. NullHop: a flexible convolutional neural network accelerator based on sparse representations of feature maps[J]. IEEE Transactions on Neural Networks, 2019, 30(3): 644-656.

[7] RAJASEGARAN J, JAYASUNDARA V, JAYASEKARA S, et al. DeepCaps: going deeper with capsule networks[C]. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2019: 10725-10733.

[8] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[9] HORI T, WATANABE S, ZHANG Y, et al. Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM[C]. Interspeech 2017, 2017.

[10] Yang Yichen, Zhang Guohe, Liang Feng, et al. Design of FPGA based convolutional neural network co-processor[J]. Journal of Xi'an Jiaotong University, 2018, 52(7): 158-164.

[11] 崔小乐，陈红英，崔小欣，等. 一种软硬件协同设计工具原型及其设计描述方法[J]. 微电子学与计算机，2007，24(6)：28-30.

[12] 吴艳霞，梁楷，刘颖，等. 深度学习 FPGA 加速器的进展与趋势[J]. 计算机学报，2019(11)：2461-2480.

[13] 卢冶，陈瑶，李涛，等. 面向边缘计算的嵌入式 FPGA 卷积神经网络构建方法[J]. 计算机研究与发展，2018，55(3)：551-562.

[14] 魏浚峰，王东，山丹. 基于 FPGA 的卷积神经网络加速器设计与实现[J]. 中国集成电路，2019，28(7)：18-22.

[15] 施一飞. 对使用 TensorRT 加速 AI 深度学习推断效率的探索[J]. 科技视界，2017(31)：26-27.

[16] 杨一晨，张国和，梁峰，等. 一种基于可编程逻辑器件的卷积神经网络协处理器设计[J]. 西安交通大学学报，2018，52(7)：158-164.

[17] 张哲，孙瑾，杨刘涛. 融合多层卷积特征的双视点手势识别技术研究[J]. 小型微型计算机系统，2019，40(3)：646-650.

[18] 余子健，马德，严晓浪，等. 基于 FPGA 的卷积神经网络加速器[J]. 计算机工程，2017，43(1)：109-114.

(收稿日期：2020-08-13)

作者简介：

程佳风(1995-)，男，硕士研究生，主要研究方向：人工智能、硬件加速。

王红亮(1978-)，男，博士研究生，教授，主要研究方向：测试系统集成、目标检测与识别。

版权声明