DOI: 10.19650/j.cnki.cjsi.J2107921

语义分割网络的 FPGA 加速计算方法综述

彭 宇,姬森展,于希明,刘胜剑
(哈尔滨工业大学测控工程系 哈尔滨 150080)

摘 要:随着深度学习技术的发展和图像场景理解需求的提升,基于现场可编程门阵列(field programmable gate array, FPGA)部署语义分割网络,为用户提供低延迟、高能效的边缘端智能服务成为研究热点。针对语义分割网络结构中计算和存储密集型特点,构建基于 FPGA 的定制计算结构是研究的重点问题。鉴于此,本文在归纳总结语义分割网络基本原理和计算结构特点的基础上,分别从面向硬件资源约束的模型压缩方法和定制硬件架构设计两个角度阐述基于 FPGA 的语义分割网络加速计算方法,并重点对硬件架构设计中的计算结构设计和内存访问优化的典型方法进行总结。最后,展望了语义分割网络 FPGA 加速计算方法的发展趋势,以期为语义分割、边缘计算、定制高能效计算以及其他相关领域的研究者提供设计参考。

关键词: 语义分割;边缘计算;现场可编程门阵列;模型压缩;硬件加速
中图分类号: TP183 TH89 **文献标识码**: A **国家标准学科分类代码**: 510.4050

A review of FPGA-accelerated computing methods for semantic segmentation network

Peng Yu, Ji Senzhan, Yu Ximing, Liu Shengjian
(*Department of Test and Control Engineering, Harbin Institute of Technology, Harbin 150080, China*)

Abstract: With the development of deep learning technology and the increasing demand for image scene understanding, the application of semantic segmentation networks based on FPGA to provide low-latency and high-energy-efficiency edge-end intelligent services for all users has become a research hotspot. The computing and storage of the semantic segmentation network structure have the intensive feature. To address this issue, the construction of a customized FPGA-based computing structure is a key research issue. In view of this, this paper summarizes the basic principles of semantic segmentation networks and analyzes the characteristics of its internal calculation structure, then elaborates FPGA-based semantic segmentation network computing acceleration methods from two perspectives: model compression methods with hardware resource constraints and custom hardware architecture design. Furthermore, this paper focuses on a summary and analysis of typical methods of computing structure design and memory access optimization in hardware architecture design. Finally, this paper looks forward to the future development trend of FPGA-accelerated computing methods for semantic segmentation networks, in order to provide design references for researchers in semantic segmentation, edge computing, customized energy-efficient computing and other related fields.

Keywords: semantic segmentation; edge computing; field programmable gate array; model compression; hardware acceleration

0 引 言

图像语义分割(image semantic segmentation, ISS)是指为图像中的每一个像素分配一个预先定义好的表示其语义类别的标签从而对图像进行标注分割[1]。它是计算机视觉领域的关键问题之一,是场景理解的基础性技术,在自动驾驶、人机交互、安防监控、医学图像研究、遥感图像处理以及虚拟现实和增强现实系统中起到至关重要的作用[2]。

传统的图像分割方法包括阈值分割法、边缘检测法、区域生长法等,这些方法只能利用图像的颜色、纹理和形状等低级语义信息,在面对复杂场景时难以取得良好的分割效果[3]。近年来,以卷积神经网络(convolutional neural

收稿日期:2021-05-13 Received Date: 2021-05-13

network, CNN)为代表的深度学习(deep learning, DL)技术飞速发展,在计算机视觉、医学成像和自然语言处理等领域取得巨大成功,基于深度学习的图像语义分割(image semantic segmentation based on deep learning, ISSbDL)方法也随之迅猛发展,分割的精度远远超过传统方法[4]。

然而,ISSbDL 方法中精度较高的语义分割网络往往都有着数千万乃至上亿的参数量和百亿次的浮点运算操作,需要上千兆字节的内存空间。为了满足深度学习的高存储和高计算需求,研究人员一般采用云计算的模式,依靠运算能力强大的云服务器进行语义分割网络的训练和推理。在这种计算模式下,用户需要将图像数据从数据源(各类边缘设备,如监控摄像头或者智能手机等)上传到云端服务器,会带来诸如高延迟、高带宽、可靠性隐患和隐私泄露等问题,从而限制 ISSbDL 方法在自动驾驶、智能家居、智慧城市等实时性、带宽和隐私安全要求较高的场景中的应用[5]。

为了解决以上问题,研究人员尝试采用边缘计算的模式去执行靠近数据源和用户的计算任务。网络的训练仍放在云端,但训练得到的网络模型直接部署在边缘设备进行推理。边缘计算模式的主要挑战是如何在资源和功耗都受限的边缘设备中满足语义分割网络的高计算和高存储需求。基于通用架构的 CPU 平台难以提供足够的计算能力和内存带宽,因此研究人员通常采用专用硬件平台对模型的推理阶段进行优化加速设计,实现模型在边缘设备中的高效部署和应用[6]。

主流的面向边缘端的硬件加速平台有 3 类,分别是:嵌入式图像处理器(embedded graphics processing unit, 嵌入式 GPU)、现场可编程门阵列(field programmable gate array, FPGA)和专用集成电路(application-specific integrated circuit, ASIC))[7]。嵌入式 GPU 的计算性能远超 CPU 等通用处理器,支持诸如 TensorRT 等深度学习框架,学习和使用成本低,但是它的能效比较差,在边缘设备的低功耗限制下可提供的计算性能有限[8]。ASIC 在性能、功耗和成本上都有优势,但是它必须针对特定算法设计专用电路,其硬件结构生成后便无法更改,开发周期长且通用性受限。相比于以上两种平台,FPGA 具有逻辑资源高度并行和可重构的特点,可以在不影响模型准确性的前提下,充分发挥神经网络的并行性,优化数据传输路径,进而加快模型的推理速度,提高模型推理运算过程的吞吐量和能效比,因此,基于 FPGA 的神经网络推理加速方法正在成为研究的热点方向[9]。

近年来,学者们针对 ISSbDL 方法的关注点一般在于该方法的基本原理、分类、数据集和评价指标等方面,比如文献[2]针对在复杂环境下的图像语义分割任务,详细分析和对比近些年来的各种面向复杂环境的图像语义分割方法;文献[4]将基于深度学习的语义分割方法分为两大类,详细介绍了每类方法的基本思想和优缺点,系统地阐述深度学习对图像语义分割领域的贡献。尽管上述工作为相关研究人员提供了较好的研究思路,但研究者较少关注 ISSbDL 方法的计算模式和部署平台,尤其缺乏面向边缘设备应用过程中,基于定制化高能效计算平台实现 ISSbDL 方法部署和加速计算方法的总结分析。综上所述,本文在对 ISSbDL 方法的基本思想和应用领域归纳分析的基础上,进一步总结在语义分割网络部署和应用过程中计算模式和硬件平台的发展,重点对面向边缘计算的语义分割网络 FPGA 加速计算方法进行调研分析,最后对语义分割网络在边缘设备中部署和应用的前景进行展望,以供相关领域研究人员参考学习。

1 基于深度学习的图像语义分割发展概述

1.1 图像语义分割方法发展现状

图像分割是指根据一定的相似性准则将图像划分成不同区域的过程,是计算机视觉、图像处理等领域的基础问题之一,自 20 世纪 60 年代以来一直都是研究的热点[3]。传统的图像分割方法往往是基于图像的低层语义信息,如图像像素的颜色、纹理和形状等,常用的方法包括阈值法、边缘法和区域法等。而图像的语义分割是在图像分割的基础上发展而来的,它不仅要求分割出图像中目标的轮廓,还需要识别出目标的类别,也就是识别出图像的高级语义信息[10]。语义分割任务需要在图像中正确识别不同的离散对象并且标记出其语义信息,然而目标图像的背景往往复杂多变,物体对象会受到诸如光照、遮挡等环境因素的影响,这都增加了语义分割的难度,使得传统的图像分割方法在复杂环境下的应用受到限制[11]。

因此,研究人员开始将以 CNN 为代表的深度学习技术引入到图像语义分割领域,其中最具开创性和代表性的工作是 Long 等[12]于 2014 年提出的全卷积神经网络(fully convolutional networks, FCN)。它在传统 CNN 的基础上进行改进,将其中的全连接层替换成卷积层,使得网络能够接受任意尺寸的图片且输出的结果是一张图片而不是一组特征向量。FCN 将神经网络对图像的识别精度从图像级提升到像素级,极大地提高了语义分割的精度[13]。研究人员以此为基础提出了 U-Net[14]、DeepLab 系列[15-17]、SegNet[18]、PSPNet[19]等多种语义分割网络。相比于传统方法,ISSbDL 方法学习的特征更丰富、表达能力更强,已经成为图像语义分割领域的主流方法[4]。

1.2 语义分割网络基本组成结构

现有的语义分割网络一般要选取一个具有良好分类效果的 CNN 作为骨干网络,然后根据语义分割任务的特

点对骨干网络进行优化,从而使其能够对图像的每个像素进行识别预测,常用的骨干网络包括 VGGNet[20]、AlexNet[21]、ResNet[22]等。分类精度高的骨干网络往往有着庞大的参数量和计算量,以 VGG-19 为例,在对一个分辨率为 224×224 的图片进行分类时,它拥有高达 1.44 亿个的浮点参数和 390 亿次的浮点运算次数,需要超过 500 MB 的存储空间。语义分割网络需要在分类网络获得的类别标签的基础上进一步输出目标的位置信息,即将类别标签分配给图像中的每一个像素,因此语义分割网络的参数量和计算量一般比 CNN 更大,在实际应用中面临着巨大的存储和计算压力。鉴于此,本节针对典型语义分割网络的结构进行分析,总结其计算和存储的瓶颈所在。

由于语义分割网络是在 CNN 的基础上优化得来的,所以它既包括卷积层、池化层和激活层等传统 CNN 的基本结构,也包括反卷积层、条件随机场(conditional random field, CRF)和跳跃连接等特定结构。而 CNN 的计算量主要集中在卷积层,以 VGG-11 为例,此模型中有 98.2% 的计算操作来自卷积层[9]。此外,语义分割网络中反卷积层同样是计算密集型任务,以 U-Net 为例,此模型中卷积层的计算量约占总计算量的 82%,反卷积层约占到 16%[23]。除此之外的池化层、激活层、批归一化(batch normalization, BN)层等网络结构对语义分割网络在存储和计算上的贡献很小。

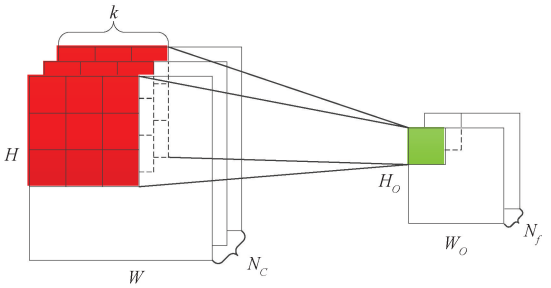
卷积层是 CNN 进行特征提取的关键结构。神经网络的卷积层是指利用卷积核,针对一组输入特征图进行二维卷积运算得到相应的输出特征图的过程[24]。设输入特征图的通道数为 N_C,长宽分别为 H 和 W,输出特征图的通道数为 N_f,长宽分别为 H_O 和 W_O,卷积核大小为 k,步长为 s,补零的个数为 p。卷积层的输入是一组大小为 $N_C \times H \times W$ 的输入特征图和一组大小为 $N_C \times N_f \times k \times k$ 的卷积核,最终得到一组大小为 $N_f \times H_O \times W_O$ 的输出特征图,其中 H_O 和 W_O 可由式(1)和(2)推导出。

$$H_O = (H + 2 \times p - k)/s + 1 \tag{1}$$

$$W_O = (W + 2 \times p - k)/s + 1 \tag{2}$$

卷积操作的具体实现过程如图 1 所示。图 1(a)表示输入特征图,图 1(b)是经过卷积运算得到的输出特征图。一组大小为 $N_C \times k \times k$ 的卷积核和对应输入特征图的一组参数进行卷积运算,最终在输出特征图的对应位置上生成一个像素。假设这一组卷积核权重参数为 $(w_1, w_2, w_3, w_4, \cdots, w_n, n = N_C \times k \times k)$,对应的输入特征图参数为 $(x_1, x_2, x_3, x_4, \cdots, x_n, n = N_C \times k \times k)$,偏置参数为 b,输出特征图参数为 y,则 y 可由式(3)得到。

$$y_{w,b} = \sum_{i=1}^{n} x_i \times w_i + b \tag{3}$$



(a) 输入特征图　(b) 输出特征图
(a) Input feature　(b) Output feature

图 1　卷积层计算过程

Fig. 1　Computing process of convolutional layers

在得到一个输出特征图参数之后,卷积核窗口会在输入特征图上滑动直至遍历所有参数,共计 N_f 组卷积核重复以上操作最终形成 N_f 张输出特征图。假设卷积运算窗口滑动的步长 $s = 1$,每个卷积层的参数量为 P,每个卷积层的计算量,即卷积层内乘法运算和加法运算的总和为 $FLOPs$,则 P 和 $FLOPs$ 可分别由式(4)和(5)得到。

$$P = N_f \times (k^2 \times N_c + 1) \tag{4}$$

$$FLOPs = 2 \times N_f \times k^2 \times N_c \times H_o \times W_o \tag{5}$$

由以上两个公式可知,卷积层的参数量和计算量由卷积核尺寸、输入特征图通道数、输出特征图通道数和输入图片的尺寸等参数共同决定,因而可以通过减少以上几个参数实现对卷积运算的优化。例如 VGGNet[20]中分别使用 2 个和 3 个大小为 3×3 的小卷积核堆叠来代替大小为 5×5 和 7×7 的大卷积核,显著减少了卷积层的参数量和计算量。

语义分割网络中的反卷积(deconvolution)又称转置卷积(transposed convolution),负责对图像进行上采样,恢复图像的尺寸[25]。假设反卷积的卷积核大小为 k',步长为 s',补零的个数是 p',那么这个反卷积运算过程就等同于卷积核大小为 k',步长为 1,补零的个数是 $k' - p' - 1$,并且在输入特征图的各个元素之间补 $s' - 1$ 个零的卷积运算。当反卷积层的输入是一个 $N_C \times H \times W$ 的输入特征图和一组 $N_C \times N_f \times k \times k$ 的卷积核时,最终得到一组 $N_f \times H'_O \times W'_O$ 的输出特征图,其中 H'_O 和 W'_O 可由式(6)和(7)推导出。

$$H'_O = s' \times (H - 1) + k' - 2 \times p' \tag{6}$$

$$W'_O = s' \times (W - 1) + k' - 2 \times p' \tag{7}$$

通过上述变换,反卷积运算可以被转置成为常规的卷积运算,其参数量和计算量同样可由式(4)和(5)计算得出。

1.3　面向 ISSbDL 计算模式发展

自 2006 年以来,由 Google、Amazon 等公司率先提出的“云计算”模式得到快速发展[26]。根据美国国家标准

与技术研究院(national institute of standards and technology, NIST)的定义[27],云计算是基于互联网实现对共享资源(如服务器、存储设备、应用程序等)随时随地、按需和便捷的访问。云计算具有资源丰富、灵活性强、可扩展性强、按需部署和成本低廉等优点,能够满足ISSbDL方法的高存储和高计算需求,因而早期大部分基于深度学习的语义分割算法在应用过程中都采用此种计算模式。但在云计算模式下,必须将数据从靠近数据源的边缘设备上传云端服务器,会带来诸如高延迟、高带宽、可靠性隐患和隐私泄露等问题[5]。随着人工智能生态的爆发,在自动驾驶等众多应用领域,基于云计算模式难以给用户提供低延迟和高可靠的服务。为此,研究人员引入一种新的计算模式:边缘计算。

2016 年 5 月,施巍松教授首先给出了边缘计算的正式定义:边缘计算是指在网络边缘,使用云服务的下游数据和物联网服务的上游数据进行计算的技术[28]。相比于云计算,面向边缘计算的语义分割不需要将数据上传到云端,响应时间短,带宽需求低,可靠性高,适合于自动驾驶、安防监控等实时性要求高,应用规模大的领域[5]。同时,将数据存储在边缘设备可以降低隐私泄露的风险,有利于诸如智慧城市和智能家居等需要用户私密数据的应用的发展[29]。但边缘设备的计算能力、存储能力、带宽和功耗都远低于云端服务器,在边缘设备上部署和应用语义分割网络就必须在资源受限的情况下满足其高存储和高计算需求。针对以上问题,研究人员在边缘设备中部署和应用语义分割网络时通常会使用专用硬件平台进行加速设计,提供一个兼顾高性能和高能效的解决方案。

主流的面向边缘端的硬件加速平台有 3 种:嵌入式 GPU、FPGA 和 ASIC[30]。由于硬件结构的不同,它们在加速深度学习应用中有着各自的优势和劣势。

1)嵌入式 GPU。计算能力强,带宽大,适合于大规模数据的并行计算,支持多种深度学习的开发框架和开发环境,学习门槛和应用难度较低,但是计算过程中包含大量与外界的存储交互导致其能效比低。

2)FPGA。包含大量并行逻辑单元,可以通过流水线和并行计算充分发掘神经网络的并行性,同时具有可重构性的特点,可以灵活地根据算法设计硬件结构,硬件实现的能效比要优于 GPU。但是 FPGA 的成本较高,缺乏类似 Caffe 或 Tensorflow 等专门用于深度学习开发的工具链,必须使用 Verilog 或者 VHDL 等硬件描述语言进行开发,开发周期和复杂度较高。

3)ASIC。根据算法的需求进行定制设计的集成电路,能在特定功能上强化,相比于 FPGA 拥有更强的性能和更低的功耗。但是 ASIC 的设计和制造流程复杂,投入巨大且耗时长,同时硬件电路生成之后无法改变,无法为处在快速发展的深度学习算法提供通用的解决方案。

综上所述,FPGA 凭借其高性能、高能效比和灵活可重构的优点,成为在资源和功耗受限的边缘设备中部署语义分割网络的重要定制化计算加速平台。

2 基于 FPGA 的语义分割网络加速优化方法

为利用 FPGA 高能效比和灵活可重构的优势,实现语义分割网络在边缘设备的高能效部署,当前研究者[23,31-41]主要从语义分割网络算法的轻量化优化与结合定制化硬件平台资源特点的加速计算两方面进行研究分析。

在语义分割网络算法的轻量化优化方面,最常用的方法是模型压缩。模型压缩通过对原模型的精简,降低其参数量和计算量,最终缩短模型的推理用时,降低运算和存储中的能量消耗[42]。模型压缩方法一般不限定具体应用的硬件平台,但部分方法能够结合 FPGA 硬件资源,通过设计特定的硬件结构取得更好的压缩效果,本文将这类压缩方法统称为面向硬件资源约束的模型压缩方法。而结合硬件平台的加速计算所采用的方法一般是基于 FPGA 的特定硬件架构设计,其优势主要体现在以下两个方面:第一,语义分割网络的层与层之间高度独立,层间无数据反馈;各层运算具有并行性和相似性,尤其是卷积层和反卷积层,在同一特征图的滑窗运算和不同特征图之间都能实现并行计算[43]。而 FPGA 具有大量的并行逻辑单元,可以通过并行计算和流水线等方法充分发掘语义分割网络的并行性,从而加快模型的推理速度。第二,FPGA 具有可重构的特点,通过烧入配置文件定义其逻辑单元的布局布线来实现指定的算法功能,因而可灵活地针对不同的网络结构设计定制化加速计算单元,实现高能效计算。

2.1 面向硬件资源约束的模型压缩方法

模型压缩是指利用神经网络参数的冗余性和网络结构的冗余性对模型进行精简,最终得到一个轻量且准确率相当的网络[42]。正如 1.2 节所述,语义分割网络绝大部分参数量和计算量都来自卷积层和反卷积层,而根据式(4)和(5)可以看出卷积操作中的参数量和计算量主要由输入特征图通道数、输出特征图通道数和卷积核尺寸等参数共同决定,卷积运算本质上就是输入特征图和卷积核参数的乘累加运算。因此可以从减少语义分割网络中决定参数量和计算量的有关参数(输入通道、输出通道和卷积核尺寸)或者减少参与乘累加运算的特征图参数和权值参数的数据位宽等角度实现对模型的压缩。常用的模型压缩方法包括参数剪枝、参数量化、低秩分解、参数共享、紧凑网络和知识蒸馏六大类[42]。

这其中,参数剪枝和参数量化最为通用,不仅能对语义分割网络起到良好的压缩效果,而且还可以利用 FPGA 对压缩后的网络进行硬件加速设计。其余的模型压缩方法虽然也能够精简 CNN,但是对基于 FPGA 的语义分割推理任务有一定的局限性。一方面,语义分割网络在 CNN 的基础上对网络结构进行改进,这导致部分模型压缩方法无法起到很好的压缩效果,比如低秩分解中奇异值分解(singular value decomposition, SVD)方法[44]就主要是针对 CNN 的全连接层进行压缩,无法在语义分割网络中起到同样的压缩效果。另一方面,以紧凑网络和知识蒸馏为代表的方法通过设计新型的卷积结构[45-46]或者删除原网络中的部分结构[47-49]来起到模型压缩和加速的效果。这些方法可能会改变神经网络中层的数量、各层的通道数和输出特征图尺寸以及层与层之间的连接关系,但是一般不改变诸如卷积、池化、反卷积等基本单元的运算规则,因此不会影响语义分割网络在 FPGA 上的计算方式,不需要设计特定硬件计算结构。

1)参数量化

参数量化是指使用较低位宽的数据来代替典型的 32 bit 浮点网络参数[42]。量化之后的网络参数可以是统一的位宽(如 16 bit,8 bit,4 bit 甚至 1 bit),也可以灵活地组合不同的位宽。

根据量化的映射函数是否是线性可以将量化方法分为线性量化和非线性量化两大类。这其中,非线性量化将原始数据通过非线性的方式映射到低位宽数据,量化前后数据的转换是一个查找表。典型的非线性量化方法是 2015 年 Han 等[50]提出的聚类量化方法,可以在不影响模型精度的前提下将 CNN 的权重压缩到 4 位,大幅减少模型存储的内存需求。而线性量化采用线性函数进行映射,量化前后的数据一般存在一个简单的线性变换关系。Yu 等[40]将语义分割网络的权重和激活量化成 8 bit 定点数,将模型压缩到原来的 1/8 而在 CamVid 数据集上的全局精度仅下降 2.04%。Huang 等[33]对语义分割网络中的特征图参数和权重进行 N 位定点线性量化,对卷积运算中的乘累加结果进行动态 M 位定点量化。同样都是被量化到 N 位,但是在不同通道的特征图参数的小数位不同。实验证明,在没有动态量化策略下,当数据位宽小于 12 bit 时就会对模型的精度有明显影响,而在使用动态量化之后,在数据位宽小于 8 bit 之前的精度损失都得到了明显缓解。Zhao 等[41]将语义分割网络编码器和解码器内的参数全部量化为 0 或 1,但在第一个卷积层仍使用高精度的参数,最终实现高达 307 fps/s 的吞吐量和 351.7 GOPs/W 的能效比。

参数量化不仅可以减少语义分割网络的存储需求和带宽需求,还可以通过降低每次运算需要的硬件成本来降低硬件的功耗,提高模型推理速度。其中,线性量化方法可以直接利用量化后的低位宽参数进行定点计算,在 FPGA 上执行所需要的面积、资源、能耗和延时都小于浮点计算,因此线性量化更适合于 FPGA 的硬件结构。但在相同比特下,线性量化的量化误差要大于非线性量化。根据语义分割网络的实际应用场景来选择合适的量化位宽,进而在网络的压缩比和精度损失之间取得平衡,是利用 FPGA 这类定制化加速器部署语义分割网络的重点关注问题。

2)参数剪枝

参数剪枝是指针对已经训练好的网络模型,设计网络参数的评价标准,并以此为依据删除冗余的参数。根据剪枝粒度的不同,参数剪枝方法可以分为结构化剪枝和非结构化剪枝[42]。结构化剪枝是粗粒度剪枝,剪枝的对象往往是一整个卷积核或某几个通道。结构化剪枝一般只会改变卷积层的通道数,不改变卷积层的计算方式,不会影响网络在 FPGA 上的部署策略,

与之相对应的,非结构化剪枝是细粒度剪枝,剪枝的粒度是神经网络中单个的权重,可以按照预设比例删除网络中冗余参数。非结构化剪枝得到的网络模型具有稀疏性,需要专门的硬件设备来加速稀疏矩阵的运算,而 FPGA 可以通过设计合适的硬件结构来实现稀疏矩阵的计算加速。Han 等[51]首先采用文献[50]中的非结构化剪枝方法删除了模型中 90% 以上的参数,然后使用 CSC(compressed sparse column)方法存储剪枝后的稀疏权值矩阵,最后设计了包括稀疏矩阵读取单元(sparse matrix read, SpmatRead)和稀疏矩阵乘法单元(sparse matrix-vector multiplication, SpMV)在内的硬件架构加速稀疏矩阵的运算。Shimoda 等[38]采用基于卷积核的剪枝方法,按照从小到大的顺序删除全卷积神经网络中 94% 的权重并采用 COO(coordinate)格式分别存储稀疏权重矩阵的行、列、通道和非零权重,然后通过 COO 解码器和 COO 计数器来实现权重的解码和非零计数,只有非零权重才会被传输到乘法器进行乘累加运算,最后在 Xilinx ZCU102 FPGA 开发板实现的速度是嵌入式 GPU(NVIDIA Jetson TX2 GPU)的 10.14 倍,能效比是后者的 24.49 倍。

参数剪枝通过删除冗余参数来减少语义分割网络的参数量和计算量,进而起到降低模型的存储需求和加快模型推理速度的效果。相比于更通用的结构化剪枝,非结构化剪枝可以删除更高比例的参数,但需要特定的硬件平台支持,设计开发的复杂度更高。基于 FPGA 设计定制化的硬件架构实现稀疏矩阵乘法的加速计算,是非结构化剪枝形成的稀疏语义分割网络高效部署和应用的关键,能够充分发挥 FPGA 灵活可重构的优势,取得更高的压缩比和硬件执行效率。

2.2 基于 FPGA 的加速计算架构设计

为了实现语义分割网络在 FPGA 上的部署和应用,在对既有模型进行压缩之后,研究人员一般会根据已有算法设计硬件架构。图 2 所示为一个典型的基于 FPGA 的 CNN 加速器设计[52]。

如图 2 所示,基于 FPGA 的 CNN 加速器由计算单元(processing element, PE)、片上存储器、片外存储器和片内/片外互联构成。在加速器启动工作之后,数据会从片外存储器传输到片上存储器最终传输到 PE 中进行处理。PE 是卷积运算的基本单元,多个 PE 并行可以加快卷积运算。片上互联负责将数据从片上存储器中传输到 PE。此外,加速器还针对网络的输入和输出部分设计两个不同的缓冲区,利用双缓冲区使得数据传输时间和计算时间重叠起来,有效地减少整个网络的执行时间。在以上设计中,整个系统的执行时间主要来自 PE 的计算用时和数据传输过程中的等待时间,因此基于 FPGA 的语义分割网络加速计算设计可以从计算结构优化和内存访问优化两方面展开[7]。同时,针对语义分割网络和 CNN 的不同点,比如反卷积层和跳跃连接,本文也从计算结构和内存访问这两方面进行分析和总结。

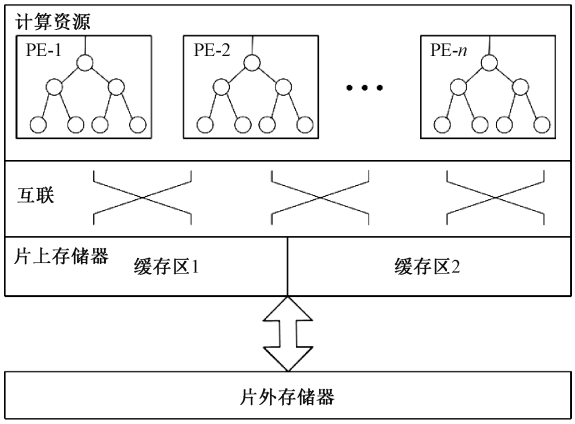


图 2 基于 FPGA 的 CNN 加速器[52]

Fig. 2 FPGA-based CNN accelerator[52]

1)计算结构优化

正如 1.2 节所述,语义分割网络绝大部分计算量都来自卷积层和反卷积层的卷积计算操作,基于 FPGA 的计算优化其实就是对卷积操作的优化。本文将从低位宽计算单元设计、并行计算设计和反卷积模块设计三个方面介绍语义分割网络在 FPGA 上的计算优化。

(1)低位宽计算单元优化

通过 2.1 节中 1)内介绍的线性量化方法可以将模型中 32 bit 浮点运算转换成低位宽的定点运算,而低位宽的定点运算需要相应位宽的定点计算单元来支持运算。在相同的运算中,使用低位宽计算单元可以节省大量计算资源,目前基于 FPGA 的语义分割网络大部分都使用定点计算单元来代替 32 bit 浮点计算单元,如在文献[23,31,39]中使用了 16 bit 定点计算单元,文献[37]和[40]中使用了 8 bit 定点计算单元。

目前 Xilinx 和 Intel 的新型 FPGA 上都设计了专门用于计算的 DSP 单元。文献[9]中的实验证明了 16 bit 及 16 bit 以下的定点数执行一次乘法或加法运算都需要至少一个 DSP 单元,因此将数据量化到 8 bit 或者更低位可能无法充分利用 DSP 单元,无法达到节约计算资源和面积的目的。针对以上问题,Nguyen 等[53]提出了一种在一个 25 bit×18 bit 的 DSP 乘法器上实现两个 8 bit 乘累加运算(multiply-and-accumulate, MAC)的方法,在资源利用率相同的情况下可以使卷积层的计算吞吐量翻倍。然而 Sabogal 等[36]针对同样的 25 bit×18 bit 的 DSP 乘法器,选择将特征图参数量化为 25 bit 定点数,将权重量化为 18 bit 定点数,实现对 DSP 乘法器单次运算所允许的数据范围的最大化利用,弥补参数量化造成的精度损失。

在 FPGA 上的低位宽计算单元设计可以有效减少单次乘法或者加法运算所需的计算资源,进而节省整个语义分割网络占用的计算资源,提高模型的计算吞吐量。针对较低位宽参数的运算可能无法充分利用 DSP 单元这一问题,研究人员可以根据实际需求设计量化策略,可以在一个 DSP 单元上执行多个低位宽乘法来提高运算吞吐量,也可以量化到 DSP 单元所允许的最大位宽来保证模型的精度。

(2)并行计算优化

由 1.2 节可知,语义分割网络的卷积运算本质上就是输入特征图和卷积核参数的乘累加操作,具体实现是以输入特征图行和列、卷积核的行和列、输入特征图通道数和输出特征图通道数为边界的六层嵌套循环。展开循环,在同一时间并行执行多个循环就可以提高硬件中计算资源的利用率,减少卷积层的计算用时。模型的循环展开参数就是模型的计算并行度,循环展开参数越大,模型的并行度就越高。但不同的语义分割网络甚至同一网络不同层的循环边界变化范围非常大。以语义分割网络常用的骨干网络 ResNet[22] 为例,它的特征图通道数从 3~2 048 不等,特征图的尺寸从 7×7 到 224×224 不等,卷积核尺寸从 1×1 到 7×7 不等,这导致难以确定一种通用且高效的循环展开策略,为此研究人员开展了大量工作。

Lyu 等[34]使用了 25 个乘法器将一个内核大小为 5×5 的卷积运算完全展开,大幅减少了卷积运算的耗时。Sabogal 等[36]将语义分割网络内的所有卷积层沿着输出特征图的通道方向展开 N 次,展开参数 N 和卷积运算的并行度具有二次关系,即 N 每扩大一倍,对应的卷积运算数量会扩大 3 倍。在 $N=2$、4 和 8 的情况下,在 Xilinx ZC706 中 DSP 计算单元的资源利用率分别为 4.44%、

16.89%和 65.78%,说明随着展开参数的增加,可以有效提高硬件资源的利用率。Liu 等[23]先将语义分割网络中内核尺寸较小的卷积和反卷积运算完全展开,然后设计自动化硬件生成工具,从 FPGA 片上的内存资源和逻辑资源两个方面建模分析,自动计算出语义分割网络在输入特征通道和输出特征通道上的最佳展开参数。

以上方法都是为语义分割网络内的所有层设计相同的并行度,然而考虑到语义分割网络中卷积层尺寸的差异,同样的循环展开参数难以在每一层上都起到良好的效果。文献[37]和文献[38]提出了一种全流水架构,它们为语义分割网络的每一层都设计了相应的硬件电路,因此能为每一层寻找最优的循环展开参数。但是这种方法需要将权重和相邻层之间的特征图参数全部储存在片上,只能适用于少数轻量级语义分割网络模型。

(3)反卷积计算单元优化

由 1.2 节可知,反卷积层也是语义分割网络的计算瓶颈之一,但在之前的工作中,研究人员一般只关注卷积层的优化加速。但相比于常规的卷积运算,反卷积运算还需要在每一个输入特征图像素的周围适当补零,因此将反卷积层直接部署在 FPGA 上的执行效率低下,在实际应用中推理用时反而超过了卷积层[23]。鉴于此,基于 FPGA 设计特定反卷积计算单元也越来越得到了相关研究人员的重视。

Fang 等[32]提出了一种名为亚里士多德(Aristotle)的深度学习部署架构,在片外存储器和片上存储器之间采用数据重排操作来高效地实现语义分割网络中的反卷积和空洞卷积。Liu 等[23]通过优化反卷积算法的实现方法,减少反卷积过程中的运算量,使其在 FPGA 上的实现效率更高。实验结果表明,该方法可以将 U-Net 中反卷积层的计算量降低 4 倍,将 FCN 中反卷积层的计算量降低 7.2 到 83.4 倍。

基于 FPGA 的反卷积计算模块设计能够提高语义分割网络中反卷积层的计算效率,加快模型的推理速度。然而,为了避免反卷积造成的计算和内存访问,部分语义分割网络直接采用了双线性插值[38]或者上池化[39]进行上采样,因此反卷积模块设计并不能适用于所有的语义分割网络。

2)内存访问优化

提高神经网络并行度的同时,片上存储系统需要在每个时钟周期将必要的数据提供给每个计算模块,使得加速器无需等待数据。仅进行循环展开而不考虑内存访问时间往往会影响并行计算的效率,造成大量的额外延时,无法实现语义分割网络的高效推理计算。FPGA 的内存访问优化主要从片上数据存储和传输优化与层融合设计两方面入手。

(1)片上数据存储和传输优化

为了提高内存访问效率,神经网络一般会在计算单元之间重用数据,但将数据直接传输到不同的计算单元可能会造成较大的扇出和布线成本,进而影响硬件的工作频率。Wei 等[54]提出脉动阵列的架构,将数据以流水线的形式从一个计算单元传递到下一个运算单元,降低数据访存次数,使 FPGA 的结构更加整齐,布线更加一致,从而提高芯片的工作频率。Lyu 等[34]为提高卷积计算过程中内存的访问效率,设计了行缓冲区。每个行缓冲区由 4 列,每列 5 个的移位寄存器组成,数据会从一个方向流入行缓冲区然后按照指定顺序流出,从而实现数据在不同计算单元之间的重用。

在对模型的需求和 FPGA 的资源进行综合分析之后,Zhao 等[41]选择使用指令手动调整各项资源的利用率,将权重参数存储在分布式存储器(distributed random access memory)中,将语义分割网络的中间特征图参数存储在块存储器(block random access memory, BRAM)中,把 FPGA 上的每一个计算单元都和相应的片上存储相关联,实现了高吞吐量和高带宽。Liu 等[23]为卷积模块和反卷积模块设计共享的输入缓冲区,使得整个硬件加速器的输入缓冲区的大小减半,缓解了模型的存储压力。

在本节中提到了采用全流水架构的网络[37-38],由于其内存读取和写出操作几乎都在片上存储器进行且整个模型的吞吐量会由吞吐量最小的层所决定,它们更需要对数据的存储和传输做优化来保证提高每一层的吞吐量和速度。文献[38]提出一种基于 FPGA 的稀疏全卷积神经网络(sparse fully convolutional network, SFCN),并且在硬件实现中采用全流水的架构,本文将以此为例介绍全流水架构中的片上数据存储和传输策略。SFCN 的总体架构如图 3 所示。

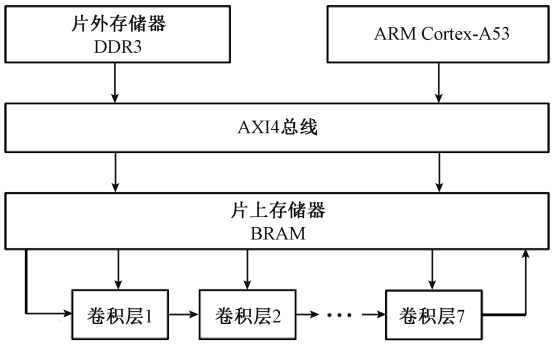


图 3　SFCN 的总体架构[38]

Fig. 3　The architecture of SFCN[38]

在启动之后,所有数据,包括图像数据和权重参数会从 DDR3 传输到片上存储器 BRAM 中,之后图像数据会被输入到神经网络的第一个卷积层内进行运算,然后依次经过所有层,最后将结果传回 ARM 处理器。在这个过

程中,所有层的权重都是直接从片上 BRAM 中读取,需要根据每层的计算并行度对权重所在的 BRAM 进行分割,提高数据读取速率。由于中间特征图数据都存储在片上,前一个卷积层的输出特征图就是后一个卷积层的输入,因此可以实现相邻卷积层之间特征图数据的复用。

(2)层融合设计

神经网络中相邻层的融合可以避免层与层的中间数据的传输过程,减少片外存储访问次数,提高模型的内存访问效率,从而起到加快网络运行速度的作用。Alwani 等[55]通过修改 CNN 输入数据的顺序,将相邻的数个卷积层融合在一起,以增加 20% 的片上存储成本为代价减少 95% 的片外存储访问。在此基础上,Xiao 等[56]提出了一种基于行缓冲区的融合架构,不需要处理复杂的边界条件即可实现卷积层的融合。在与文献[55]方法的对比实验中,该方法取得了平均 1.99 倍的速度提升。Zhao 等[41]将二进制卷积层和池化层融合在一起,可以在一个时钟周期内同时进行二进制卷积和最大值池化运算,从而提高整个语义分割网络的吞吐量。

然而,语义分割网络一般需要通过跳跃连接融合图像低层和高层的语义信息,所以部分卷积层的输出结果会在上采样过程中重复使用。如果想利用层融合设计优化这些卷积层,避免将它们的中间结果存储到片外,就必须将这些结果一直存储在片上,这会付出巨大的存储代价,甚至影响整个加速器的架构。因此对语义分割网络进行层融合设计时要充分考虑网络中存在的跳跃连接关系,避免影响网络的正常功能。

3 发展前景展望

随着深度学习技术的发展,语义分割在自动驾驶、无人机、安防监控和智能家居等领域展现了良好的应用前景,基于 FPGA 的语义分割网络优化加速技术也越来越受到研究人员的重视。然而,随着人工智能生态的爆发,现在的技术还无法满足各种各样复杂场景的应用需求,需要研究人员继续探索,具体体现在以下几个方面。

1)异构计算。基于 CPU+FPGA 的异构计算平台是目前用于加速语义分割网络的主流 FPGA 平台。异构计算通过组合不同硬件平台的优势来实现对语义分割网络的优化加速。随着语义分割网络应用场景的日趋复杂,融合多种硬件的异构计算平台可以提供更加通用且高效的解决方案,具有很高的研究价值。

2)面向边缘计算的高精度语义分割。在边缘设备中部署和应用的语义分割网络一般都会为了速度或者节省资源而牺牲一部分精度。但在自动驾驶等领域,图像语义分割的结果会关乎乘客的安全,对精度要求非常高。探索如何进一步提高在边缘端的图像语义分割的精度,是该领域的一个研究热点。

3)新型存储技术的发展和应用。边缘设备中的内存有限,而语义分割网络需要大量的数据交互,一般需要对内存进行反复读写。频繁的内存访问不仅增加了推理用时,还造成了能源的浪费。随着存储技术的发展,诸如高带宽存储器(high bandwidth memory, HBM)[57]和混合内存立方体(hybrid memory cube, HMC)[58]等新型存储技术可以提供更高的存储密度、更高的带宽和更好的能效比,但由于他们的价格昂贵,目前主要用于服务器和台式机。推动新型存储技术在边缘设备中的应用,帮助语义分割网络在边缘设备上的高效部署,具有很高的研究价值。

4)自动化压缩和加速。当前语义分割网络压缩和加速技术中的很多参数,比如线性量化的位宽和循环的展开参数,都是根据设计者的知识经验和实际实验来确定的,这给研究人员带来了巨大工作量。因此,开发全自动的模型压缩和加速方法和工具,降低语义分割网络的使用成本是一个非常有价值的方向。文献[23]已经进行了一些自动化设计的探索工作。

5)优化技术的组合。现有的研究工作从算法、软件和硬件的不同方面解决语义分割网络在 FPGA 上应用过程中的问题。如果可以将已有的各种技术结合在一起,就可以进一步提高语义分割网络的性能,降低使用成本。技术的结合需要软件和硬件的高度协同,尽管 DNNDK 等商业工具[32]已经在这个方向迈出了第一步,但还需要更多的研究和技术支持。

4 结 论

随着深度学习技术的应用和发展,基于深度学习的语义分割已成为自动驾驶、安防监控、无人机和智能家居等多个领域关注的热点技术。这其中,基于边缘计算的语义分割网络在实时性、可靠性和隐私保护展现出巨大的优势,能够为每一个用户提供低成本、高质量和高可靠性的智能服务,但其高存储和高计算需求和边缘设备中有限的资源和功耗之间的矛盾亟待解决。FPGA 凭借其高性能、高能效、低功耗和可重构等优势成为在边缘设备中部署和应用的语义分割的重要加速计算平台。

本文在语义分割网络基本原理和计算结构介绍的基础上,结合 FPGA 的硬件特点,总结基于 FPGA 的语义分割网络加速优化方法,将其分为面向硬件资源约束的模型压缩方法和基于 FPGA 的加速计算架构设计方法两大类。接着从参数剪枝和参数量化两个角度介绍面向硬件资源约束的模型压缩方法,重点分析以上两种方法在 FPGA 上造成的计算结构的改变和影响;从计算结构优化和内存访问优化两个角度介绍基于 FPGA 的加速计算结构设计方法,针对每种方法的代表性成果进行研究分

析,总结其优缺点。最后,梳理目前研究存在问题,并从异构计算、面向边缘的高精度语义分割、新型存储技术的发展和应用、自动化压缩和加速以及优化技术的组合这五个方面展望未来的发展方向,以期为语义分割、边缘计算和定制高能效计算等相关领域的研究人员提供有益的启发和借鉴。

参考文献

[1] CSURKA G, PERRONNIN F. An efficient approach to semantic segmentation [J]. International Journal of Computer Vision, 2011, 95(2): 198-212.

[2] 王嫣然, 陈清亮, 吴俊君. 面向复杂环境的图像语义分割方法综述[J]. 计算机科学, 2019, 46(9): 36-46.
WANG Y R, CHEN Q L, WU J J. Research on image semantic segmentation for complex environments [J]. Computer Science, 2019, 46(9): 36-46.

[3] 姜枫, 顾庆, 郝慧珍, 等. 基于内容的图像分割方法综述[J]. 软件学报, 2017, 28(1): 160-183.
JIANG F, GU Q, HAO H ZH, et al. Survey on content-based image segmentation methods [J]. Journal of Software, 2017, 28(1): 160-183.

[4] 田萱, 王亮, 丁琪. 基于深度学习的图像语义分割方法综述[J]. 软件学报, 2019, 30(2): 440-468.
TIAN X, WANG L, DING Q. Review of image semantic segmentation based on deep learning [J]. Journal of Software, 2019, 30(2): 440-468.

[5] WANG X, HAN Y, LEUNG V C, et al. Convergence of edge computing and deep learning: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 869-904.

[6] CHEN Y, ZHENG B, ZHANG Z, et al. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions[J]. ACM Computing Surveys (CSUR), 2020, 53(4): 1-37.

[7] 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, 41(11): 2461-2480.
WU Y X, LIANG K, LIU Y, et al. The progress and trends of FPGA-based accelerators in deep learning [J]. Chinese Journal of Computers, 2019, 41(11): 2461-2480.

[8] QASAIMEH M, DENOLF K, LO J, et al. Comparing energy efficiency of cpu, gpu and fpga implementations for vision kernels [C]. 2019 IEEE International Conference on Embedded Software and Systems (ICESS), 2019: 1-8.

[9] GUO K, ZENG S, YU J, et al. [DL] a survey of fpga-based neural network inference accelerators [J]. ACM Trans. Reconfigurable Technol. Syst., 2019, 12(1): 1-26.

[10] 汪志文. 基于深度学习的高分辨率遥感影像语义分割的研究与应用[D]. 北京: 北京邮电大学, 2019.
WANG ZH W. Research and application of semantic segmentation of high-resolution remote sensing image based on deep learning[D]. Beijing: Beijing University of Posts and Telecommunications, 2019.

[11] 景庄伟, 管海燕, 彭代峰, 等. 基于深度神经网络的图像语义分割研究综述[J]. 计算机工程, 2020, 46(10): 1-17.
JING ZH W, GUAN H Y, PENG D F, et al. Survey of research in image semantic segmentation based on deep neural network [J]. Computer Engineering, 2020, 46(10): 1-17.

[12] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3431-3440.

[13] 邝辉宇, 吴俊君. 基于深度学习的图像语义分割技术研究综述[J]. 计算机工程与应用, 2019, 55(19): 12-21+42.
KUNG H Y, WU J J. Survey of image semantic segmentation based on deep learning [J]. Computer Engineering and Applications, 2019, 55(19): 12-21+42.

[14] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]. International Conference on Medical Image Computing and Computer-assisted Intervention, 2015: 234-241.

[15] CHEN L C, PAPANDREOU G, KOKKINOS I, et al. Semantic image segmentation with deep convolutional nets and fully connected crfs[C]. Computer Science, 2014: 357-361.

[16] CHEN L C, PAPANDREOU G, KOKKINOS I, et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(4): 834-848.

[17] CHEN L C, ZHU Y, PAPANDREOU G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation [C]. Proceedings of the European Conference on Computer Vision (ECCV), 2018: 801-818.

[18] BADRINARAYANAN V, KENDALL A, CIPOLLA R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(12): 2481-2495.

[19] ZHAO H, SHI J, QI X, et al. Pyramid scene parsing network [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2881-2890.

[20] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [C]. International Conference on Learning Representations, 2015, 1-14.

[21] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

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

[23] LIU S, FAN H, NIU X, et al. Optimizing cnn-based segmentation with deeply customized convolutional and deconvolutional architectures on fpga [J]. ACM Transactions on Reconfigurable Technology and Systems (TRETS), 2018, 11(3): 1-22.

[24] 蓝金辉, 王迪, 申小盼. 卷积神经网络在视觉图像检测的研究进展[J]. 仪器仪表学报, 2020, 41(4): 167-182.
LAN J H, WANG D, SHEN X P. Research progress on visual image detection based on convolutional neural network [J]. Chinese Journal of Scientific Instrument, 2020,41(4): 167-182.

[25] 郭玥, 于希明, 王少军, 等. 遥感图像云检测的多尺度融合分割网络方法[J]. 仪器仪表学报, 2019, 40(6): 31-38.
GUO Y, YU X M, WANG SH J, et al. Cloud detection in remote sensing images with multilevel scale fused network [J]. Chinese Journal of Scientific Instrument, 2019, 40(6): 31-38.

[26] 罗军舟, 金嘉晖, 宋爱波, 等. 云计算:体系架构与关键技术[J]. 通信学报, 2011, 32(7): 3-21.
LUO J ZH, JIN J H, SONG AI B, et al. Cloud computing: Architecture and key technologies [J]. Journal on Communications, 2011, 32(7): 3-21.

[27] LIU F, TONG J, MAO J, et al. Nist cloud computing reference architecture [J]. NIST Special Publication, 2011, 500(2011): 1-28.

[28] SHI W, CAO J, ZHANG Q, et al. Edge computing: Vision and challenges [J]. IEEE Internet of Things Journal, 2016, 3(5): 637-646.

[29] CHEN J, RAN X. Deep learning with edge computing: A review[J]. Proceedings of the IEEE, 2019, 107(8): 1655-1674.

[30] 胡雷钧, 陈乃刚, 李健, 等. Fpga 异构计算平台及其应用[J]. 电力信息与通信技术, 2016, 14(7): 6-11.
HU L J, CHEN N G, LI J, et al. FPGA heterogeneous computing platform and its applications [J]. Electric Power Information and Communication Technology, 2016, 14(7): 6-11.

[31] CHANG A X M, ZAIDY A, CULURCIELLO E. Efficient compiler code generation for deep learning snowflake co-processor [C]. 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), 2018: 24-28.

[32] FANG S, TIAN L, WANG J, et al. Real-time object detection and semantic segmentation hardware system with deep learning networks [C]. 2018 International Conference on Field-Programmable Technology (FPT), 2018: 389-392.

[33] HUANG H, WU Y, YU M, et al. Edssa: An encoder-decoder semantic segmentation networks accelerator on opencl-based fpga platform [J]. Sensors, 2020, 20(14): 3969.

[34] LYU Y, BAI L, HUANG X. Real-time road segmentation using lidar data processing on an fpga[C]. 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018: 1-5.

[35] LYU Y, BAI L, HUANG X. Chipnet: Real-time lidar processing for drivable region segmentation on an fpga[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2018, 66(5): 1769-1779.

[36] SABOGAL S, GEORGE A, CRUM G. Recon: A reconfigurable cnn acceleration framework for hybrid semantic segmentation on hybrid socs for space applications [C]. 2019 IEEE Space Computing Conference (SCC), 2019: 41-52.

[37] SADA Y, SHIMODA M, JINGUJI A, et al. A dataflow pipelining architecture for tile segmentation with a sparse mobilenet on an fpga[C]. 2019 International Conference on Field-Programmable Technology (ICFPT), 2019: 267-270.

[38] SHIMODA M, SADA Y, NAKAHARA H. Filter-wise pruning approach to fpga implementation of fully convolutional network for semantic segmentation [C]. International Symposium on Applied Reconfigurable Computing, 2019: 371-386.

[39] TANN H, ZHAO H, REDA S. A resource-efficient embedded iris recognition system using fully convolutional networks[J]. ACM Journal on Emerging Technologies in Computing Systems (JETC), 2019, 16(1): 1-23.

[40] YU M, HUANG H, LIU H, et al. Optimizing fpga-based convolutional encoder-decoder architecture for semantic segmentation[C]. 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), 2019: 1436-1440.

[41] ZHAO S, AN F, YU H. A 307-fps 351.7-gops/w deep learning fpga accelerator for real-time scene text recognition[C]. 2019 International Conference on Field-Programmable Technology (ICFPT), 2019: 263-266.

[42] 高晗, 田育龙, 许封元, 等. 深度学习模型压缩与加速综述[J]. 软件学报, 2021, 32(1): 68-92.
GAO H, TIAN Y L, XU F Y, et al. Survey of deep learning model compression and acceleration[J]. Journal of Software, 2021, 32(1): 68-92.

[43] 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+119.
YU Z J, MA D, YAN X L, et al. FPGA-based accelerator for convolutional neural network [J]. Computer Engineering, 2017, 43(1): 109-114+119.

[44] DENTON E, ZAREMBA W, BRUNA J, et al. Exploiting linear structure within convolutional networks for efficient evaluation[C]. 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014, 2014: 1269-1277.

[45] IANDOLA F N, HAN S, MOSKEWICZ M W, et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size[C]. International Conference on Learning Representations, 2017: 1-13.

[46] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 6848-6856.

[47] BA L J, CARUANA R. Do deep nets really need to be deep? [C]. Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, 2014: 2654-2662.

[48] LI Z, HOIEM D. Learning without forgetting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(12): 2935-2947.

[49] CROWLEY E J, GRAY G, STORKEY A. Moonshine: Distilling with cheap convolutions[C]. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018: 2893-2903.

[50] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding [J]. International Conference on Learning Representations, 2016: 1-14.

[51] HAN S, KANG J, MAO H, et al. Ese: Efficient speech recognition engine with sparse lstm on fpga [C]. Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2017: 75-84.

[52] ZHANG C, LI P, SUN G, et al. Optimizing fpga-based accelerator design for deep convolutional neural networks[C]. Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2015: 161-170.

[53] NGUYEN D, KIM D, LEE J. Double mac: Doubling the performance of convolutional neural networks on modern fpgas [C]. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, 2017: 890-893.

[54] WEI X, YU C H, ZHANG P, et al. Automated systolic array architecture synthesis for high throughput cnn inference on fpgas[C]. Proceedings of the 54th Annual Design Automation Conference 2017, 2017: 1-6.

[55] ALWANI M, CHEN H, FERDMAN M, et al. Fused-layer cnn accelerators[C]. 2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2016: 1-12.

[56] XIAO Q, LIANG Y, LU L, et al. Exploring heterogeneous algorithms for accelerating deep convolutional neural networks on fpgas[C]. Proceedings of the 54th Annual Design Automation Conference 2017, 2017: 1-6.

[57] JUN H, CHO J, LEE K, et al. Hbm (high bandwidth memory) dram technology and architecture[C]. 2017 IEEE International Memory Workshop (IMW), 2017: 1-4.

[58] JEDDELOH J, KEETH B. Hybrid memory cube new dram architecture increases density and performance[C]. 2012 Symposium on VLSI Technology (VLSIT), 2012: 87-88.

作者简介

彭宇,2004 年于哈尔滨工业大学获得博士学位,现为哈尔滨工业大学教授、博士生导师,主要研究方向为虚拟仪器和自动测试、故障预测与健康管理、可重构计算等。

E-mail: pengyu@hit.edu.cn

Peng Yu received his Ph. D. degree from Harbin Institute of Technology in 2004. He is currently a professor and a Ph. D. advisor at Harbin Institute of Technology. His main research fields include virtual instrument and automatic test technology, prognostics and system health management, and reconfigurable computing, etc.

姬森展,2019 年于哈尔滨工业大学获得学士学位,现为哈尔滨工业大学测控工程系在读硕士研究生,主要研究方向为图像语义分割和基于 FPGA 的硬件加速技术。

E-mail: 19s001050@stu.hit.edu.cn

Ji Senzhan received his B. Sc. degree in 2019 from Harbin Institute of Technology (HIT). He is a master student in the Department of Test and Control Engineering at HIT. His research interests include image semantic segmentation and FPGA-based hardware acceleration technology.

于希明(通信作者),于 2016 年于哈尔滨工业大学获得学士学位,现为哈尔滨工业大学博士生,主要研究方向为遥感图像处理与深度学习模型计算加速等。

E-mail: yuximing@hit.edu.cn

Yu Ximing (Corresponding author) received his B. Sc. degree from Harbin Institute of Technology in 2016. He is currently a Ph. D. candidate at Harbin Institute of Technology. His main research field includes remote sensing image processing and computing acceleration for deep learning model, etc.