基于FPGA的Skynet网络结构优化及高时效实现

唐维伟[1,2],钟 胜[1,2],卢金仪[1,2],颜露新[1,2],谭富中[1,2],邹 旭[1,2],徐文辉[1,2]
(1. 华中科技大学人工智能与自动化学院,湖北武汉 430074;
2. 华中科技大学多谱信息处理技术国家级重点实验室,湖北武汉 430074)

摘 要: 基于卷积神经网络(Convolutional Neural Network,CNN)的目标检测算法有着鲁棒性强、准确度高等优点,被广泛用于计算机视觉任务领域. 然而,CNN参数量大、计算量大的特性使得其难以在边缘计算平台实时实现,为此,本文针对目标检测网络Skynet进行结构优化,并基于高效的层内并行流水的加速架构,在现场可编程门阵列(Field Programmable Gate Array,FPGA)上对其进行实时实现. 该方法对Skynet进行剪枝,合并其卷积层与归一化层,利用(Kullback-Leibler,KL)相对熵及极大值量化方法对权重及特征图进行8 bit定点量化,同时将偏置参数及缩放系数定点化,并合并激活操作与饱和截断操作,在减少存储量和计算量的同时,加快前向推理速度. 此外,以滑窗操作为基础,采用通道及像素并行计算,设计深度可分离卷积的流水策略,将串行的前向推理结构优化为并行流水的结构,极大减少了前向推理的时间. 实验表明,在UA-DETRAC数据集上,本文实现的系统识别精度为0.752,在160×160的图像分辨率上,速度达到115FPS,与CPU相比,提速11倍,达到了GPU的75%,功耗分别为CPU的10.6%,GPU的7.43%,而且,与同类基于FPGA的CNN加速工作相比,本文方法在速度和能效比上均表现最优.

关键词: 目标检测网络;定点量化;现场可编程门阵列;流水计算;skynet
基金项目: 国家自然科学基金(No.61806081);国防基础科研计划资助(No.JCKY2018204B068)
中图分类号: TN47 **文献标识码:** A **文章编号:** 0372-2112(2023)02-0314-10
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210028

Network Structure Optimization and High-Efficiency Implementation of Skynet Based on FPGA

TANG Wei-wei[1,2], ZHONG Sheng[1,2], LU Jin-yi[1,2], YAN Lu-xin[1,2], TAN Fu-zhong[1,2], ZHOU Xu[1,2], XU Wen-hui[1,2]
(1. *School of Artificial Intelligence and Automation*, *Huazhong University of Science and Technology*, *Wuhan*, *Hubei* 430074, *China*;
2. *National Key Laboratory of Science & Technology on Multi-Spectral Information Processing*, *Huazhong University of Science and Technology*, *Wuhan*, *Hubei* 430074, *China*)

Abstract: The object detection algorithm based on convolutional neural network (CNN) has the advantages of strong robustness and high accuracy, and is widely used in the field of computer vision tasks. However, the size of CNN parameters and the amount of calculation make it difficult to implement in real-time on edge computing platforms. For this reason, this paper optimizes the structure of the object detection network Skynet, and realizes on the field programmable logic gate array (FPGA) based on an efficient intra-layer parallel pipeline acceleration architecture. This method prunes skynet, merges its convolutional layer and normalization layer, uses the (KL) relative entropy method and maximum quantization method to perform 8 bit fixed-point quantization on the weights and feature maps, and converts bias and scaling coefficients into fixed point, then merges the activation operation and saturation truncation operation for speeding up the CNN forward calculation. In addition, this paper optimizes serial structure to pipeline parallel structure based on the sliding window operation, parallelizes channel and pixel calculation, then designs a pipeline strategy for depthwise separable convolution, which greatly reduces time to forward calculation. Experiments show that on the UA-DETRAC dataset, the method recognition accuracy of this paper is 0.752, and the frame rate reaches 115FPS at an image resolution of 160×160, which is 11 times faster than the CPU and reaches 75% of the GPU. The power is reduced to 10.6% of the CPU and 7.43% of the GPU. Moreover, the proposed method has the best performance in both speed and energy efficiency ratio by comparing with the similar CNN acceleration methods based on FPGA.

收稿日期:2020-12-28;修回日期:2021-03-30;责任编辑:王天慧

Key words： object detection network; fixed-point quantization; field programmable gate array; pipeline calculation; Skynet

Foundation Item(s)： National Natural Science Foundation of China (No.61806081); Industrial Technology Development Program grant (No.JCKY2018204B068)

1 引言

目标检测任务是指找出图像中感兴趣的目标并加以标识[1]. 随着深度神经网络的快速发展,学术界涌现出很多目标检测网络,如一阶段YOLO[2]或二阶段RCNN[3]等. 它们被广泛用于自动驾驶、人脸检测和视频监控等场景中,并取得了比传统方法更好的性能,但由于其庞大的计算量和参数量,在资源有限的边缘设备上部署受到限制.

为了解决该问题,Howard等人[4]提出Mobilenet的轻量型网络,Chollet等人[5]则提出Xception的轻量型网络,此类网络使用深度可分离卷积(Depthwise Separable Convolution,DSC)替代标准卷积,以降低计算量和参数量,一定程度上提高了目标检测网络在边缘设备上的运行效率. Zhang等人[6]在深度可分离卷积基础上,针对边缘设备设计出硬件友好型网络Skynet,其相较于MobileNet以及Xception,Skynet结构更规整,模块复用率更高,但仍对部署平台有较高算力要求. 为进一步减少参数量,Nakahara等人[7,8]及Nguyen等人[9]开展了二值化网络的研究,但精度受限,且在通用平台部署并不具有优势. 为满足边缘应用场景,诸多神经网络处理器(Neural network Processing Unit,NPU)被设计出,但此类专用芯片的设计开发周期过长,开发成本及承担风险较高. FPGA作为灵活性较高的可编程器件,利用硬件并行流水结构实现新网络,未来可提供子单元进行抽象,为未来NPU进一步发展提供参考. 因此,逐渐有研究者利用FPGA开展网络部署工作,主要从简化(Convolutional Neural Networks,CNN)计算复杂度、改善浮点计算对FPGA的资源消耗以及提高FPGA时钟频率三个方面展开.

为简化CNN在FPGA上的计算复杂度,可利用网络稀疏性[10]、循环合并[11]、权重重排[12]等策略. 其中,利用网络稀疏性[10]可避免0元素的计算,但与之相结合的矩阵乘法会消耗FPGA较多存储资源;循环合并[11]指合并具有相同循环次数且数据间无依赖的循环结构,权重重排[12]则是配合计算模块,相应地改变权重存储结构,提高读入参数效率,这两种操作可共同开展,以减少前向推理的复杂度.

为改善CNN中大量浮点数计算严重消耗FPGA片上乘法器资源及存储资源的问题,一般采用量化[13~15]、循环分块[13,16~18]等方法. 其中,对网络模型进行量化[13~15],损失部分精度以避免浮点数运算,同时降低片上资源消耗,但冗余度较低的网络量化后精度下降较严重;利用循环分块[13,16~18]的操作,减缓了存储的压力同时提升实时性,但这会带来额外读写外部DRAM存储器的时间消耗. 文献[19]则是采取复用卷积计算及池化计算模块以节省资源的方法,但设计的计算模块内部并行通道数过大,导致实际上板时布局布线困难,时序难以满足要求,时钟频率很难提高.

为提升FPGA时钟频率,Fan等人[14]、Bai[15]以及Zhao等人[20]利用滑窗与行缓存的方法,契合卷积计算中卷积核滑动的特点,从任务层面,将消耗多个周期才能完成的一次卷积计算任务分解为单周期计算的流水线任务,缩短了关键路径,充分提高了时钟频率.

为满足边缘场景目标检测网络应用需求,提高实时性,本文以轻量化、高性能且硬件友好型网络Skynet为基础,优化其网络结构,并基于FPGA设计高效的硬件加速结构. 通过对特征图和权重参数进行量化,在保证精度的同时减少存储及计算资源消耗. 同时,重新构造前向推理结构,合并DSC中计算操作,进一步减少计算量. 而且,以滑窗为基础,设计完全的硬件流水加速架构,有效提升了频率. 最终结果表明,在UA-DETRAC数据集上,FPGA上板推理速度相较于CPU提升了11倍,精度仅下降了2.34%.

2 Skynet网络结构优化

Skynet具有精度高、参数少的优点,但其网络结构包含旁路分支,结构较复杂,且该分支将连续的特征图存储到不连续的地址上,地址跳转过于频繁,影响流水效率,不利于设计高效的计算架构.

卷积神经网络高冗余度[21]的特点使得结构上的精简具有可操作性. 因此我们首先进行剪枝的相关实验,将Skynet分支剪枝掉,网络结构优化为直筒型. 表1为Skynet剪枝前后精度对比,以平均精度(Average Precision,AP)为评价指标,其精度下降不到0.03,在满足实际应用需求的情况下,剪枝该分支能避免烦琐的地址跳转操作,设计更高效的并行流水架构,极大提高Skynet在FPGA上实时计算的效率.

表1　剪枝前后精度对比

模型类型	平均精度
完整模型[6]	0.797
剪枝后模型(本文方法)	0.770

优化后的Skynet如图1所示,整体网络一共包含8层,分别是3通道输入层(CHL3)、中间层(CHL32~CHL96)、回归层(CHL30),其中,CHL后数字代表特征图通道数量,如CHL3表示该层特征图有3个通道.

每层之间卷积使用深度可分离卷积实现，网络结构规整，便于模块复用，形成高效的计算结构.

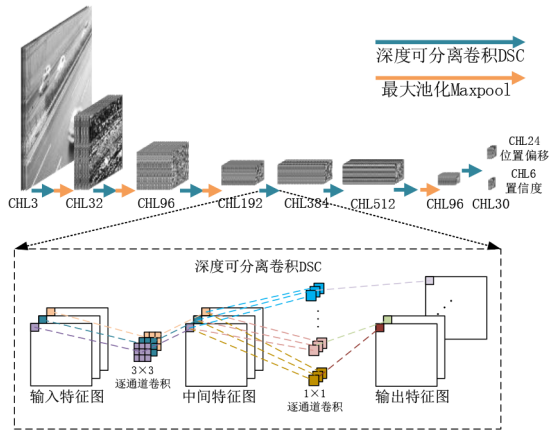

图1 优化后Skynet网络结构图

DSC如图1下半部分所示，主要包含逐通道(Depth-wise，DW)卷积和逐点(Pointwise，PW)卷积，除DSC外，前三层及第六层后接入最大池化模块，网络末端引入回归检测框，实现目标的边界框的位置回归.

仅考虑卷积层的参数量和计算量，设输入通道数为T_{cin}，输出通道数为T_{cout}，卷积核尺寸为K，输出特征图尺寸为W_{out}和H_{out}. 则进行单层DSC卷积参数量为

$$\text{Param\_num}=K\times K\times T_{cin}+T_{cin}\times T_{cout} \tag{1}$$

乘法次数为

$$\text{OP}_{Mul}=K\times K\times T_{cin}\times W_{out}\times H_{out}+T_{cin}\times T_{cout}\times W_{out}\times H_{out} \tag{2}$$

逐通道卷积中卷积核尺寸为3×3，设深度可分离卷积的参数量及计算量占标准卷积的S_0，其中：

$$S_0=\frac{1}{T_{cout}}+\frac{1}{K^2} \tag{3}$$

同样设$T_{cin}=32$，$T_{cout}=96$，$W_{out}=80$，$H_{out}=80$，则该层参数量为3 360，乘法次数21.504 M，二者均只占标准卷积的12.15%，能够大幅度降低对边缘设备的压力.

3 结构及参数重构

3.1 权重量化

模型导出的权重参数均为单精度浮点型，将其量化到低字节可节省大量FPGA乘法器资源及查找表资源.

量化的本质是用一个次分布代替原始的分布，图2为各层权重参数分布情况，除个别层外，大部分权重分布都十分均匀和对称，因此直接采用极值缩放量化方式，即非饱和式量化.

非饱和式量化需选取最大值，最大值的选取有两种：

(1)选取当前层所有卷积核中的最大值；

(2)选取各输出通道对应的卷积核中的最大值.

如图2所示，尽管分布十分均匀，但仍存在数量较

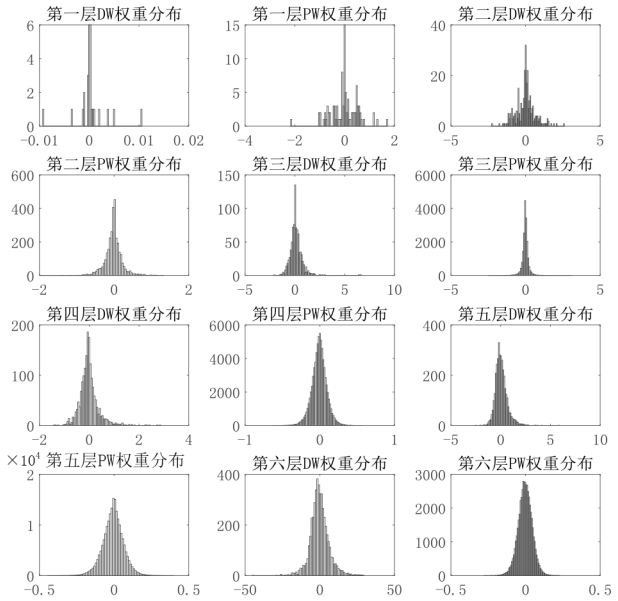

图2 各层权重分布图

少，值比较突出的权重系数：按(2)选取最大值进行缩放系数的计算能最小化量化带来的精度损失；按(1)选取，会导致多数通道的卷积核缩放后分布不均匀，大部分值集中在零点附近，表达信息的能力减弱. 因此按通道进行权重参数量化.

设各通道对应的原始权重为$\boldsymbol{w}$，量化后权重为$\boldsymbol{q_w}$，缩放系数为$\textbf{scale_w}$，fabs表示取标量值的绝对值，则对应关系如式(4)和式(5)所示，为便于运算，边界取127，最终得到的$\boldsymbol{q_w}$均匀分布在-127~127之间.

$$\text{scale\_}w=\frac{127}{\max[\text{fabs}(\boldsymbol{w})]} \tag{4}$$

$$\boldsymbol{q\_w}=\boldsymbol{w}\times\textbf{scale\_w} \tag{5}$$

3.2 特征图量化

对卷积后的结果进行非线性激活，会导致特征图数据分布不均匀，且其最大值会随输入图像变化. 与非饱和式量化方法相比，对特征图进行基于KL相对熵的饱和式量化，能显著减小精度损失.

饱和式量化如图3所示，选取一阈值T，将原始分布处在$-T$~T范围内的值等比例缩放到-127~127，图中红色的值表缩放后超出范围外，进行饱和处理，直接取饱和值表示该部分值.

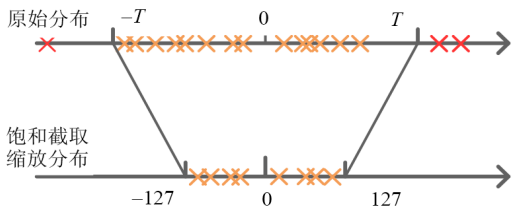

图3 饱和式截取缩放量化示意图

该量化方式依赖阈值T的选取,通过遍历的方式,得到不同阈值下的量化后分布,为确定使量化后精度损失最小的T值,需对量化前后数据分布使用KL相对熵定量分析.

KL相对熵可以衡量两个分布之间的差异情况,值越小表示两个分布越相似. 设量化前原始分布为p,使用阈值T量化后分布为q,原始分布信息熵为$H(p)$,原始分布与量化后分布交叉熵为$H(p,q)$,KL相对熵为$D_{\mathrm{KL}}(p\|q)$,则

$$H(p)=\sum_{i=1}^{N}p_i\times\log\frac{1}{p_i} \tag{6}$$

$$H(p,q)=\sum_{i=1}^{N}p_i\times\log\frac{1}{q_i} \tag{7}$$

$$D_{\mathrm{KL}}(p\|q)=\sum_{i=1}^{N}p_i\times\log\frac{p_i}{q_i} \tag{8}$$

根据KL相对熵可判断阈值选择是否最佳,同时可得缩放系数为scale_fm,其中:

$$\mathrm{scale\_fm}=\frac{127}{|T|} \tag{9}$$

推理过程中,将特征值与scale_fm相乘进行缩放,之后饱和截取即完成特征图的量化.

3.3　合并卷积层与归一化层

DSC包含归一化层,在训练网络模型时,归一化层能加速网络收敛,控制过拟合,解决梯度消失和梯度爆炸的问题. 当模型训练完成后,所有参数都已固定下来,此时对网络中的卷积层参数和归一化层参数进行合并,如图4所示,可以有效简化网络结构,减少计算量,提高计算效率.

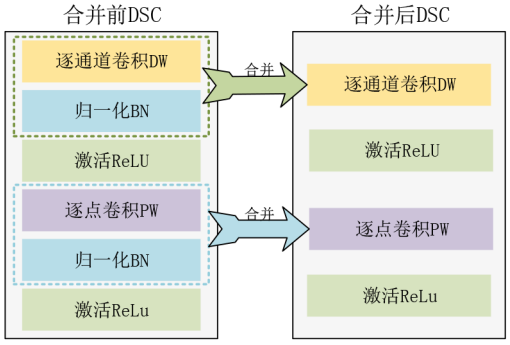

图4　合并卷积层与归一化层

设卷积输出为$\boldsymbol{y}_1$,输入为$\boldsymbol{x}$,权重为$\boldsymbol{w}$,偏差为$\boldsymbol{b}$,$\boldsymbol{x}$、$\boldsymbol{w}$和$\boldsymbol{b}$均为向量,归一化层输出为$\boldsymbol{y}_2$,均值为μ,标准差为σ,缩放系数为γ,缩放偏移为β,取$\varepsilon=1\mathrm{e}^{-6}$,则卷积计算公式为

$$\boldsymbol{y}_1=\boldsymbol{w}\boldsymbol{x}+\boldsymbol{b} \tag{10}$$

归一化层计算公式为

$$\boldsymbol{y}_2=\gamma\frac{\boldsymbol{y}_1-\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{11}$$

其中:

$$\mu=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \tag{12}$$

$$\sigma^2=\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{x}_i-\mu)^2 \tag{13}$$

将式(10)代入式(11),设W为融合后权重,$\boldsymbol{B}$为融合后偏置,合并后有输出:

$$\begin{aligned}\boldsymbol{y}_3&=\gamma\frac{\boldsymbol{y}_1-\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta\\&=\gamma\frac{(\boldsymbol{w}\boldsymbol{x}+\boldsymbol{b})-\mu}{\sqrt{\sigma^2+\varepsilon}}+\beta\\&=W\boldsymbol{x}+\boldsymbol{B}\end{aligned} \tag{14}$$

其中:

$$W=\frac{\boldsymbol{w}\gamma}{\sqrt{\sigma^2+\varepsilon}} \tag{15}$$

$$\boldsymbol{B}=\gamma\frac{(\boldsymbol{b}-\mu)}{\sqrt{\sigma^2+\varepsilon}}+\beta \tag{16}$$

融合后不再有归一化层计算,减小了模型尺寸,节省了计算资源,为前向推理带来性能上的提升.

3.4　偏置及缩放系数定点化

为进一步提高乘法器资源的利用率,将偏置及缩放系数等浮点型参数定点化. 定点化前后计算示意图分别如图5和图6所示.

图5为每层前向推理计算过程的示意图(定点化前),卷积得到的结果首先进行反量化,之后加上偏置Bias,复原为当前层真实输出后,乘以下一层的量化系数,同时进行饱和截断,得到下一层8 bit整型的特征图输入.

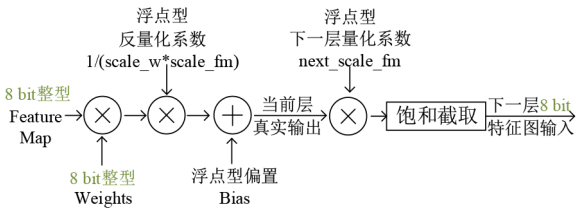

图5　定点化前的每层计算过程

推理过程中,偏置参数及量化参数均为浮点数,将其定点化可减少乘法器资源消耗,进一步提升计算频率. 最终前向推理计算过程(定点化后)如图6所示.

其中,推理计算过程定点化中有两个关键:一是定点化系数确定,二是fetch操作.

定点化系数确定　为将浮点数定点化,将前向推理中出现的浮点数单元进行合并,并放大取整,最终系数采用32位整型数保存. 设当前层卷积累加结果

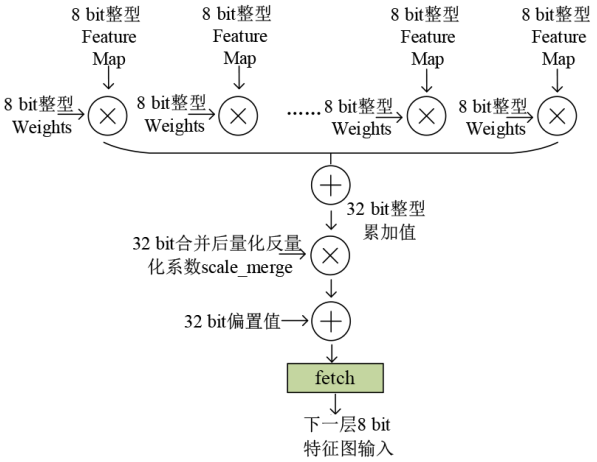

图6 定点化后的每层计算过程

为result，权重量化系数为scale_w，当前层特征图量化系数为scale_fm，偏置为bias，下一层量化系数为scale_next_fm.

设合并及放大取整后的反量化系数为scale_merge，偏置系数为bias_merge，放大倍数为shift_coe，下一层输入为next_input，有

$$\text{next\_input}=\text{fetch}[(\text{result}\times\text{scale\_merge}+\text{bias\_merge})] \tag{17}$$

其中：

$$\text{bias\_merge}=\text{int}(\text{bias}\times\text{scale\_next\_fm}\times\text{shift\_coe}) \tag{18}$$

$$\text{scale\_merge}=\text{int}\left(\frac{\text{scale\_next\_fm}\times\text{shift\_coe}}{\text{scale\_}w\times\text{scale\_fm}}\right) \tag{19}$$

shift_coe在本文中取2^{16}，采取该方法可将前向推理中涉及到的浮点数全部定点化.

fetch操作 对结果的缩小还原以及饱和截断，在FPGA中，可由式(17)及图6中的fetch操作进行. 图7为fetch操作示意图.

fetch操作进行32 bit到8 bit的转换，实质上完成了ReLu激活及饱和截断两个过程. 用定点化后的数据计

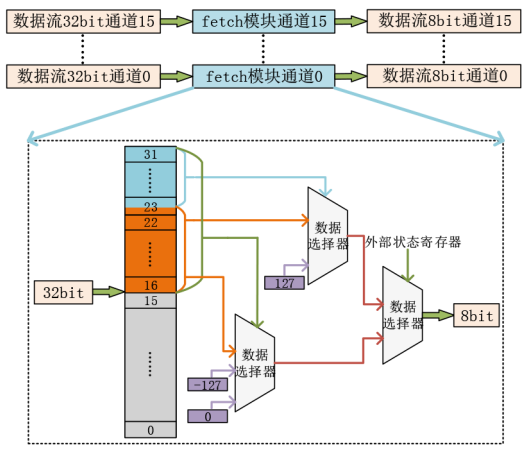

图7 fetch操作示意图

算后，需要将结果缩小为之前的1/shift_coe，对于FPGA来讲，可直接取第16位到第23位，中途需要对输入数据的正负进行判断：若为正，则进行饱和截断；若为负，则输出激活后的值0.

4 硬件加速结构设计

4.1 整体框架

本研究的目标板为Xilinx公司的ZC706开发板，具有可编程逻辑(Progarmmable Logic，PL)侧和端系统(Processing System，PS)侧.

Skynet算法加速部分使用PL侧的逻辑实现，在ZC706平台开展验证，同时PL侧承担状态管理模块、参数配置模块、使能计算流模块、(Direct Memory Access，DMA)数据处理模块以及存储偏移位置模块的实现.

PS侧负责搭建测试环境，开展初始化及配置参数等任务，获取图像数据并存放于内存，同时将PL侧返回的结果进行非极大值抑制处理，抑制多余的预测框，得到最佳的目标预测位置.

PS侧初始化完成并启动整个系统后，DMA数据处理模块将图像数据从DDR搬运到片上，之后PL侧加速器自动开始计算流程.

加速器整体框架设计图如图8所示.

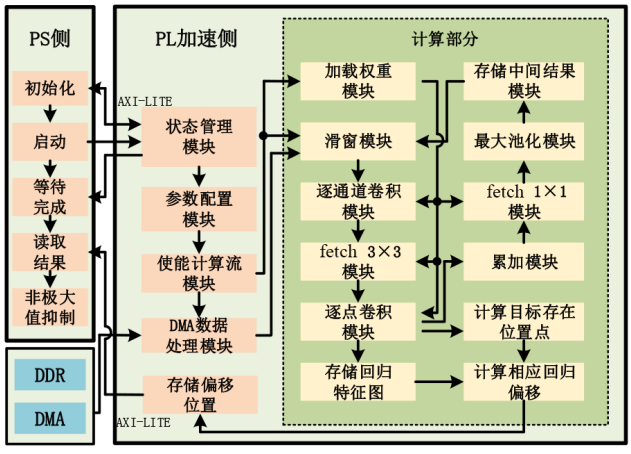

图8 整体加速器结构框图

PL侧根据参数配置模块的数据，首先进行权重的加载，之后进行图像数据的滑窗输出，其中，逐通道卷积、fetch 3×3、逐点卷积、最大池化以及fetch 1×1模块都是同时进行运算，数据从最开始的逐通道卷积流动到最后的etch 1×1操作，在滑窗过程以及池化过程中，有部分数据是无效的，因此每个模块都有一数据有效信号，该信号从滑窗模块传出，伴随整个计算过程.

计算完成后，在PL侧寻找置信度特征图中大于阈

值的点,并找到对应位置偏移特征图的偏移数据,之后计算出偏移后的位置数据,将计算出的位置数据存储到片上,由PS侧读取后进行非极大值抑制处理,得到最终的目标回归框.

4.2　滑窗设计

采用滑窗设计可提高计算效率. 本文采用移位寄存器对数据进行滑窗缓存,可在一个周期同时输出$K\times K$个有效数据,无需进行烦琐的地址跳转.

图9为一个5×5大小的特征图与3×3大小的卷积核通过滑窗方式进行卷积计算的示意图. 其中,Shift ram为移位寄存器,在移位寄存器中,新数据输入会出现在首端位置,旧数据会逐步往末端位置移动,当移动到末端位置时便会输出. 利用移位寄存器,缓存2行数据后,即可每一个时钟周期同时输出卷积窗口的3×3个数据,卷积计算时则直接使用这9个寄存器的数据,可一个周期输出一个有效结果.

通常来讲,输入图像的尺寸是固定的,这意味着推理过程中,所有特征图的尺寸都能被提前计算,根据特征图尺寸以及卷积的步长,当计算到下一层时,可根据实际情况相应地实时配置更改每一层所需的Shift ram长度.

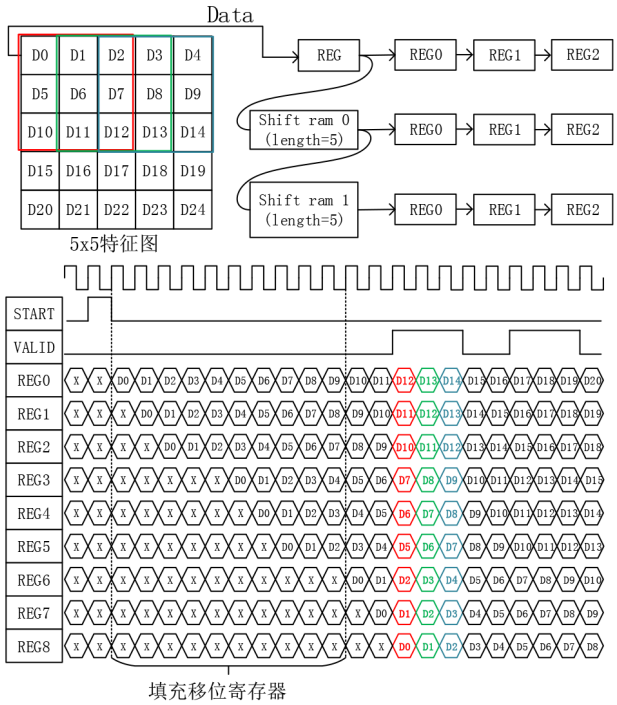

图9　卷积滑窗示意图

采用滑窗形式计算卷积时,消耗的周期数与输出特征图尺寸相关. 设一次卷积输出特征图尺寸为W_{out}和H_{out},卷积核尺寸为K,设此次卷积输出周期数为T_{clk},有

$$T_{clk}=W_{out}\times H_{out}+(K-1)\times H_{out}+W_{out}\times(K-1)+1 \quad (20)$$

在输入带宽一定的情况下,该方法能最快实现一张特征图的遍历,配合并行通道计算,能够大幅度加快当前层的卷积计算.

4.3　并行/流水化DSC计算

4.3.1　卷积计算并行性分析

卷积神经网络前向推理中,卷积是最耗时和耗资源的过程,因此对卷积过程进行并行性分析.

如图10所示,卷积过程中存在三个层次的并行结构.

(1)输入通道并行. 卷积中各输入通道之间无数据依赖,且输入特征图有各自的权重系数,因此可一次开展多个通道的计算.

(2)输出通道并行. 卷积输出时,各输出通道之间的权重系数是独立的,此时可共享输入通道特征值,用这些值同时计算多个输出通道,实现输出通道并行.

(3)像素并行. 应用滑窗流水的形式计算时,每一个周期出现$K\times K$个有效数据,这$K\times K$个数据间同样无数据依赖,可并行计算.

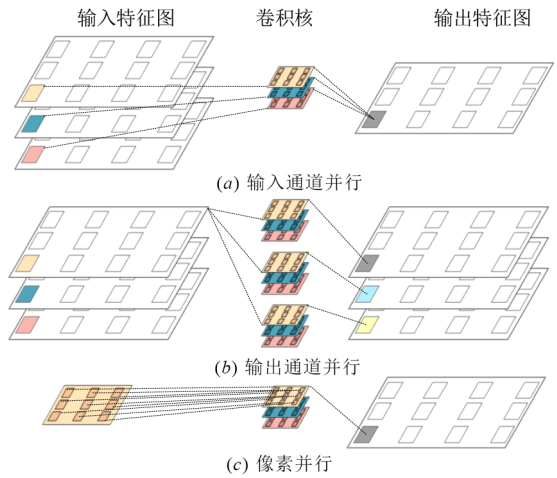

图10　三种并行方式

4.3.2　层内流水分析

层内流水对资源消耗更友好. 层内计算主要由合并后的DSC进行,特征图首先经过逐通道卷积,此时通道间无信息交流,输入与输出通道一一对应,接着经过fetch操作,此时仍然是分通道进行,通道间无数据交流,再之后经过逐点卷积,逐点卷积与标准卷积的区别在于卷积核大小为1×1,每输出一张完整的特征图需要所有输入特征图参与,输入通道间存在信息交流. 考虑到之前3×3逐通道卷积通道间无信息交流,因此可将3×3逐通道卷积以及fetch 3x3视为1×1逐点卷积预处理操作,整个流程可完全流水化.

4.3.3 并行流水化DSC

基于以上分析，并考虑具体资源使用情况，将并行度设置为输入16通道并行，输出16通道并行，像素9并行.

对于输出通道，每输出16通道，需要所有输入通道的值参与运算，但输入通道只有16并行，因此需要累加1×1逐点卷积的结果，当输入通道全部被计算后，此时累加的值为完整的16通道输出值，对此时的通道输出值做fetch 1x1的操作，即得到最终的16通道输出值.

图11为DSC的流水化并行化计算示意图.

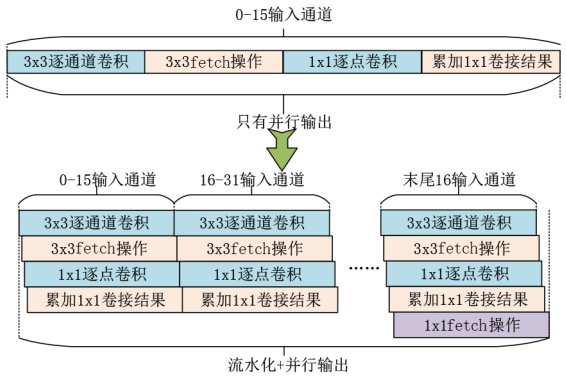

图11 流水化及并行计算DSC

用这种结构进行运算时，无需将3×3逐通道卷积结果进行保存，每次计算1×1逐通道卷积时，重新计算一遍3×3逐通道卷积即可. 由于所有操作均为流水操作，相邻计算之间只相隔几个周期，因此重新计算在节省大量存储资源的同时，并不会带来时间上的消耗.

5 实验结果

本文提出的检测网络的训练在CPU为i5-4460，GPU为GTX 1060的个人计算机上完成，同时检测网络的纯软件版本性能测试同样在该平台进行，硬件版本性能测试则在FPGA开发板ZC706上完成.

5.1 定点化实验

将本文优化后的Skynet网络模型在UA-DETRAC数据集上进行训练，得到软件版本所需的参数，训练完成后对其进行定点化，得到硬件版本所需的参数，对比定点化前后网络模型在不同平台下检测性能.

UA-DETRAC数据集由真实道路交通摄像头拍摄的视频组成，视频原始分辨率为960×540，缩放至160×160作为网络训练及验证的数据集. 本研究从缩放后的UA-DETRAC数据集中选取3 000张进行训练，600张进行测试，测试目标为数据集中真实运行车辆，目标尺寸不限，选用平均精度(AP)作为目标检测的评价指标. 以一张图像为例，卷积神经网络推理出有N个目标，其中正确目标有K个，实际上该图像有M个真实目标，则准确率P为

$$P=\frac{K}{N} \tag{21}$$

召回率R为

$$R=\frac{K}{M} \tag{22}$$

检测出目标后，若目标框与真实框的交并比(Intersection-over-Union，IoU)大于0.5则为正确检测，设一共检测D张图，则最终AP值为PR曲线下的面积.

$$\mathrm{AP}=\sum_{i=1}^{D}P_i\times\Delta R_i \tag{23}$$

模型大小由参数规模和参数的数据类型确定，即参数量与参数数据类型所占存储空间的乘积. 在测试集上开展验证，在GPU和CPU上进行纯软件版本的前向推理，在FPGA上进行硬件实现版本的前向推理，最终结果如表2所示.

表2 纯软件实现与硬件实现性能对比

	纯软件版本		硬件实现版本
权重数据类型	32 bit浮点型		8 bit整型
测试平台	GPU	CPU	FPGA
模型大小(MB)	1.41		**0.359**
平均精度(AP)	0.770		0.752
精度损失	-		2.34%
功耗(W)	120	84	**8.916**
速度(FPS)	153.8	10.92	115.1
能效比(FPS/W)	1.28	0.13	**12.91**

由表2可知，硬件实现版本模型大小缩小为纯软件版本的25.5%，精度相较于纯软件实现版本仅下降了2.34%. 硬件实现版本的前向推理速度相较于通用CPU提速了11倍，功耗只有通用CPU的10.6%，速度达到了通用GPU的75%，功耗却仅为通用GPU的7.43%，能效比则分别是CPU的99倍和GPU的10倍.

图12给出了纯软件版本检测结果以及硬件实现版本的检测结果. 结合已检出的目标及表2，硬件实现版本相较于纯软件实现版本的误差在可接受范围内.

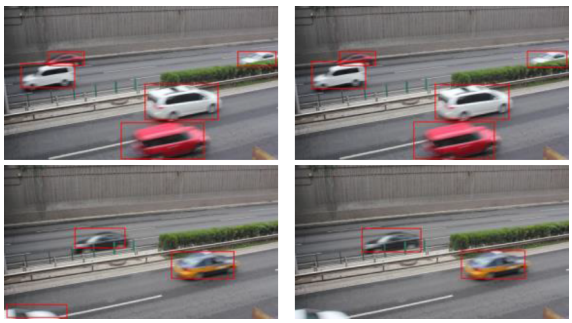

图12 纯软件版本与硬件版本检测结果对比图

5.2 FPGA上板运行结果

实验目标板为ZC706,最终实现的加速器消耗资源如表3所示. 由表3可知,存储资源BRAM与计算资源DSP消耗较高,但查找表LUT、查找表存储块LUTRAM以及触发器FF消耗极少.

表3 最终资源利用率

资源	本文消耗	总数量	利用率
LUT	23 912	218 600	10.94%
LUTRAM	5 222	70 400	7.42%
FF	24 245	437 200	5.55%
BRAM	398.50	545	73.12%
DSP	499	900	55.44%

图13展示了板卡与计算机的连接情况,上位机将验证集图像通过以太网传输到板卡,每发送一张图像,启动加速器进行检测,检测结果通过以太网返回到上位机进行显示.

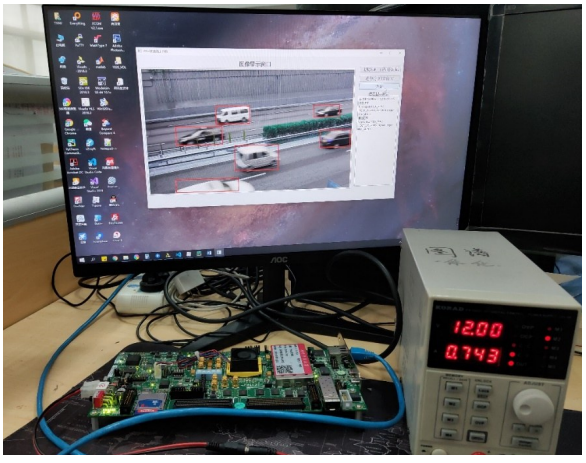

图13 板卡与计算机连接情况

选取文献[6]、文献[8]、文献[14]开展对比实验. 其中,文献[6]首次提出Skynet网络,同时在FPGA上进行加速实现;文献[8]则是设计了二值化的YOLOv2,降低操作量以提高推理速度;文献[14]同样采用了滑窗操作及量化操作,并对轻量型网络SSDLite进行加速实现. 设各文献硬件实现精度为HW_p,软件实现精度为SW_p,则精度损失$loos_p$为

$$\text{loss}_P = \frac{\text{SW}_P - \text{HW}_p}{\text{SW}_P} \times 100\% \tag{24}$$

各文献方法与本文提出的方法对比情况如表4所示.

如表4所示,本文方法在速度和能效比上取得最优,与文献[6]相比,本文方法在速度上提高了4.59倍,能效比提高了3.74倍,由于剪枝、量化以及定点化缩放系数等操作的影响,精度损失略有增加,开发平台以及资源利用率的不同也导致最终功耗略大于文献[6]. 与文献[8]和文献[14]相比,本文方法在速度、能效比和精度损失三个指标上都有较大提升,特别是能效比和精度损失提升明显.

表4 同类工作对比

方法	精度损失	速度(FPS)	功耗(W)	能效比(FPS/W)
文献[6]	2.05%	25.05	7.26	3.45
文献[8]	3.94%	109.3	18.29	5.97
文献[14]	7.23%	64.8	9.9	6.55
本文	2.34%	**115.1**	8.916	**12.91**

综上所述,本文方法与CPU和GPU上实现的纯软件版本相比,有效降低了功耗,提高了能效比;同时,与基于FPGA加速CNN的同类方法对比,显著提高了速度和能效比,且精度损失只略高于最优.

6 结论

本研究基于FPGA提出一种目标检测卷积神经网络加速方案,为卷积神经网络在边缘设备上部署困难提出了解决方法. 采用轻量型网络Skynet作为特征提取网络,对结构进行修改,将训练好的模型进行量化以及参数合并,并将缩放及偏置参数定点化,采用多通道并行及像素并行计算,设计完全流水化的硬件加速结构,合并前向推理中逐通道卷积及逐点卷积操作. 最终实验表明,本研究在速度上,相较于CPU提升了11倍,略低于GPU,能效比相较于CPU和GPU分别提升了99倍和10倍;与同类CNN加速工作对比,本文方法显著提高了速度与能效比,有效提高了基于CNN的检测网络在边缘设备上的实现效率,具有较强的应用价值.

参考文献

[1] 李旭冬,叶茂,李涛. 基于卷积神经网络的目标检测研究综述[J]. 计算机应用研究, 2017, 34(10): 2881-2886, 2891.
LI X D, YE M, LI T. Review of object detection based on convolutional neural networks[J]. Application Research of Computers, 2017, 34(10): 2881-2886, 2891. (in Chinese)

[2] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE Press, 2016: 779-788.

[3] GIRSHICK R. Fast R-CNN[C]//Proceedings of The IEEE International Conference on Computer Vision. Nice: IEEE Press, 2015: 1440-1448.

[4] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[EB/OL]. (2017-04-17) [2020-12-28]. https://arxiv.org/abs/1704.04861.

[5] CHOLLET F. Xception: Deep learning with depthwise separable convolutions[EB/OL]. (2016-10-07) [2017-04-04]. https://arxiv.org/abs/1610.02357.

[6] ZHANG X, LU H, HAO C, et al. SkyNet: A hardware-efficient method for object detection and tracking on embedded systems[J]. Proceedings of Machine Learning and Systems, 2020, 2: 216-229.

[7] NAKAHARA H, YONEKAWA H, FUJII T, et al. A lightweight YOLOV2: A binarized CNN with a parallel support vector regression for an FPGA[C]//Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2018: 31-40.

[8] NAKAHARA H, YONEKAWA H, SATO S. An object detector based on multiscale sliding window search using a fully pipelined binarized CNN on an FPGA[C]//International Conference on Field Programmable Technology (ICFPT). Tokyo: IEEE Press, 2017: 168-175.

[9] NGUYEN D T, NGUYEN T N, KIM H, et al. A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection[J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2019, 27(8): 1861-1873.

[10] 刘勤让, 刘崇阳. 利用参数稀疏性的卷积神经网络计算优化及其FPGA加速器设计[J]. 电子与信息学报, 2018, 40(6): 1368-1374.
LIU Q R, LIU C Y. Calculation optimization for convolutional neural networks and FPGA-based accelerator design using the parameters sparsity[J]. Journal of Electronics & Information Technology, 2018, 40(6): 1368-1374. (in Chinese)

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

[12] QIU J, WANG J, YAO S, et al. Going deeper with embedded FPGA platform for convolutional neural network [C]//Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2016: 26-35.

[13] DING C, WANG S, LIU N, et al. REQ-YOLO: A resource-aware, efficient quantization framework for object detection on FPGAs[C]//Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2019: 33-42.

[14] FAN H, LIU S, FERIANC M, et al. A real-time object detection accelerator with compressed SSDLite on FPGA [C]//International Conference on Field-Programmable Technology (FPT). Piscataway: IEEE Press, 2018: 14-21.

[15] BAI L, ZHAO Y, HUANG X. A CNN accelerator on FPGA using depthwise separable convolution[J]. IEEE Trans on Circuits and Systems II: Express Briefs, 2018, 65(10): 1415-1419.

[16] ZENG H, CHEN R, ZHANG C, et al. A framework for generating high throughput CNN implementations on FPGAs[C]//Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 2018: 117-126.

[17] YU Y, WU C, ZHAO T, et al. Opu: An fpga-based overlay processor for convolutional neural networks[J]. IEEE Trans on Very Large Scale Integration (VLSI) Systems, 2019, 28(1): 35-47.

[18] WU D, ZHANG Y, JIA X, et al. A high-performance CNN processor based on FPGA for MobileNets[C]//the 29th International Conference on Field Programmable Logic and Applications (FPL). Barcelona: IEEE Press, 2019: 136-143.

[19] 蹇强, 张培勇, 王雪洁. 一种可配置的CNN协加速器的FPGA实现方法[J]. 电子学报, 2019, 47(7): 1525-1531.
JIAN Q, ZHANG P Y, WANG X J. An FPGA implementation method for configurable CNN co-accelerator[J]. Acta Electronica Sinica, 2019, 47(7): 1525-1531. (in Chinese)

[20] ZHAO R, NIU X, WU Y, et al. Optimizing CNN-based object detection algorithms on embedded FPGA platforms [C]//International Symposium on Applied Reconfigurable Computing. Berlin: Springer, 2017: 255-267.

[21] HAN S, MAO H, DALLY W J, et al. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[EB/OL]. (2015-10-01) [2016-02-15]. https://arxiv.org/abs/1510.00149.

作者简介

唐维伟　男,1997年生,重庆人. 华中科技大学人工智能与自动化学院硕士生. 主要研究方向为人工智能、计算机视觉、嵌入式高性能计算.
E-mail: M201972649@hust.edu.cn

钟　胜　男,1972年生,湖北人. 华中科技大学人工智能与自动化学院副院长、教授、博士生导师.主要研究方向为数字图像处理、模式识别在成像自主导航和成像自动目标探测及其实时实现.
E-mail: zhongsheng@hust.edu.cn

卢金仪　男,1995年生,广西玉林人. 华中科技大学人工智能与自动化学院硕士生. 主要研究方向为计算机视觉、深度神经网络、嵌入式高性能计算.
E-mail: 15071437004@139.com

颜露新　男,1978年生,湖北人. 华中科技大学人工智能与自动化学院教授、博士生导师. 主要研究方向为嵌入式视频/图像实时处理系统开发、图像恢复理论与算法、目标检测与跟踪.
E-mail: yanluxin@gmail.com

谭富中　男,1997年生,重庆人. 华中科技大学人工智能与自动化学院硕士生. 主要研究方向为自动目标识别,并行计算,专用处理硬件结构.
E-mail: M202072755@hust.edu.cn

邹　旭　男,1991年生,湖北武汉人. 华中科技大学人工智能与自动化学院博士后. 主要研究方向为计算机视觉、图像理解、目标检测识别.
E-mail: zoux@hust.edu.cn

徐文辉(通讯作者)　男,1981年生,湖北人. 华中科技大学博士. 主要研究方向为目标检测、并行加速计算.
E-mail: xuwenhui@hust.edu.cn