0254-6124/2023/43(5)-0950-13 *Chin. J. Space Sci.* 空间科学学报

JIANG Kangning, ZHOU Hai, BIAN Chunjiang, WANG Ling. Hardware Acceleration of YOLOv5s Network Model Based on Aerospace-grade FPGA (in Chinese). *Chinese Journal of Space Science*, 2023, **43**(5): 950–962. DOI:10.11728/cjss2023.05.2022–0044

基于宇航级 FPGA 的 YOLOv5s 网络模型硬件加速*

蒋康宁[1,2] 周 海[1] 卞春江[1] 汪 伶[1,2]

1(中国科学院国家空间科学中心 北京 100190)

2(中国科学院大学 北京 100049)

摘 要 由于遥感图像具有分辨率高和背景信息复杂的特点，其对目标检测的精确性和鲁棒性要求越来越高，因此遥感图像处理领域逐渐引入了卷积神经网络算法。然而此类算法通常模型复杂且计算量庞大，难以在空间与资源受限的星上平台高效运行。针对这一问题，提出一种基于宇航级现场可编程门阵列（Filed Programmable Gate Array, FPGA）的卷积神经网络硬件加速架构，并选用 YOLOv5s 作为目标网络，采用输入与输出通道并行展开以及数据流水线控制的策略进行架构设计。实验结果表明，在使用该处理架构加速 YOLOv5s 的推理阶段，卷积模块的工作频率可以达到 200 MHz，其运算性能高达 394.4GOPS（Giga Operations Per Second），FPGA 的功耗为 14.662 W，数字信号处理（Digital Signal Processing, DSP）计算矩阵的平均计算效率高达 96.29%。

关键词 星上系统，卷积神经网络，硬件加速，现场可编程门阵列

中图分类号 V557

Hardware Acceleration of YOLOv5s Network Model Based on Aerospace-grade FPGA

JIANG Kangning[1,2] ZHOU Hai[1] BIAN Chunjiang[1] WANG Ling[1,2]

1(*National Space Science Center, Chinese Academy of Sciences, Beijing* 100190)

2(*University of Chinese Academy of Sciences, Beijing* 100049)

Abstract With the rapid development of my country's remote sensing engineering technology, the resolution of remote sensing images that can be obtained is getting higher and higher, and the image background information is also more complex, which brings great challenges to the accuracy and robustness of traditional target detection methods. With the development of deep learning, the convolutional

* 中国科学院青年创新促进会项目资助（E0293401）

2022-08-19 收到原稿，2022-11-25 收到修定稿

E-mail：18838980607@163.com

©The Author(s) 2023. This is an open access article under the CC-BY 4.0 License (https://creativecommons.org/licenses/by/4.0/)

neural network algorithm has better performance in terms of detection accuracy and robustness than traditional methods. In order to improve the accuracy and robustness of remote sensing image target detection with high resolution and complex background, the remote sensing image target detection algorithm based on convolutional neural network is applied in this field. However, such algorithms usually have complex models and a large amount of calculation, making it difficult to run efficiently on space and resource-constrained on-board platforms. Aiming at this problem, a convolutional neural network forward inference hardware acceleration architecture based on aerospace-grade FPGA (Field Programmable Gate Array) is proposed, and the YOLOv5s network model is selected as the target algorithm for architecture design. Since the main body of the YOLOv5s network is composed of a large number of convolutional layers, the center of gravity of the accelerator architecture design lies in the convolutional layer. In the design of the architecture, the parallel expansion of input channels and output channels and the optimization strategy of data pipeline control are adopted to effectively improve the real-time processing performance of the inference stage is improved. The experimental results show that when using this processing architecture to accelerate the inference stage of YOLOv5s, the operating frequency of the convolution module can reach 200 MHz, and its computing performance can reach 394.4GOPS (Giga Operations Per Second). The power consumption is 14.662 W, and the average calculation efficiency of the DSP (Digital Signal Processing) calculation matrix is as high as 96.29%. It shows that the use of FPGA for hardware acceleration of convolutional neural networks in resource and power constrained on-board platforms has significant advantages.

Key words On-board system, Convolutional Neural Network (CNN), Hardware acceleration, Filed programmable gate array

0 引言

随着中国航天与遥感工程技术的高速发展，目前卫星所能采集到的空间遥感图像具有很高的分辨率和复杂多样的场景信息，对遥感数据的获取量也日益增加。面对如此大量的复杂高分辨率遥感图像，传统的遥感图像处理方法在精确性和鲁棒性方面的缺陷愈发明显[1]，而基于卷积神经网络（Convolutional Neural Network, CNN）[2]的目标检测算法在这两个方面具有较为明显的优势，将其引入遥感图像处理领域[3]，能够有效推动该领域的发展。然而卷积神经网络算法通常模型复杂且计算量庞大，如何将其部署在空间与资源有限的星上平台并高效运行，是本文研究的重点。

自从卷积神经网络被引入到图像处理领域，目标检测算法得到高速发展。其被分为两大类：一是基于候选区域的两阶段目标检测，典型的算法有 R-CNN[4]，Fast R-CNN[5]，Faster R-CNN[6]，R-FCN[7] 和 Mask R-CNN[8] 等；二是不需要候选区域的单阶段目标检测，主要包括 SSD[9] 和 YOLO[10–13] 系列算法。YOLO 系列算法的核心思想是利用整张图作为网络的输入，直接在输出层回归 Bounding Box（边界框）的位置及其所属的类别，其中 YOLOv5 是目前 YOLO 系列中最新的目标检测算法。相比前几代算法，YOLOv5 检测速度更快，更加轻量化，十分适合在对资源、功耗、实时性要求严格的星上系统部署。此外 YOLOv5 网络对遥感图像的检测具有良好的性能，Tan 等[14] 提出了一种基于改进 YOLOv5 的舰船目标检测方法，该方法将目标检测框的长宽作为参数进行考虑并将损失函数进行曲线优化，同时结合坐标注意力机制，实现了对舰船目标的高速与高精度检测，将检测精度由原来的 92.3% 提升到 96.7%，平均精度均值（Mean Average Precision, mAP）由原来的 92.5% 提升到 97.2%。

FPGA 具有体积小、功耗低、可重构性和性能强等优势，在加速卷积神经网络方向受到了越来越多的

关注[15], 此外相比大体积和高功耗的图形处理器(Graphics Processing Unit, GPU), FPGA 更加适合在资源和空间受限的星上平台使用。Zhang 等[16] 针对 YOLOv2-tiny 网络模型提出了一种基于 FPGA 的低延迟加速器架构, 通过引入双符号乘法校正电路来减少卷积运算的计算时间, 取得了较高的运算速率, 但加速器在设计中对数据采用了 8 bit 量化, 导致精度有所损失, 而且硬件资源 DSP 的使用率也不是很高。Bi 等[17] 通过对 YOLOv2 算法研究, 设计了特殊的浮点数矩阵乘法单元和双缓存数据处理电路, 提高了卷积神经网络的计算速度, 但加速器整体架构的工作频率较低。Zhang 等[18] 提出了一种基于 ARM + FPGA 异构架构的可重构卷积神经网络加速器, 二者通过高级可扩展接口(Advanced eXtensible Interface, AXI)总线进行通信, FPGA 接收 ARM 发送的配置信号, 分时完成各卷积层的运算, 最终达到了较高的峰值性能。Nguyen 等[19] 为避免数据与外部存储的频繁交互, 提出了一种 Tera-OPS 流式传输架构设计, 通过重用中间数据以最小化每个卷积层的输入缓冲区大小, 同时避免了对片外存储器的访问, 实现了较高的处理速率, 但其为降低数据存储量, 采用的是二值化网络, 导致数据精度下降。使用 FPGA 加速卷积神经网络的性能与片上资源关系甚密, 如何利用有限的硬件资源设计出高效的加速器处理架构是很重要的研究问题[20]。

针对星上平台空间与资源受限, 难以将模型复杂、计算量庞大的卷积神经网络目标检测算法部署在星上系统高效运行的问题, 本文通过对 YOLOv5s 算法模型的详细分析, 设计了一种高性能实时并行处理架构, 以及对卷积模块、池化模块、切片模块和残差模块基于 FPGA 进行相应的分析与电路设计; 此外针对 FPGA 片上存储资源不足的问题, 提出对特征图与权重参数进行数据分块与复用的优化策略, 进而对该架构的实时处理性能进行仿真测试验证, 从计算速度和资源利用率等方面对设计进行综合性能分析, 并与其他算法模型在 FPGA 上的加速状况进行对比。

1 加速器系统架构设计

1.1 整体架构设计

YOLOv5s 网络结构如图 1 所示[21], 主要由 4 个部分组成, 即输入端、Backbone 主干网络、Neck 网络和 Prediction 输出端。

根据对 YOLOv5s 网络模型的分析, 基于 FPGA 平台设计了如图 2 所示的加速器整体架构, 由于卷积运算占据了 YOLOv5s 网络模型绝大部分的运算量, 因此本端主要通过在 FPGA 上运行卷积运算实现 YOLOv5s 的前向推理加速。在图 2 所示的处理器架构中, 由于 FPGA 内部的存储资源有限, 特征图和权重参数的数据量相对而言过于庞大, 无法将其全部缓存在 FPGA 的片上缓存中, 因此使用双倍数据速率(Double Data Rate, DDR)外部存储器储存输入特征图、输入权重和输出特征图数据; 存储访问接口控制模块通过控制内存接口生成器(Memory Interface Generator, MIG)核调度外部存储器与片上缓存块随机存取存储器(Block Random Access Memory, BRAM)之间数据的存储与访问; BRAM 是片上缓存资源, 外部存储器的数据无法直接与 FPGA 的计算单元进行通信, 需要通过 BRAM 缓存后再参与计算; 处理元件(Processing Element, PE)单元阵列负责完成输入特征图和输入权重的卷积运算, 其主要由 DSP 和加法器等计算单元组成。

图 2 中加速器架构的工作流程如下: 首先 FPGA 通过存储访问接口控制模块控制 MIG 核从 DDR 外部存储器中读取部分输入特征图和输入权重数据, 并将其分别缓存在输入特征图缓存队列和输入权重缓存队列中; 之后根据设计好的循环计算顺序, 依序将输入特征图数据和权重数据送入 PE 单元阵列进行处理, 处理完毕后将输出特征图的中间结果暂存在输出缓存队列中; 当计算得到输出特征点时, 将其传输到 ReLU 函数激活, 部分卷积层需要经过残差模块和池化模块的处理, 最终通过存储访问接口控制模块控制 MIG 核将输出特征图数据传输到 DDR 外部存储器中。

1.2 卷积模块设计

在 YOLOv5s 网络模型中, 卷积层占据整个神经网络 90% 以上的运算量, 消耗最多的计算资源和时间, 因此基于 FPGA 对卷积神经网络进行加速的设计侧重于卷积并行计算的优化, 以下两方面是本文着重考虑的优化方向。

(1) 在 FPGA 片上计算资源允许的范围之内, 根据卷积神经网络模型本身的特点, 选择合理的循环展

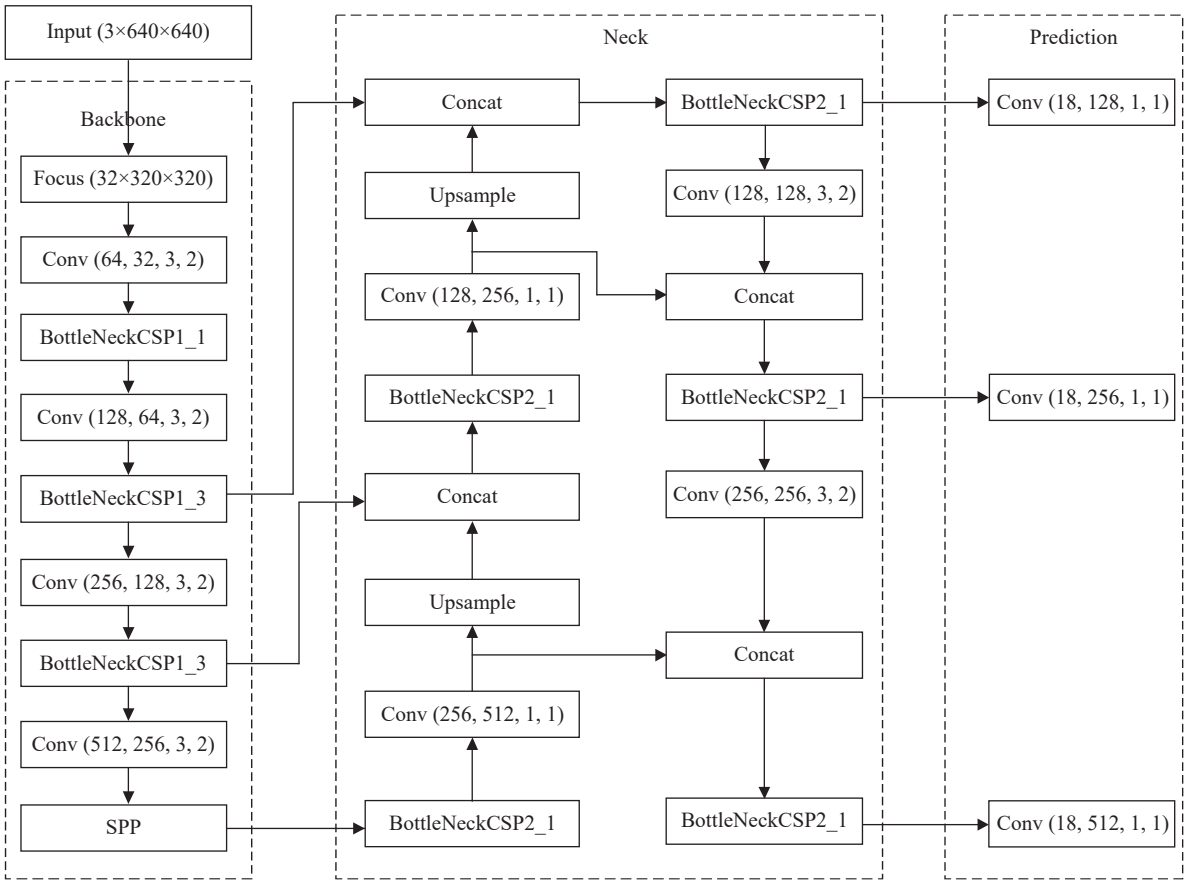


图 1　YOLOv5s 的网络结构

Fig. 1　Network structure of YOLOv5s

开方式, 尽可能地搭建最大规模的并行流水计算架构, 缩减计算阵列在流水间所需的等待时钟, 从而提升加速器架构的性能。

(2) 在 FPGA 片上缓存资源允许的范围之内, 针对数据复用进行优化设计, 最大限度地减少对外部存储的访问次数, 以节约功耗。

1.2.1　循环并行展开设计

卷积神经网络中的卷积计算由 4 个层次的循环组成, 分别为卷积核循环、输入通道循环、输入特征图循环和输出通道循环[22], 这 4 层循环有着较大的优化空间, 其并行展开的方式和维度决定了卷积计算架构的设计, 从而影响加速器优化设计的数据复用和存储访问模式, 需要根据卷积神经网络模型本身的特点来选取合适的循环展开方式和维度。

YOLOv5s 的网络模型结构较深, 卷积层数多达 70 层, 且卷积核的尺寸包括 3×3 和 1×1 两种规格, 其中卷积核尺寸为 1×1 的卷积层占据了大多数, 如果采用卷积核循环展开进行处理架构的优化设计, 会造成较大的资源浪费。 通过对 YOLOv5s 的网络参数分析可知, 对于该网络中的 70 层卷积层, 其中 66 层的输入通道数和输出通道数的最大公因数均为 32, 且各层输入特征图的最大公因数为 20×20。考虑到 FPGA 片内的乘法器数量和片上存储资源 BRAM 的数量, 为了尽可能最大化处理架构的运算并行度, 同时要满足片上存储资源的要求, YOLOv5s 网络比较适合采用输入通道和输出通道循环展开, 此时输入通道和输出通道循环展开的并行度均为 32, 因此硬件处理架构的整体并行度为 1024。

在上述处理架构中, 总共用到了 1024 个 DSP 单元, 而实际本文实验所用到的 FPGA 中总共拥有 3600 个 DSP 单元。之所以在设计中没有使用那么多计算单元, 是因为该架构是针对星上平台设计的, 考虑到太空中复杂的空间环境, 空间高能粒子造成的辐射和冲击会对 FPGA 产生诸多影响[23], 其中最主要的影响为单粒子效应 (Single Event Effect, SEE)[24]。为尽量避免单粒子效应对电路造成的负面影响, 需要

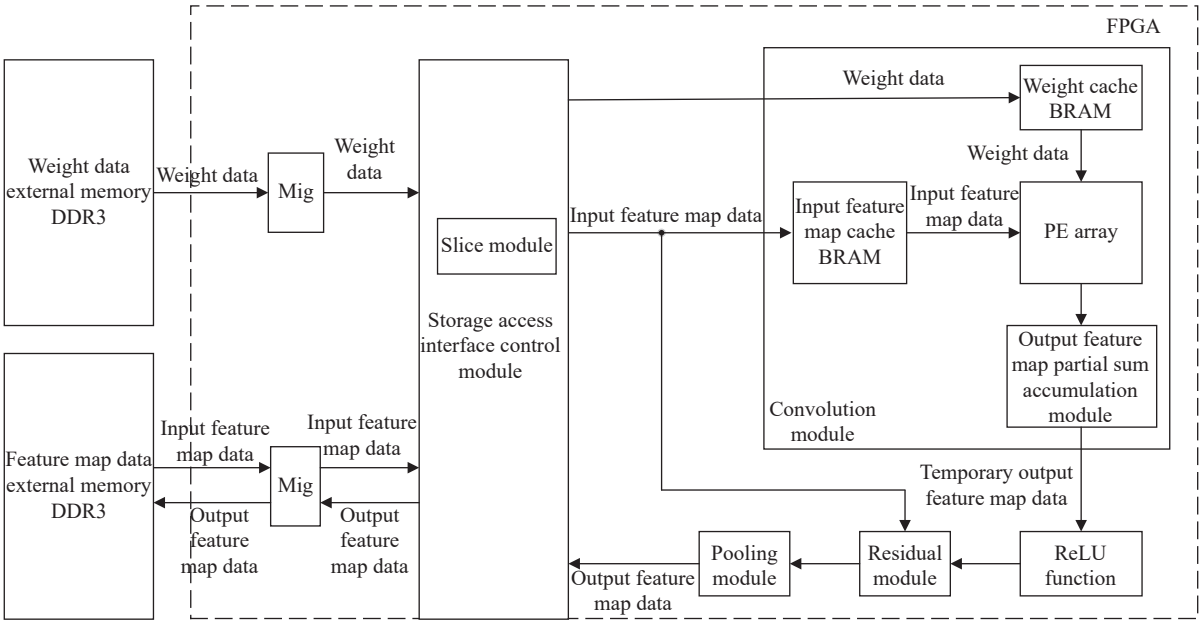


图 2 加速器整体架构设计

Fig. 2 Accelerator overall architecture design

进行抗辐照加固设计, 本文所采取的措施为三模冗余技术（Triple Modular Redundancy, TMR）[25], 因此在本次设计中 FPGA 中可用的 DSP 单元数量降低为总数的 1/3, 即 1200 个, 同时考虑到输入通道和输出通道循环展开策略, 最终将所需的 DSP 数量定为 1024 个。

1.2.2 卷积计算架构

根据 YOLOv5s 的循环展开优化分析, 设计了如图 3 所示的并行计算矩阵, 该矩阵以输入通道和输出通道进行循环展开, 总共有 32×32 个计算单元, 即需要 1024 个 DSP 单元。由于权重数据需要从缓存队列中同时为多个计算单元更新, 因此将权重缓存队列设置为 1×32×32, 也即总共有 32 个权重缓存队列, 且每个队列中包含 32 个 BRAM 权重缓存单元, 共计有 1024 个权重缓存单元。在进行权重数据更新时, 每个权重缓存队列只更新当前计算矩阵纵列的 32 个计算单元, 因此整个计算矩阵每次可更新 1024 个计算单元的权重。针对输入特征图缓存队列, 采用逐点递进的方式对计算单元阵列进行更新特征点, 更新特征点时每个输入通道将各自当前的一个特征点复制成 32 个后填充给矩阵横排的 32 个计算单元, 此时每个输入通道只更新了一个特征点, 而整个计算矩阵包含 32 个输入通道, 因此输入特征图每次可以更新 32 个特征点。当权重数据和输入特征点更新完毕后, 每个计算单元里的乘法器将权重数据和输入特征点数据进行相乘, 然后每个输出通道采用五级流水的加法树将本通道内的 32 个乘积进行相加, 即可得到一个中间结果并将其存储在中间结果缓存队列中。

1.2.3 带宽分析

根据提出的卷积计算架构, 权重参数和特征图的更新是同时进行的, 因此二者的外部缓存各自使用了一个 DDR3 存储器, 存储器的工作频率为 800 MHz, 且双沿有效, 数据位宽为 64 bit。考虑到 MIG 核的工作效率为 70% 左右, 则 DDR3 存储器的最高带宽约为 70 Gbit·s^{-1}, 即

$$R_{\mathrm{DDR3}} = 2 f_{\mathrm{DDR3}} D_{\mathrm{bw}} E_{\mathrm{MIG}}. \tag{1}$$

其中, f_{DDR3} 为存储器的工作频率, D_{bw} 为存储器的数据位宽, E_{MIG} 为 MIG 核的工作效率。针对权重参数和特征图, 本文选用的数据精度为 16 bit, FPGA 的工作频率为 200 MHz。对权重参数和特征图的带宽需求进行分析。

在卷积计算架构中, 每个时钟需要更新 1024 个权重参数, 因此权重的带宽需求高达 3200 Gbit·s^{-1}, 即

$$R_{\mathrm{weight}} = f_{\mathrm{FPGA}} N_{\mathrm{weight}} P_{\mathrm{weight}}. \tag{2}$$

其中, f_{FPGA} 为 FPGA 的工作频率, N_{weight} 为权重参数一次更新的数量, P_{weight} 为权重参数的数据精度。

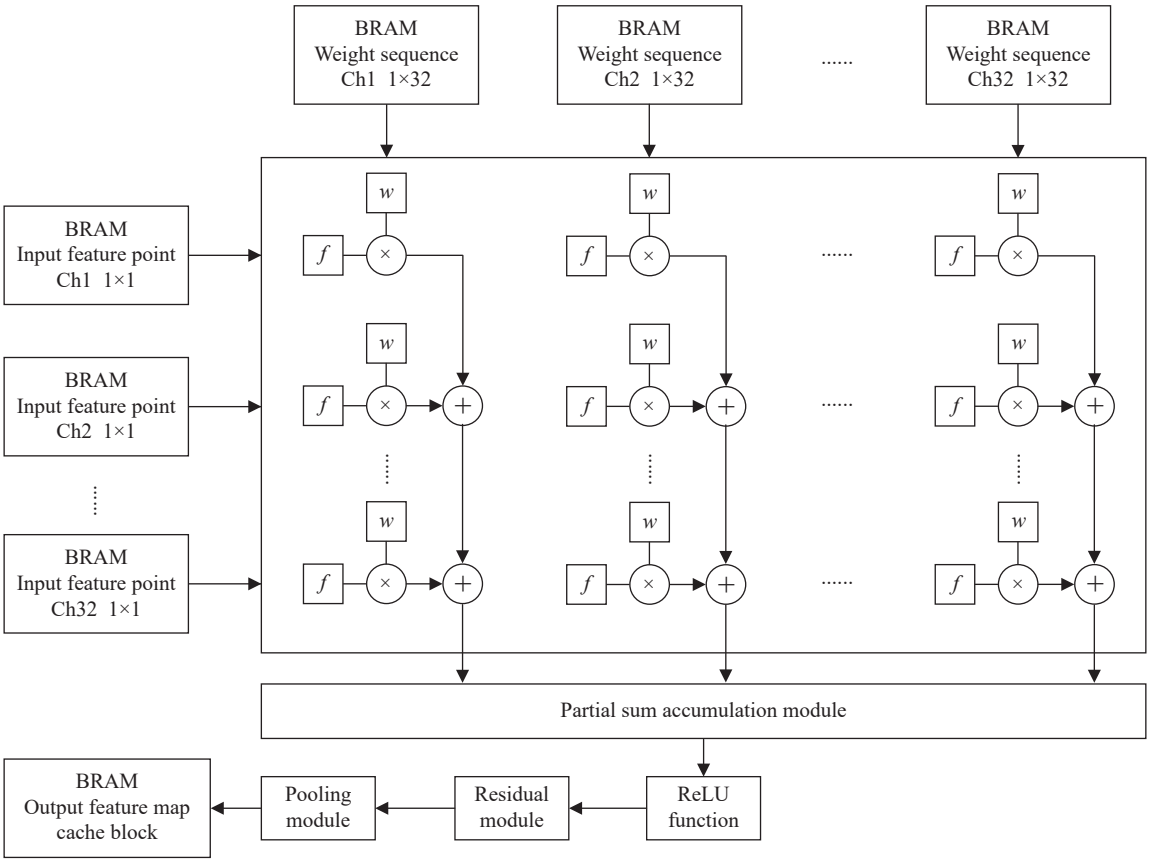


图 3 YOLOv5s 卷积层并行计算矩阵

Fig. 3 Parallel calculation matrix of YOLOv5s convolution layer

根据式(2)的计算,显然权重的带宽需求远超 DDR3 所能提供的带宽速率。如果在计算时采用对权重外部存储频繁访问的方式,则计算单元阵列需要等待的周期极长,从而会造成加速器架构的计算效率低下。由于 YOLOv5s 是一个轻量化网络,其每层的权重参数量并不算很大,以 FPGA 的片上缓存可完全存下其任意单独一层的权重,因此只需在每层卷积开始时把当前层的所有权重读入 FPGA 片上缓存中,之后便可持续复用,从而避免了权重读取成为加速器架构高效运行的瓶颈。

相比权重参数,特征图每个时钟周期只需更新 32 个特征点,而且特征图更新的带宽需求与其复用度有关,即

$$R_{\mathrm{FM}} = (f_{\mathrm{FPGA}} N_{\mathrm{FM}} P_{\mathrm{FM}})/U_{\mathrm{FM}}. \tag{3}$$

其中,f_{FPGA} 为 FPGA 的工作频率,N_{FM} 为特征点参数一次更新的数量,P_{FM} 为特征图的数据精度,U_{FM} 为特征图的复用度。式(3)表明,特征图的复用度越高,其带宽需求也就越低,因此当特征图不被复用时,其带宽需求最高,为 100 Gbit·s^{-1},略大于 DDR3 的带宽速率。其中,决定特征图复用度大小的因素根据卷积核尺寸的不同而有所区别:针对卷积核尺寸为 1×1 的层,特征图的复用度即为输出通道数量与输出通道并行度的比值;而针对卷积核尺寸为 3×3 的层,特征图复用度除了与上述比值有关以外,还由于 3×3 的卷积窗口在特征图上滑动时也会对特征图产生一定的复用,因此其有着比卷积核尺寸为 1×1 的层更高的特征图复用度。此外,虽然 DDR3 本身的最高带宽约为 70 Gbit·s^{-1},但特征图的 DDR 存储器不仅需要读出输入特征图,还需要写入输出特征图,因此其实际的特征图读取带宽速率还会有所降低,这就会导致特征图的带宽需求成为瓶颈。通过计算分析可知,只有卷积核尺寸为 1×1 且输出通道数为 64 及以下的卷积层,其带宽需求会略有瓶颈,实际上这些卷积层的计算量占比较少,而其余卷积层的理想计算效率几乎均可达到 100%,因此加速器整体上是十分高效的。

1.2.4 数据分块与数据复用

通常卷积神经网络的输入特征图和权重参数数量庞大, FPGA 内部的存储资源有限, 难以将其完全存下, 需要将完整的数据等分为多个较小的数据块, 依次读入 FPGA 片上缓存, 也即数据分块。其中输入特征图数据分块的实现方式是沿着特征图的输入通道方向、以输入通道并行度为基本单元进行划分(见图 4), 图 4 中彩色数据带即为被分块的数据, 分块后的特征图数据易于缓存和复用, 有利于节约资源和功耗开销。针对特征图的复用, 主要表现为卷积窗口在特征图上滑动时所产生的复用(见图 5), 针对尺寸为 3×3 的卷积核, 使用三行线性缓冲队列进行特征图的预取, 当卷积窗口在预取的特征图上滑动时, 当前窗口会与前一个窗口的数据存在部分重合, 即可实现特征图的复用。由于卷积窗口在行方向和列方向上均有滑动, 因此对特征图有着很高的复用率, 当滑动的步长为 1, 2 时, 数据复用率分别可达 88.9%, 55.6%。

另外, 由于 YOLOv5s 任意单独一层的权重数量并不算很多, 可由 FPGA 片上缓存全部存下, 因此对其采用了循环分块的优化方式。在每一层的卷积开始时就把当前层的所有权重数据加载至 FPGA 片上缓存, 以卷积计算矩阵的并行度为基本单元将权重划分为若干小块, 如图 6 所示。每次加载部分特征图到卷积计算矩阵的同时, 加载相对应的小块权重, 直至当前部分的特征图与所有权重计算完毕后, 加载下一部分特征图以及相对应的小块权重, 即可实现对权重数据的复用, 以此避免了因频繁访问外部存储所造成的时间与功耗开销。

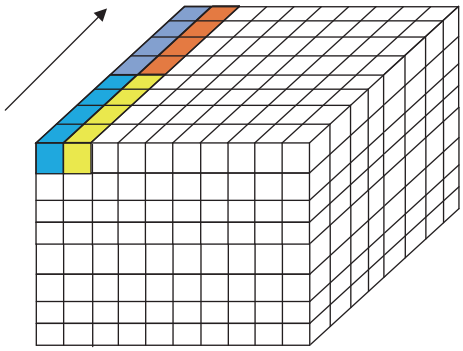


图 4 输入特征图数据分块(彩色数据带为被分块的数据)

Fig. 4 Input feature map data block (The colored data bands are the data being chunked)

1.2.5 循环计算顺序优化

根据之前的分析, 由于 YOLOv5s 网络模型本身的特点, 在基于 FPGA 实现该网络时适合选用输入通道和输出通道循环展开, 为此设计了如下循环计算顺序: 首先将被分块后的输入特征图数据沿着 32 个输入通道从外部 DDR 存储器中读入到输入特征图片上缓存区, 同时将所有的权重数据从外部存储器中读到权重片上缓冲区; 卷积计算开始后, 提取 32 个输入通道输入特征图以及相应的卷积权重到 PE 阵列进行计算, 计算完成后判断是否所有输入通道的特征图计算完毕, 如果没有则更新下一 32 个通道的输入特征图, 并对当前 32 个输出通道的权重进行相应切换, 直至完成所有输入通道的特征图与当前 32 个输出通道权重的计算; 然后判断是否所有输出通道的权重都已计

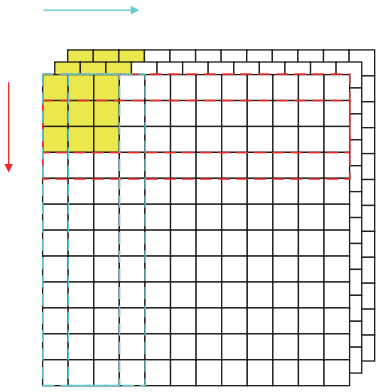


图 5 卷积窗口在特征图上滑动产生的复用(黄色部分代表卷积窗口)

Fig. 5 Multiplexing generated by sliding the convolution window on the feature map (The yellow part represents the convolution window)

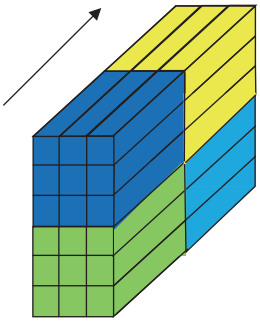


图 6 权重数据的循环分块(不同颜色的部分代表各分块权重)

Fig. 6 Cyclic block of weight data (Parts of different colors represent each weight block)

算完毕, 如果没有, 则更新下一 32 个输出通道的卷积权重, 并重复上述步骤, 直至完成所有输出通道, 卷积计算结束。YOLOv5s 的循环计算顺序如图 7 所示。

1.3 池化模块设计

YOLOv5s 网络模型中并没有使用传统的池化层进行特征图的下采样, 其只在 SPP 结构中使用了步长均为 1, 尺寸分别为 5×5, 9×9 和 13×13 的最大值池化层。由于这三种最大值池化层的池化窗口尺寸都比较大, 且步长为 1, 如果直接使用比较器阵列求取每个池化窗口的最大值, 不仅需要大量的时间开销, 而且数据复用度很低, 会造成较大的资源浪费。考虑到上述情况, 为降低时间开销和减少资源浪费, 针对上述三种较大尺寸的池化窗口, 选用多级小尺寸池化窗口串联代替大尺寸池化窗口的方式进行优化, 以 5×5 池化窗口为例, 具体优化设计如图 8 所示。

从图 8 可以直观看出, 使用一个步长为 1、尺寸为 3×3 的池化窗口对一个 5×5 的特征图进行最大值池化时, 每次池化均会得到一个当前窗口的最大特征点, 当遍历完整张特征图时, 会得到一个新的 3×3 特征图, 再使用 3×3 的池化窗口对该特征图进行最大值池化一次, 即可得到原 5×5 特征图的最大值, 等价于直接使用一个 5×5 的池化窗口对 5×5 的特征图直接操作取最大值。

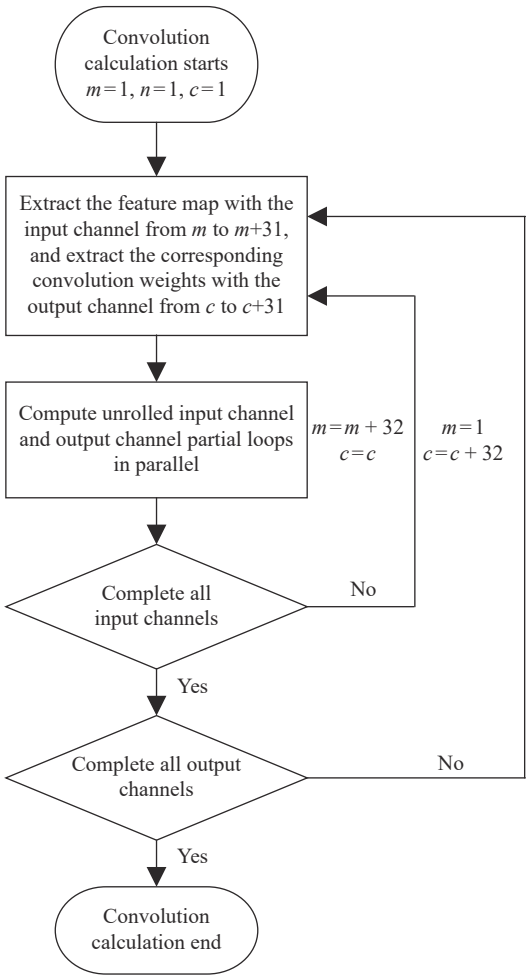


图 7 循环计算顺序

Fig. 7 Cycle calculation sequence

从图 8 还可以看出, 一个 2 级 3×3 的池化窗口可以代替一个 5×5 的池化窗口进行运算。使用 5×5 池化窗口进行最大值池化时, 采用流水线的工作方式, 一个池化窗口需要使用 24 个比较器才能每经过 1 个时钟输出一个窗口的最大值; 而在使用 2 级 3×3 池化窗口实现最大值池化时, 同样是采用流水线的工作方式, 只需要使用 16 个比较器即可在 1 个时钟内完成上述相同的工作。同理, 对于 9×9 和 13×13 的池化窗口可以分别使用 4 级和 6 级 3×3 的池化窗口级联进行替代, 此时 9×9 池化窗口所需的比较器数量从 80 个降低至 32 个, 13×13 池化窗口所需的比较器数量从 168 个降低至 48 个。这种采用多级 3×3 池化窗口代替多种较大尺寸池化窗口的优化设计方法, 不仅能够在保证计算结果不变的前提下, 有效减少计算资源的开销, 而且可以将多种尺寸的池化窗口归一化至 3×3 一种尺寸, 从而极大地降低了设计的复杂度和工程实现的工作量。

对于 3×3 池化模块的设计, 可以使用流水线的方式将其分解成多级小模块进行实现, 如图 9 所示, 池化模块实现方式如下: 开辟 3 行首尾相连的缓存队列, 每行队列缓存输入特征图完整的一行数据; 当输入特征图数据缓存完毕后, 将队列中的数据依次填充进 3×3 池化窗口中, 填充完成后选择窗口中前两列的

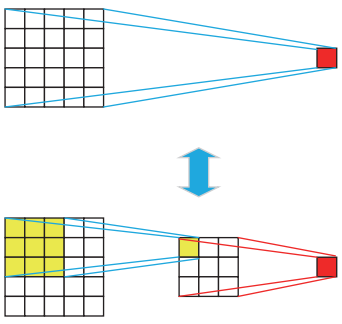


图 8 5×5 池化窗口多级串联优化设计

Fig. 8 Multi-stage series optimization design of 5×5 pooling window

数据进行比较，并将每行得到的较大值和窗口第三列的数据传输给下一级小模块进行比较，输出得到尺寸为 1×3 的数据列；同样将其传输给下一级小模块进行比较，得到 1×2 的数据列；将其传输给最后一级小模块比较输出，即可得到原 3×3 池化窗口的最大值。而且由于该池化模块是流水线的设计，当第二级小模块工作时，第一级小模块可以同时处理新的 3×3 池化窗口的数据，有效降低了时间开销。

1.4 切片模块设计

Focus 组件主要由切片操作和卷积操作构成，先将尺寸为 3×640×640 的输入特征图进行切片，得到尺寸为 12×320×320 的特征图，再使用 32 个卷积核进行卷积操作，得到尺寸为 32×320×320 的特征图。其中切片操作是一种类似 2 倍下采样的操作，且不会丢失特征图信息，切片操作如图 10 所示。

根据切片操作的方法和硬件数字电路以数据流的形式传输数据的特点，切片模块的设计如下：假设单张输入特征图的尺寸为 $N\times N$，当特征图数据以数据流的形式进行输入时，每输入一个有效数据，计数加 1：当计数小于等于 N 时，说明这是输入特征图的第奇数行数据，当计数为奇数时，将该计数的有效数据从第一通道输出，当计数为偶数时，将该计数的有效数据从第二通道输出；当计数大于等于 N 且小于等于 2 N 时，说明这是输入特征图的第偶数行数据，当计数为奇数时，将该计数的有效数据从第三通道输出，当计数为偶数时，将该计数的有效数据从第 4 通道输出；当计数大于 2 N 时，将计数器清零，再次输入就又是特征图奇数行的数据，重复上述操作步骤，直至单张输入特征图输入完成，即可得到切片后的无损 2 倍下采样特征图。切片操作流程如图 11 所示。

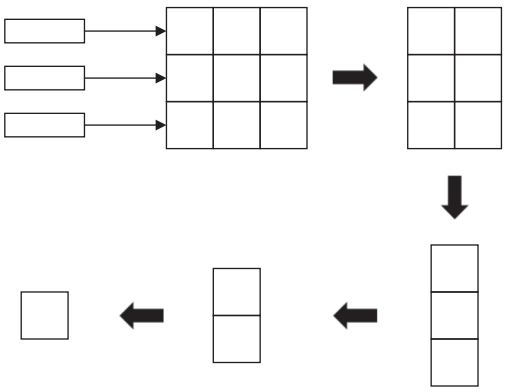


图 9 池化模块实现方式

Fig. 9 Working method of the pooling module

切片操作采用上述方法进行实现，主要利用了 FPGA 流式传输数据的特点，切片操作是针对输入特征图进行一种类似 2 倍下采样的处理，并没有对特征图数据进行实际的计算，当数据流进来时只需要将其按照特定的顺序进行分开输出即可完成，当特征图输入完成时也即完成了对其的切片操作，整个处理过程十分简洁高效。

1.5 残差组件的实现

YOLOv5s 网络模型在 CSP 结构中引入了残差组件，残差结构如图 12 所示，由两个卷积层和一个向量加法器构成，因此残差组件是在卷积计算模块的基础上实现的。残差计算模块由输入数据缓冲器、卷积计算模块、加法器模块和内存控制器模块组成，其中输入特征图三队列缓冲用于缓存卷积计算模块的输入特征图数据，残差模块特征图缓冲器也缓存相同的数据且保持不变，当三队列缓冲器的特征图数据完成两级卷积计算后，将其输出数据与缓冲器 2 的数据同时流向向量加法器，完成残差计算，其中由内存控制器模块负责实现数据缓冲器的输入与输出。

2 实验结果

2.1 实验环境设置

实验使用 Xilinx Vivado 2020.2 集成设计环境进行硬件平台开发，硬件编程语言为 Verilog HDL，通过专用的仿真软件 ModelSim 2019 进行功能仿真，并使用 Xilinx 公司 Virtex-7 系列的 VC709 连接套件作为硬件平台，其搭载一型号为 XC7 VX690 T 的 FPGA 芯片，且该套件最高可以输出 200 MHz 的时钟频率，拥有 693120 个逻辑单元、52920 kbit 的 BRAM 和 3600 个 DSP 等片上资源。

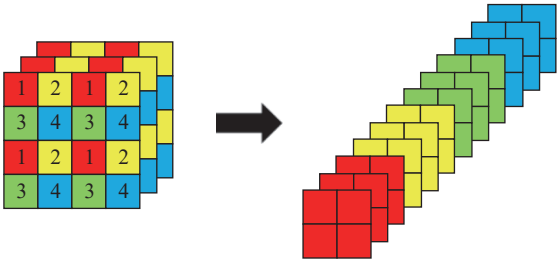


图 10 切片操作

Fig. 10 Slice operation

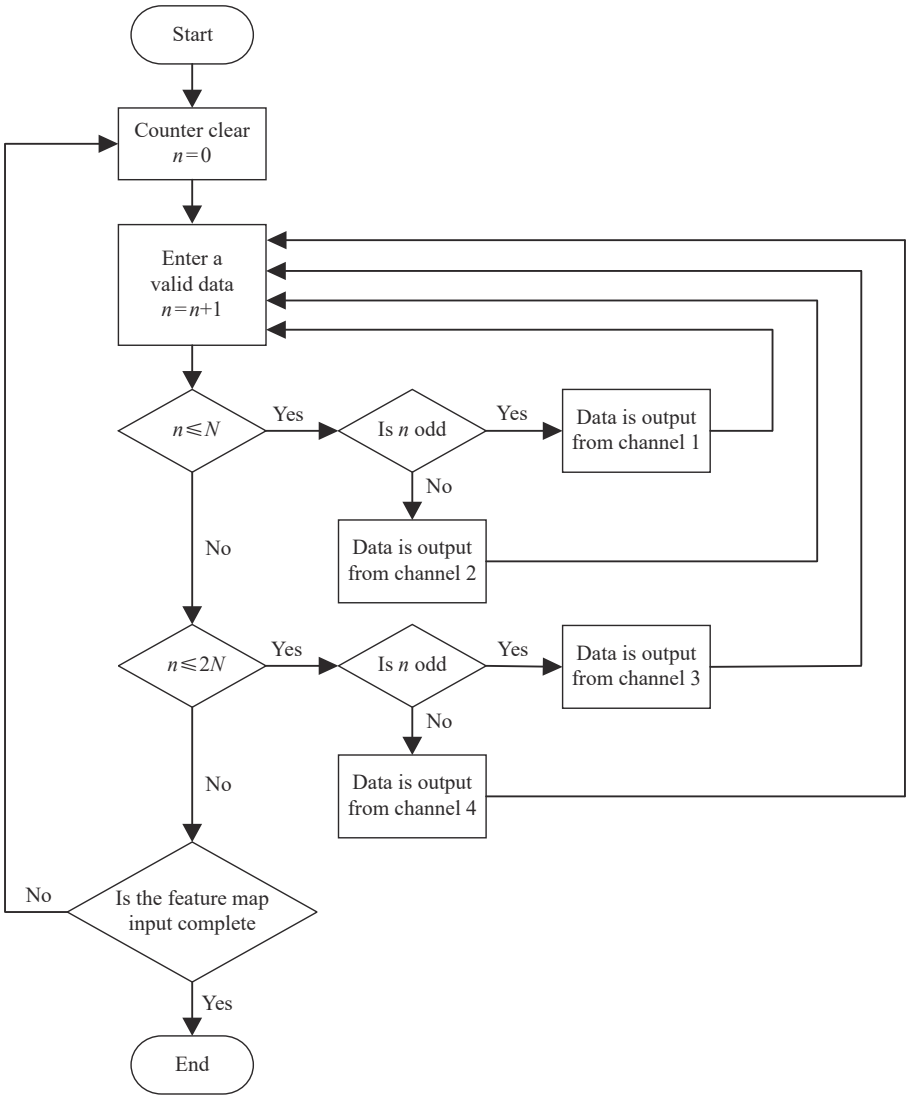


图 11　切片操作实现流程

Fig. 11　Implementation flow chart of the slicing operation

2.2　实验结果与分析

实验采用的数据精度为 16 bit 定点，并将 FPGA 运行的时钟频率设置为 200 MHz。表 1 给出的是将本文实验结果与其他文献进行对比的情况，其中不同文献的仿真环境版本略有差异，但仿真结果能够反映 FPGA 的真实运行情况，仿真环境版本的不同对实验结果并无实际影响。从表中可以看出，相较于文献 [17] 和 [18]，本文在吞吐率和能耗比方面都有着碾压性的优势；而对于文献 [16] 和 [19]，之所以在吞吐率方面本文结果略低，是因为前两者在数据精度方面分别使用了 8 bit 量化和二值化网络，而本文采用的是 16 bit 量化，若是采用与之相同的数据精度，则在吞吐率和能耗比方面将会有更好的表现；此外，在 DSP 计算效率方面，文献 [16] 略低于本文结果。

根据文献 [26]，引入 E_{DSP} 用于评估 DSP 的计算效率，其计算方法如下：

$$E_{\mathrm{DSP}} = P_{\mathrm{DSP}}/(\beta N_{\mathrm{DSP}} f_{\mathrm{FPGA}}). \tag{4}$$

其中，P_{DSP} 为 DSP 矩阵的实际计算性能（394.4 GOPS）；β 为单个 DSP 在一个时钟周期内所能处理的操作数，当数据精度为 16 bit 时 β=2；N_{DSP} 为 DSP 矩阵中 DSP 的数量，取值 1024；f_{FPGA} 为 FPGA 工作频率，取值 200 MHz。

根据前文对带宽需求的理论分析，各卷积层的 DSP 计算效率受到卷积核尺寸和输出通道数大小的影响，通过实际的 Vivado 仿真波形测试得知，不同卷积层的 DSP 计算效率分类如下：对于卷积核尺寸为

1×1、输出通道数为 32 及以下的卷积层, DSP 计算效率约为 55.3%, 占据 7 层; 对于卷积核尺寸为 1×1、输出通道数为 64 的卷积层, DSP 计算效率约为 83.5%, 占据 11 层; 而对于输出通道数为 128 及以上、任意卷积核尺寸的卷积层, 以及卷积核尺寸为 3×3、任意输出通道数的卷积层, DSP 计算效率几乎均可以达到 99% 左右, 占据 52 层。以某一计算效率达到 99% 的卷积层为例, 该卷积层的卷积核尺寸为 128×64×3×3、输入特征图大小为 64×160×160, 步长为 1, FPGA 的工作频率为 200 MHz, 理想计算时间为 9.216 ms, 而实际自特征图从 DDR 加载开始, 直至本层所有特征图卷积计算结束, 总共用时 9.23948 ms, DSP 计算效率高达 99.7%。通过综合计算, 对于整个包含 70 层卷积层的 YOLOv5s 网络来说, 平均 DSP 计算效率高达 96.29%。

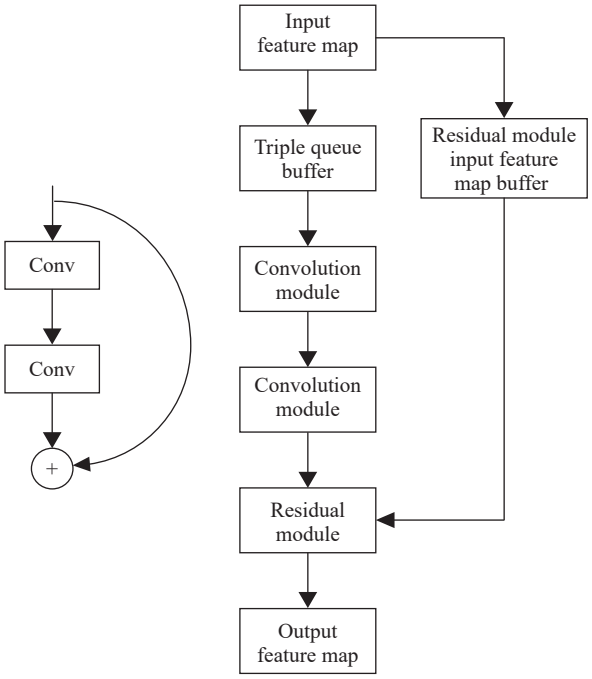


图 12 残差结构及数据流向

Fig. 12 Residual structure and data flow

针对部分卷积层的 DSP 计算效率较低的情况, 根本原因是特征图读取的带宽需求超出了单片 DDR3 存储器所能提供的带宽速率, 可以通过增加存储器的数量以满足所有卷积层的特征图读取带宽需求, 从而实现所有卷积层的计算效率均达到 99%。但是对于星上平台来说, DDR3 存储器成本很高, 而 DSP 计算效率较低的卷积层占比较少, 目前整个网络的平均 DSP 计算效率已高达 96.29%, 通过增加存储器数量所获得的加速器性能提升并不明显, 因此综合考虑来看, 本文设计的硬件加速器在性能与成本方面均有着较好的表现。

3 结论

随着中国遥感工程技术的进步, 所获取的遥感图像具有高分辨率和背景复杂的特点, 为提高对其目标检测的精确性和鲁棒性, 基于卷积神经网络的遥感图像处理算法被应用于在轨实时目标检测任务。但是

表 1 不同的 CNN 在 FPGA 上实现的情况比较

Table 1 Comparison of different CNN implementations on FPGA

Method	文献[16]	文献[17]	文献[18]	文献[19]	本文
FPGA	ZC709	XC7 K325 T	ZCU102	VC707	VC709
Network	YOLOv2-tiny	YOLOv2	YOLOv2	YOLOv2-tiny	YOLOv5s
精度/bit	8	32	16	BNN	16
频率/MHz	200	100	300	200	200
DSP	610/900	—	609	168/2800	1024/1200
BRAM	256/545	—	491	1026/1030	1094/1470
LUT	84000/219000	—	95000	86000/304000	166000/433000
FF	65000/437000	—	90000	60000/607000	228000/866000
吞吐量/GOPS	464.5	6.222	102.2	464.7	394.4
功耗/W	10.25	2.555	11.8	8.72	14.662
能耗比/(GOPS·W^{-1})	45.3	2.435	8.66	53.29	26.90
E_{DSP}	95.2%	—	—	—	96.29%

星上平台的空间与资源十分有限, 难以将模型复杂、计算量庞大的卷积神经网络算法部署在星上系统高效运行。针对这种情况, 本文选用 YOLOv5s 网络作为目标算法, 并基于宇航级 FPGA 设计了一种卷积神经网络前向推理硬件加速架构。

通过对 YOLOv5s 网络的结构进行分析, 基于 FPGA 设计了一种卷积神经网络实时处理硬件加速架构, 该架构以卷积加速模块为核心, 包括池化模块、切片模块以及残差模块等部分。其中, 针对网络中 5×5, 9×9 和 13×13 三种大尺寸池化模块, 提出一种使用 3×3 池化窗口级联代替上述三种大尺寸池化窗口的轻量化设计, 使得这三种池化模块分别节省了 33.3%, 60% 和 71.4% 的比较器资源。最后对该架构进行实现与仿真测试, 结果表明在 200 MHz 的 FPGA 工作频率下, 该架构的数据吞吐量为 394.4 GOPS, 功耗仅为 14.662 W, 说明在资源和功耗受限的星上平台中使用 FPGA 对卷积神经网络进行硬件加速具有显著优势。

针对星上卷积神经网络遥感图像目标检测算法难以部署的问题, 首次提出将基于 FPGA 的 YOLOv5s 网络部署在星上系统, 并在 FPGA 片上资源允许的范围内, 搭建出最大规模的适用于 YOLOv5s 网络计算的硬件加速架构, 用于加速网络的前向推理计算。针对 YOLOv5s 网络的模型参数进行分析, 得出其卷积层在输入通道和输出通道方向均存在高度的并行性, 且二者的展开并行度均为 32, 由此沿着这两个方向做并行展开计算, 提出一种并行度为 1024 的高性能 DSP 计算矩阵, 使得整个网络最终获得的平均 DSP 计算效率高达 96.29%。

参考文献

[1] ZAIDI S S A, ANSARI M S, ASLAM A, *et al*. A survey of modern deep learning based object detection models[J]. *Digital Signal Processing*, 2022, **126**: 103514

[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. *Communications of the ACM*, 2017, **60**(6): 84-90

[3] HUANG M Y, XU Y, QIAN L X, *et al*. A bridge neural network-based optical-SAR image joint intelligent interpretation framework[J]. *Space: Science & Technology*, 2021, **2021**: 9841456

[4] KONG T, SUN F C, LIU H P, *et al*. FoveaBox: Beyound anchor-based object detection[J]. *IEEE Transactions on Image Processing*, 2020, **29**: 7389-7398

[5] XU Y C, FU M T, WANG Q M, *et al*. Gliding vertex on the horizontal bounding box for multi-oriented object detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, **43**(4): 1452-1459

[6] LIU Z, CAI Y F, WANG H, *et al*. Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2022, **23**(7): 6640-6653

[7] SINGH B, LI H D, SHARMA A, *et al*. R-FCN-3000 at 30 fps: Decoupling detection and classification[C]//Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 1081-1090

[8] HE K M, GKIOXARI G, DOLLÁR P, *et al*. Mask R-CNN[C]//Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017: 2980-2988

[9] LIU W, ANGUELOV D, ERHAN D, *et al*. SSD: Single shot multiBox detector[C]//14 th European Conference on Computer Vision. Amsterdam: Springer, 2016: 21-37

[10] LI Z G, WANG J T. An improved algorithm for deep learning YOLO network based on Xilinx ZYNQ FPGA [C]//2020 International Conference on Culture-oriented Science & Technology (ICCST). Beijing: IEEE, 2020: 447-451

[11] WANG Z X, XU K, WU S X, *et al*. Sparse-YOLO: Hardware/software Co-design of An FPGA accelerator for YOLOv2[J]. *IEEE Access*, 2020, **8**: 116569-116585

[12] WANG J, GU S S. FPGA implementation of object detection accelerator based on Vitis-AI[C]//2021 11 th International Conference on Information Science and Technology (ICIST). Chengdu: IEEE, 2021: 571-577

[13] CAI Y F, LUAN T Y, GAO H B, *et al*. YOLOv4-5 D: An effective and efficient object detector for autonomous driving[J]. *IEEE Transactions on Instrumentation and Measurement*, 2021, **70**: 4503613

[14] TAN Xiandong, PENG Hui. Improved YOLOv5 ship target detection in SAR image[J]. *Computer Engineering and Applications*, 2022, **58**(4): 247-254 (谭显东, 彭辉. 改进YOLOv5的SAR图像舰船目标检测[J]. 计算机工程与应用, 2022, **58**(4): 247-254)

[15] GAREA A S, HERAS D B, ARGÜELLO F. Caffe CNN-based classification of hyperspectral images on GPU[J]. *The Journal of Supercomputing*, 2019, **75**(3): 1065-1077

[16] ZHANG J M, CHENG L F, LI C F, *et al*. A low-latency FPGA implementation for real-time object detection [C]//2021 IEEE International Symposium on Circuits and Systems (ISCAS). Daegu: IEEE, 2021: 1-5

[17] BI F H, YANG J. Target detection system design and FPGA implementation based on YOLO v2 algorithm[C]// 2019 3 rd International Conference on Imaging, Signal Processing and Communication (ICISPC). Singapore: IEEE, 2019: 10-14

[18] ZHANG S G, CAO J, ZHANG Q, *et al*. An FPGA-based reconfigurable CNN accelerator for YOLO[C]//2020 IEEE 3 rd International Conference on Electronics Technology (ICET). Chengdu: IEEE, 2020: 74-78

[19] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, 2019, **27**(8): 1861-1873

[20] CHEN Haomin, YAO Senjing, XI Yu, *et al*. Design and FPGA implementation of YOLOv3-tiny hardware acceleration[J]. *Computer Engineering and Science*, 2021, **43**(12): 2139-2149 (陈浩敏, 姚森敬, 席禹, 等. YOLOv3-tiny的硬件加速设计及FPGA实现[J]. 计算机工程与科学, 2021, **43**(12): 2139-2149)

[21] ZHOU Qikai, ZHANG Wei, LI Dongjin, *et al*. Ship classification and detection method for optical remote sensing images based on improved YOLOv5s[J]. *Laser and Optoelectronics Progress*, 2022, **59**(16): 1628008 (周旗开, 张伟, 李东锦, 等. 基于改进YOLOv5s的光学遥感图像舰船分类检测方法[J]. 激光与光电子学进展, 2022, **59**(16): 1628008)

[22] ZHOU Hai, HOU Qingyu, BIAN Chunjiang, *et al*. An infrared small target detection network under various complex backgrounds realized on FPGA[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2023, **49**(2): 295-310 (周海, 侯晴宇, 卞春江, 等. 一种FPGA实现的复杂背景红外小目标检测网络[J]. 北京航空航天大学学报, 2023, **49**(2): 295-310)

[23] DODD P E, SHANEYFELT M R, SCHWANK J R, *et al*. Current and future challenges in radiation effects on CMOS electronics[J]. *IEEE Transactions on Nuclear Science*, 2010, **57**(4): 1747-1763

[24] BINDER D, SMITH E C, HOLMAN A B. Satellite anomalies from galactic cosmic rays[J]. *IEEE Transactions on Nuclear Science*, 1975, **22**(6): 2675-2680

[25] HU Kongyang, HU Haisheng, LIU Xiaoming. The application of TMR on the high performance and anti radiation DSP[J]. *Microelectronics & Computer*, 2019, **36**(3): 58-60 (胡孔阳, 胡海生, 刘小明. 三模冗余在高性能抗辐射DSP中的应用[J]. 微电子学与计算机, 2019, **36**(3): 58-60)

[26] ZHANG X F, WANG J S, ZHU C, *et al*. DNNBuilder: An automated tool for building high-performance DNN hardware accelerators for FPGAs[C]//2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Diego: IEEE, 2018: 1-8