文章编号：1671-0576(2020)03-0011-07

面向舰船检测的神经网络加速器设计

肖 奇[1]， 程利甫[2]， 蒋仁兴[2]， 柳宜川[2]， 王 琴[1]

(1. 上海交通大学电子信息与电气工程学院，上海 200240；
2. 上海航天电子技术研究所，上海 201109)

摘 要：针对卫星遥感图像的舰船目标检测需求，设计了基于现场可编程门阵列(Field Programmable Gate Array，FPGA)的卷积神经网络(Convolution Neural Network，CNN)加速器。运算单元采用多层次并行化结构，底层采用乘法器级并行结构，使用行缓存单元优化数据流；顶层采用模块级并行结构，可灵活调整输出通道的并行度。针对片外数据访问延时高的问题，提出了基于 FPGA 块随机存储器(Block Random Access Memory，BRAM)的阵列式片上数据缓存单元，保证数据的实时读取和数据流的灵活分配。实验结果表明：加速器移植到 Xilinx KC705 开发平台，工作频率达 100 MHz，平均吞吐率为 217 GOPS，能效比为 86.8 GOPS/W，对连续遥感舰船图像的检测速率可达 105 帧/秒。

关键词：舰船目标检测；卷积神经网络加速器；现场可编程门阵列；并行计算

中图分类号：TN911.73 **文献标志码**：A

DOI：10.3969/j.issn.1671-0576.2020.03.002

Design of Neural Network Accelerator for Ship Recognition

XIAO Qi， *CHENG Li-fu*， *JIANG Ren-xing*， *LIU Yi-chuan*， *WANG Qin*

(1. School of Electronic Information and Electrical Engineering, Shanghai Jiaotong University, Shanghai 200240, China;
2. Shanghai Aerospace Electronic Technology Institute, Shanghai 201109, China)

Abstract: In response to the requirements of ship target detection in satellite remote sensing images, a convolution neural network (CNN) accelerator based on field programmable gate array (FPGA) is designed. The computing unit adopts a multi-level parallel design. The bottom layer adopts multiplier-level parallelism, and the line buffer unit is used to optimize data flow. The top layer adopts module-level parallelism, which can flexibly adjust the parallelism of the output channels. Aiming at the problem of high latency of off-chip data access, an array-type on-chip data cache unit based on FPGA block random access memory (BRAM) is proposed to ensure real-time data reading and flexible distribution of data streams. Experimental results show that the accelerator is deployed on the Xilinx KC705 development platform, with working frequency of 100 MHz, average

收稿日期：2020-07-09

作者简介：肖 奇(1996—)，男，硕士研究生，主要从事神经网络加速器设计与 SoC 集成技术研究。E-mail：xiaoqi470@sjtu.edu.cn

throughput rate of 217 GOPS, energy efficiency ratio of 86.8 GOPs/W, and detection rate of up to 105 frames per second for continuous remote sensing ship images.

Key words: ship target detection; CNN accelerator; FPGA; parallel computing

0 引言

目标检测网络是卷积神经网络的一个重要分支,可以对图像中的特定目标进行检测、定位。近年来检测速度和准确度最好的目标检测网络是2018年REDMON提出的YOLOv3(You Only Look Once version 3)网络模型[1]。YOLOv3网络在军事[2]、遥感[3]、工业[4]等领域得到广泛应用,与传统目标检测方法相比更先进、更高效。

要使目标检测网络发挥出优势,必须脱离高性能电脑,使其能够在终端设备上运行。设计并实现一款高性能、低功耗的CNN加速器,是目标检测网络在终端设备高效运行的基础。FPGA具有可编程、低功耗、重构成本低的优点,基于FPGA可以快速、灵活地开发CNN加速器。

CNN加速器的设计核心在于运算单元并行度的提升和数据流的优化。文献[5]基于网络压缩的思想,对算法模型反复试验,得到了轻量化的CNN模型,避免了频繁的片外访存,降低了加速器的功耗。但网络压缩的方法会造成算法精度降低,且更换算法模型后,需重新训练。文献[6]基于经典的Winograd算法,对YOLO网络进行了循环迭代优化,在FPGA上的峰值性能可达到200GOPS,GOPS表示运算能力为每秒十亿次运算数(Giga Operations Per Second,GOPS)。但该方案通用性不高,更换应用场景后,可能需要重新优化网络算法。文献[7]基于高层次综合(High-Level Synthesis,HLS)工具实现CNN加速器,采用C语言完成代码编写,将FPGA上的嵌入式处理器作为系统主控,搭建出完整的加速器系统。但HLS工具编写的加速器对FPGA的硬件使用效率偏低,限制了加速器的性能提升。

本文面向卫星遥感图像的船舰目标检测,设计了一款基于FPGA的CNN加速器。为了保证加速器的通用性,提出了一种高效的卷积并行运算架构,支持不同计算资源下并行度的灵活调整。借鉴指令集的设计思想,加速器的基本运算模块可配置,最大限度地保证了硬件的通用性。针对CNN的图像、权重数据存储需求,设计了阵列式数据存储单元,支持不同并行度的数据输入和输出。基于高级可扩展接口(Advanced eXtensible Interface,AXI)设计了加速器的顶层接口,使加速器便于移植到片上系统(System on Chip,SoC)。最后,将CNN加速器移植到FPGA上开展性能验证,并与同类研究进行比较。

1 目标检测网络及加速器设计思路

1.1 YOLOv3-Tiny网络

YOLOv3-Tiny网络是YOLOv3网络的简化版本,识别精度略微降低,但得到了10倍以上的效率提升,其结构如图1所示。网络主体包含13个卷积层、6个池化层、3个路由层和1个融合层。路由层、融合层能够实现不同尺度图像的拼接、融合,以提升对不同尺度目标的检测效果。卷积层的运算量占网络总运算量的90%以上。因此,CNN加速器的核心功能是高效完成卷积层的加速。

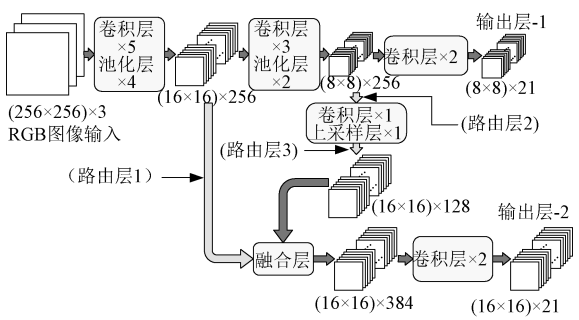


图1 YOLOv3-Tiny网络的结构

CNN的卷积运算可表示为

$$y=f\left(\sum_{i=1}^{K}\sum_{j=1}^{K}w_{(K-i+1),(K-j+1)}x_{i,j}\right) \tag{1}$$

式中:y为卷积核内所有乘积求和,再经过非线性

变换的结果；$f(\cdot)$ 为非线性激活函数；w 为卷积核的权重；x 为图像的像素点；K 为卷积核的边长；(i, j) 为卷积核窗口内的像素点坐标索引。整个卷积层包含多个通道，不同通道的卷积结果进一步求和，最后添加偏置，得到输出特征图的一个像素点。完整的卷积层结构如图 2 所示。

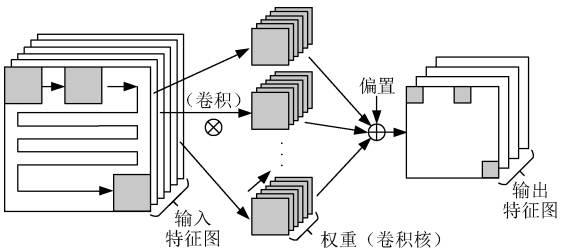


图 2　卷积层的运算过程

对于输入特征图，卷积核会按一定顺序遍历整张图片，在每个位置都做一次卷积运算，得到一个卷积结果。将卷积层不同输入特征图上相同位置产生的卷积结果进一步累加，最后再添加非线性参数（偏置），即得到输出特征图的一个像素点。该过程多次迭代，即可计算得到所有的输出特征图。

1.2　CNN 加速器的设计思路

在加速器设计上，本文将算法优化与硬件设计紧密结合。算法层面，对原版 YOLOv3-Tiny 网络进行结构调整，使网络的运算功能可以完全由硬件描述语言（Hardware Description Language，HDL）实现。硬件层面，基于 Verilog HDL 实现卷积、池化等运算的加速模块，并尽可能充分利用 FPGA 的 BRAM 资源，实现片上数据存储。由文献[8]可知，8 bits 数据能够满足 CNN 推断的精度要求。因此，本文对图像数据和网络参数（权重和偏置）均采用 8 bits 定点量化。网络整体采用逐层运算的方式，中间层的结果缓存在片上，待网络的输出层计算完毕，将结果写入片外存储器。

1.3　特殊运算的处理

YOLOv3-Tiny 的特殊运算层为路由层、融合层和图像上采样层。路由层的结果缓存在片上的 BRAM 中；融合层通过调度路由层的地址映射实现；上采样层通过额外的控制单元实现。这样，整个 CNN 加速器都能基于 FPGA 实现，使加速器保持统一的架构风格，保证运算的高效。

2　CNN 加速器设计

2.1　CNN 加速器总体架构

加速器的总体架构如图 3 所示。加速器的主体为两套同时工作的加速器并行单元，共包含 128 个卷积模块。权重缓存单元通过 AXI 接口向外部的双倍速率同步动态随机存储器（Double Data Rate Synchronous Dynamic Random Access Memory，DDRSDRAM，以下简称 DDR）读取权重，而偏置数据量较小，故全部存储在片上。片上数据缓存单元基于 BRAM 实现，缓存卷积层和路由层的运算结果（特征图数据），并为下一层的计算提供数据来源。原始图像通过 AXI 接口输入，输出层的结果通过 AXI 接口输出到 DDR。CNN 逐层运算控制单元采用状态机实现，将 CNN 每一层的运算模式进行参数化配置。一层运算启动之前，控制单元对每一级子模块的状态寄存器进行配置，并启动运算；一层运算完成后，控制单元重新配置所有子模块，并开启下一层的运算。

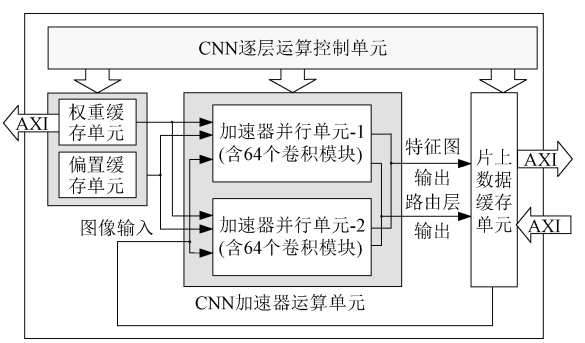


图 3　加速器的总体架构

CNN 加速器运算单元的内部并行结构如图 4 所示。

图 4 中，8 卷积并行运算单元（以下简称 8 卷积单元）为核心运算模块；通道互连单元负责改变数据流向；加法树实现不同输入通道数据的逐级累加；累加单元用来缓存、累加卷积单元输出的中间特征图，当输出通道累加完成时，将结果传递到下一级；激活单元实现非线性函数运算；池化单元采用 2×2 最大池化；中间层输出缓冲单元和路由层输出缓冲单元负责输出数据的转换，将 8 个

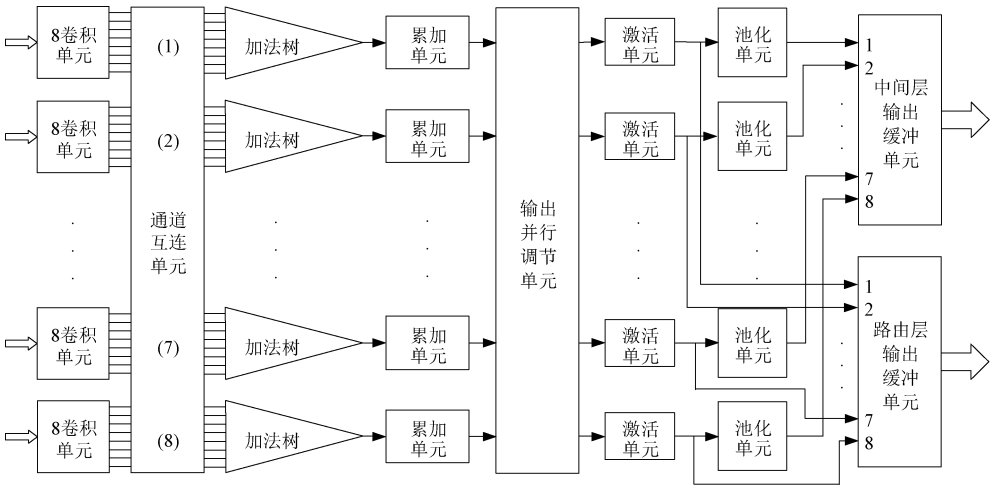


图 4 CNN 加速器并行单元架构

8 bits的数据合并为一个 64 bits 的数据，然后输出到片上数据缓存单元；输出并行调节单元用来改变输出通道的并行度。累加单元和池化单元均支持旁路，以兼容 CNN 不同层的运算模式。这一设计方案具有良好的通用性，能够很好地支持 YOLOv3-Tiny 网络每一层的运算。表 1 列举了加速器可支持的所有运算模式。

表 1 加速器支持的运算模式

卷积核尺寸	输入并行通道数	输出并行通道数
3×3	8	8
3×3	16	4
3×3	32	2
3×3	64	1
1×1	64	8

累加单元需要消耗 BRAM 资源，为了降低 BRAM 开销，将 YOLOv3-Tiny 网络的前两层设置为输出优先，输入通道完全展开，从而避免大尺寸特征图的片上累加。这样，每个累加单元仅消耗 1 块 4 kB 的 BRAM。

路由层输出的特征图将缓存到专用的路由层缓存单元(内嵌于图 3 中的片上数据缓存单元)，该单元还同时集成了路由层和上采样层的功能。

2.2 卷积单元

卷积单元的基本结构如图 5 所示，主要包括卷积时序控制单元、卷积核缓存单元、3×3 乘法运算阵列和行缓存单元。卷积核缓存单元负责向乘法单元提供权重。3×3 乘法运算阵列由 9 个乘法单元组成，每个乘法单元包含 1 个具有使能信号的乘法器和 1 个数据寄存器。行缓存单元用于实现数据的复用。卷积时序控制单元主要负责控制 9 个乘法单元的使能信号，实现卷积时序的精确控制。

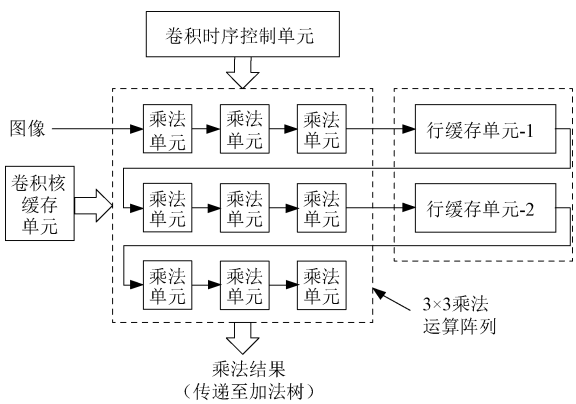


图 5 卷积单元的结构

采用行缓存的结构，可避免图像数据的重复输入，且每个周期只需输入一个像素值，极大地降低了对输入带宽的需求。当行缓存填满后，卷积运算开始，9 个乘法单元并行工作，计算一个 3×3 卷积核中的乘法，所得乘积并行输出至下一级的加法树。

在计算 1×1 卷积时，将 9 个乘法单元中的 8 个开启，每个乘法单元计算一个卷积核。需要注意，在计算 3×3 卷积时，9 个乘法单元的输出需要累加到一起，而在 1×1 卷积中，8 个乘法单元的输出不具有相关性，即每个结果对应一个独立

的输出通道。因此，在进行结果累加前，应对数据流向进行合理选择。

2.3　8 卷积单元

图 4 中的 8 卷积单元包含 8 个并行工作的卷积单元、8 个加法树和 1 个通道互连单元，其内部结构如图 6 所示。在计算 1×1 卷积时，每个卷积单元会输出 8 个输出通道的部分和。因此设计 8 个卷积单元并行工作，每个卷积单元计算不同的输入特征图，通过权重重排，使 8 个卷积单元的输出映射到 8 个输出通道，这样便可以跨单元将卷积单元的输出结果相加。

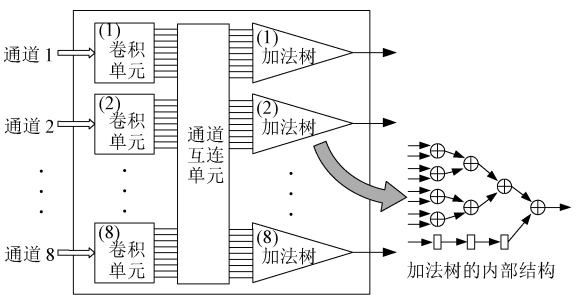


图 6　8 卷积单元的结构

图 6 和图 4 中的通道互连单元在功能上是相同的，区别只在于两侧互连的通道数目。图 4 中是 8×8 通道映射到 8×8 通道，而图 6 中是 9×8 通道映射到 9×8 通道。互连单元的输入输出关系可以表示为

$$y_{i,j}=\begin{cases}x_{i,j}, & K=3\\ x_{j,i}, & K=1\end{cases} \tag{2}$$

式中：y 表示通道互连单元的输出；x 表示通道互连单元的输入；(i,j)表示卷积单元和加法树的互连通道编号。以图 6 的通道互连单元为例，卷积单元 1 的输出通道编号为 $x_{11}\sim x_{19}$，卷积单元 2 的输出通道编号为 $x_{21}\sim x_{29}$，其余编号可类推，加法树输入通道编号方式与卷积单元的输出通道一致。

2.4　存储单元设计

本文充分利用 FPGA 的 BRAM 存储资源，根据加速器的运算需求，设计了多种缓存单元。下面主要对中间层缓存单元、路由层缓存单元、输出层缓存单元和权重缓存单元作详细介绍。

(1) 中间层缓存单元

中间层缓存单元是所有缓存单元中最主要、读写最频繁的单元，其硬件结构如图 7 所示。

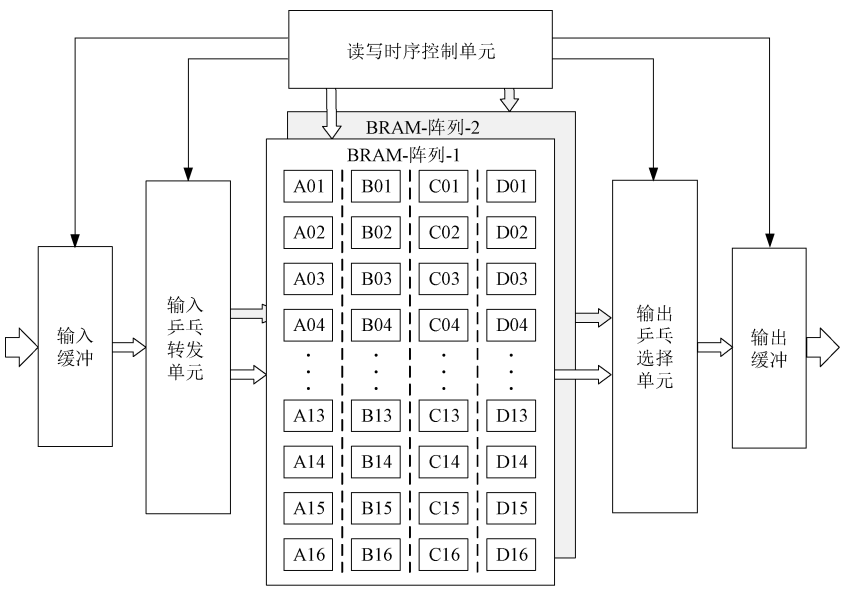


图 7　中间层缓存单元的结构

中间层缓存单元的核心部件为 BRAM 阵列，每个阵列由 64 块 BRAM 组成，将它们划分为 4 列，每列 16 块，每块 BRAM 对应加速器的一个输出通道。在数据缓存阶段，只有一列 BRAM 工作，通过控制单元调度，使 CNN 一层的输出特征图均匀存储在阵列中。在数据读出阶段，根据加速器输入并行度的需求，读取若干 BRAM 中的数据，参与后续运算。这样的阵列设计能够很好地

满足不同卷积层的输入并行度需求和输出存储需求，使每一层都能高效运算。如：计算 1×1 卷积时，加速器的输入并行度需求为 64，则控制 64 块 BRAM 同时向加速器提供数据；计算 3×3 卷积时，若输入并行度需求为 16，则由控制单元调度，从一列 BRAM(16 块)中读取数据。

中间层缓存单元整体采用了乒乓式结构，一个阵列缓存加速器的输出结果，另一个阵列向加速器提供数据，避免了数据读写的冲突。

(2) 路由层缓存单元

路由层缓存单元用于缓存 YOLOv3-Tiny 网络中 3 个路由层的输出，主要由读写控制单元和 16 块 BRAM 组成。图 1 中，路由层 2 经过卷积层和上采样层处理后得到路由层 3，然后与路由层 1 进行融合。因此，将需要进行融合的两个路由层缓存到连续的 BRAM 存储空间，稍后输出时即可完成融合。

为了降低路由层的存储开销，在 BRAM 中只存储路由层 2 经过卷积运算后的结果(8×8 大小的特征图)。在输出时，采用上采样单元实现上采样层的功能，使加速器接收到 16×16 大小的特征图。具体方法为：对原始图像的每一行，每隔两个周期，从 BRAM 中读出一个像素点，但输出端的有效信号维持两个周期，使得加速器重复接收两个相同的像素。同时，采用先入先出队列(First In First Out，FIFO)缓存上采样后的一行图像数据，原始数据的一行读取完毕后，BRAM 暂停输出，将 FIFO 中的数据发送给加速器。图 8 为上采样单元的结构。

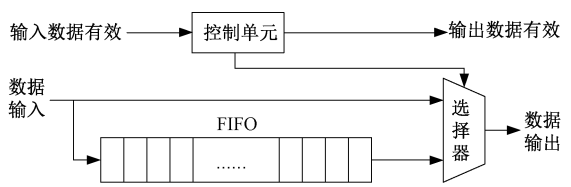


图 8 上采样单元的结构

(3) 输出层缓存单元

本文采用的面向舰船检测的 YOLOv3-Tiny 网络包含两个输出层，每个输出层有 21 个输出通道。21 个输出通道可分为 3 组，每组的 7 个通道可划分为若干个 7 维向量，即预测框。基于输出数据的特点，设计了输出层缓存单元，如图 9 所示。其中，每块 BRAM 存储一组输出通道的所有预测框，完成后，该单元会通过 AXI 接口将数据写回 DDR，交由其他单元作进一步处理。

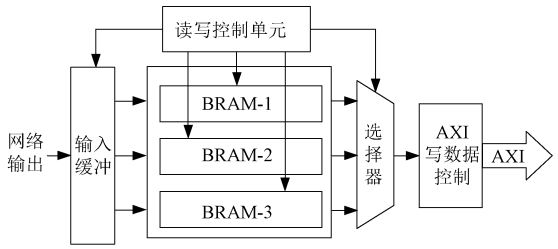


图 9 输出层缓存单元的结构

(4) 权重缓存单元

在 CNN 推断过程中，参与加速器运算的权重需要不断更新。引入权重缓存单元，可以使加速器在运算过程中及时更新权重。权重缓存单元的结构如图 10 所示。

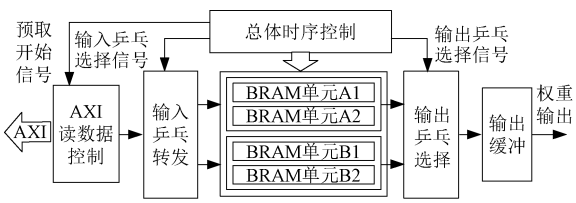


图 10 权重缓存单元的结构

权重缓存单元同样采用了乒乓式结构，一块缓存向加速器输出权重，另一块将 DDR 预取来的权重保存下来。一块缓存中的权重读取完毕后，两块缓存的功能交换。同时，权重缓存单元通过 AXI 接口向 DDR 发送读请求，读入下一批权重。读取权重采用 AXI 协议的猝发读写功能，一次读入 45 kb 的权重数据。每当加速器完成一张特征图的运算，权重缓存单元就将新的一批权重发送给加速器，保证加速器有持续的权重来源。

在算法训练完成后，需要确定 CNN 在加速器上的映射方案，并将训练好的权重进行重排序。这样，权重缓存单元只需按照既定的程序工作，每次预取固定批次的权重，便可保证权重与加速器的计算时序相匹配。

3 实验验证

3.1 实验环境

本文基于 FPGA 搭建了卷积神经网络加速器的实验系统，将加速器挂载到 AXI 总线，并集

成主控处理器、直接存储器访问(Direct Memory Access，DMA)单元、DDR 控制器、闪存等模块。系统结构如图 11 所示。启动阶段，主控处理器调用 DMA，将闪存中的图像、权重等数据上载到 DDR。然后，将第一张舰船图像输入到加速器中，同时加速器预取第一批权重和偏置。之后，主控处理器启动加速器，加速器开始工作。加速器输出结果的处理由电脑完成。

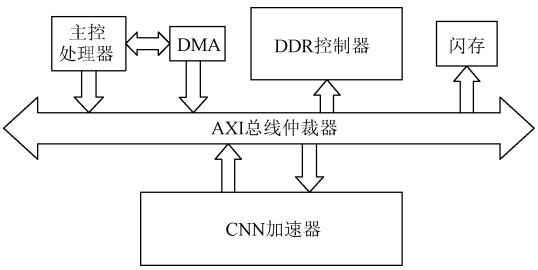
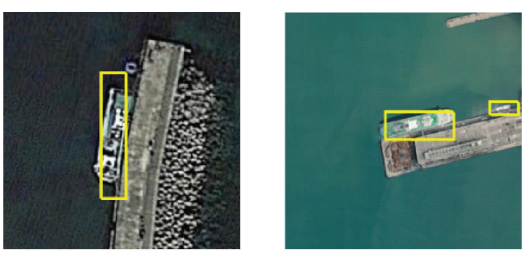


图 11　CNN 加速器实验系统的结构

3.2　实验结果

将 3.1 节设计的实验系统移植到 Xilinx KC705 FPGA 平台，经过软件综合，加速器可稳定工作在 100 MHz 时钟下，功耗为 2.5 W，平均吞吐率达到 217 GOPS。

采用包含舰船信息的卫星遥感图像对加速器进行测试。结果表明，遥感图像中的舰船目标能够被正确识别，如图 12 所示。



(a) 测试样例1　　(b) 测试样例2

图 12　加速器进行舰船检测的结果展示

经过测试，加速器推断一张尺寸为 256×256 的舰船遥感图像，耗时 9.50 ms，对于连续遥感图像的检测速率可达 105 帧/秒。

将加速器的性能指标与同类研究进行对比，评估加速器的性能，结果如表 2 所示。

对比可知，本文设计的神经网络加速器功耗较低，能效比较高，运算能力能够满足图像实时检测的要求。

表 2　与同类设计比较结果

对比指标	本文	文献[5]	文献[6]	文献[7]
硬件平台	KC705	VC707	ZC706	Z-7045
时钟频率/MHz	100	200	167	140
数据位宽/bit	8	6	16	16
查找表占用数	108.5k	86.0k	—	100.0k
触发器占用数	89.1k	60.0k	—	61.0k
数字信号处理单元占用数	814	168	900	864
BRAM(4kB)占用数	275	513	545	320
功耗/W	2.5	8.7	9.4	10.0
平均吞吐率/GOPS	217	465	203	169
能效比/(GOPS/W)	86.8	53.3	21.4	16.9

4　结束语

本文面向卫星图像实时船舰目标检测的应用，设计并实现了基于 FPGA 平台的通用卷积神经网络加速器。基于目标检测网络 YOLOv3-Tiny设计了最优的卷积运算并行架构，并充分利用 FPGA 的硬件资源，设计了高效的数据存储单元和权重缓存单元，最终实现了一款基于 AXI 接口的高能效 CNN 加速器。该加速器移植到 FPGA 平台后，经测试验证，工作频率达 100 MHz，平均吞吐率达到 217 GOPS，功耗仅为 2.5 W，能效比达 86.8 GOPS/W，对连续遥感图像的检测速率达到 105 帧/秒，满足实时目标检测的需求。

参考文献

[1] REDMON J，FARHADI A. Yolov3：an incremental improvement[EB/OL]. arXiv preprint：1804.02767 (2018-04-08). https://arxiv.org/abs/1804.02767.

[2] 马旗，朱斌，张宏伟，等. 基于优化 YOLOv3 的低空无人机检测识别方法[J]. 激光与光电子学进展，2019，56(20)：279-286.

[3] 戴伟聪，金龙旭，李国宁，等. 遥感图像中飞机的改进 YOLOv3 实时检测算法[J]. 光电工程，2018，45(12)：180350.

[4] 张广世，葛广英，朱荣华，等. 基于改进 YOLOv3 网络的齿轮缺陷检测[J]. 激光与光电子学进展，2020，57(12)：121009.

(下转第 45 页)