DOI:10.3969/j.issn.1003-5060.2025.07.007

基于 Zynq 的卷积神经网络加速器设计

孟凡开[1,2], 张 峰[2], 李 森[2], 张多利[1]
(1.合肥工业大学 微电子学院,安徽 合肥 230601; 2.中国科学院自动化研究所 国家专用集成电路设计工程技术研究中心,北京 100190)

摘 要:针对卷积神经网络(convolutional neural network,CNN)嵌入式部署资源开销大、运行速度慢等问题,文章提出一种以 Tiny-YOLOv3 作为算法模型的 CNN 硬件加速器。首先,基于 Tiny-YOLOv3 网络各层的特性和要求设计 CNN 加速器实现方案,将权重系数按位分割,面向单 bit 权重设计卷积加速器,通过逐位实施达到处理速度和识别率的高效平衡;然后,采用查表选择法实现卷积算子的乘加运算,设计一款 6×3×16 的三维加速器计算阵列,可单周期完成 288 个卷积窗口计算;最后,在 Xilinx Zynq UltraScale+MPSoC 系列芯片上对设计的 CNN 加速器进行性能测试。实验结果表明,该 CNN 加速器在 200 MHz 频率下具有 518.4 GOPS 的算力,比现有的解决方案性能提高了约 63%。

关键词:卷积神经网络(CNN);Tiny-YOLOv3 网络模型;硬件加速;流水阵列;并行运算
中图分类号:TN47 **文献标志码**:A **文章编号**:1003-5060(2025)07-0904-06

Design of convolutional neural network accelerator based on Zynq

MENG Fankai[1,2], ZHANG Feng[2], LI Miao[2], ZHANG Duoli[1]
(1. School of Microelectronics, Hefei University of Technology, Hefei 230601, China; 2. National ASIC Design Engineering Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: In view of the problems of high cost and slow running of embedded deployment of convolutional neural network(CNN), a CNN hardware accelerator using Tiny-YOLOv3 as algorithm model is presented. According to the characteristics and requirements of each layer of Tiny-YOLOv3 network, a CNN accelerator implementation scheme is proposed, which divides the weight coefficients bit by bit, designs the convolution accelerator for single bit weight, and achieves a high-efficiency compromise between processing speed and recognition efficiency through bit-by-bit implementation. The multiplication and addition of convolution operator is implemented by a table-looking selection method. A 6×3×16 3D accelerator calculation array is designed, and 288 convolution windows can be calculated in a single cycle. Finally, the performance of CNN accelerator is tested on Xilinx Zynq UltraScale+MPSoC series chips. The results show that the designed CNN accelerator has a computational power of 518.4 GOPS at 200 MHz, achieving a performance improvement of about 63% compared to existing solutions.

Key words: convolutional neural network(CNN); Tiny-YOLOv3 network model; hardware acceleration; pipeline array; parallel operation

0 引 言

随着深度学习技术的发展,目标检测算法得到进一步改进,例如 YOLO(you only look once)算法将目标检测视为回归问题,实现了高检测速度和高精度的统一。文献[1]提出首个基于单个

收稿日期:2023-05-12;**修回日期**:2023-06-02
基金项目:国家自然科学基金资助项目(61874156);安徽省高校协同创新资助项目(GXXT-2019-030)
作者简介:孟凡开(1998—),男,黑龙江七台河人,合肥工业大学硕士生;
张 峰(1977—),男,山西晋城人,博士,中国科学院自动化研究所正高级工程师;
张多利(1976—),男,黑龙江七台河人,博士,合肥工业大学教授,博士生导师,通信作者,E-mail:zhangduoli@hfut.edu.cn.

神经网络的目标检测系统 YOLO；文献[2]推出 YOLOv3 版本，相较于之前版本，YOLOv3 加深了骨干网络深度，但是降低了系统速度。为弥补这一不足，YOLO 团队推出轻量化的 Tiny-YOLOv3，将 YOLOv3 的 3 个预测分支缩减为 2 个独立分支，减少了参数，提高了运行速度。因此，Tiny-YOLOv3 是目前工程领域中被使用最多的方案。

随着卷积神经网络（convolutional neural network，CNN）计算量持续增加，硬件加速已成为基本实现形式。主流硬件载体有通用图形处理器（general-purpose computing on graphics processing units，GPGPU）、专用集成电路（application specific integrated circuit，ASIC）和现场可编程门阵列（field programmable gate array，FPGA）3 类：GPGPU 有极高的并行计算能力，但功耗巨大难以应用于功耗受限的平台；ASIC 在吞吐量、延迟、功耗等方面均有优异性能，但研发成本高，缺乏灵活性；FPGA 具备灵活的算法适应性和较短的设计周期，能够快速响应不同应用的需求，可作为新型可编程的逻辑器件。Zynq 平台集成了一个双核 ARM Cortex-A9 处理器和一个传统的现场可编程门阵列逻辑部件（即 FPGA），其兼具 ARM 处理器操作灵活性和 FPGA 高并行、高吞吐、低功耗的优点，一经问世，即成为最受欢迎的 CNN 硬件载体。

近年来，多种基于 Zynq 器件的 CNN 加速器相继问世。文献[3]设计了可以扩展到不同的 CNN 网络和 FPGA 设备通用框架，验证了 Zynq 设备的吞吐量估计性能；文献[4]采用软硬件协同的方法在 Zynq 平台上实现 4 个目标检测器的评估和验证；文献[5]设计了使用 Zynq 的 FPGA 加速的 CNN 边缘智能垃圾分类系统，在 CNN 模型中采用了硬件计算模块重用策略，大幅降低了模型对 FPGA 硬件资源的需求；文献[6]使用 Zynq 实现了高实时性、高识别效率、高准确度、低成本的便携式嵌入式识别系统。

上述研究都没有深入探讨 CNN 微结构优化问题，而此类工作能够为嵌入式部署提供更优的性价比。本文主要工作如下：针对 Tiny-YOLOv3 算法模型中网络各层的特性和要求，提出将权重系数按位分割且面向单 bit 权重的卷积加速器方案，通过位计算达到处理速度和识别率的高效平衡；并设计基于查表法的卷积算子乘加器完成一款 6×3×16 的三维加速器计算阵列设计，可单周期完成 288 个卷积窗口计算。

1　CNN 加速器优化设计

本文设计的张量运算硬件加速器使用优化后的卷积运算方案，输入/输出的特征激活值张量采用 8 位无符号定点数表示，滤波器系数为 1 位二进制数，激活值与权重系数按“滑窗”形式组织卷积运算。

1.1　整体架构

CNN 卷积层运算规则为：输入和输出数据均为 8 位无符号定点数，数据组织形式为行、列、通道的三维张量。根据上述规则设计的硬件加速器结构如图 1 所示。

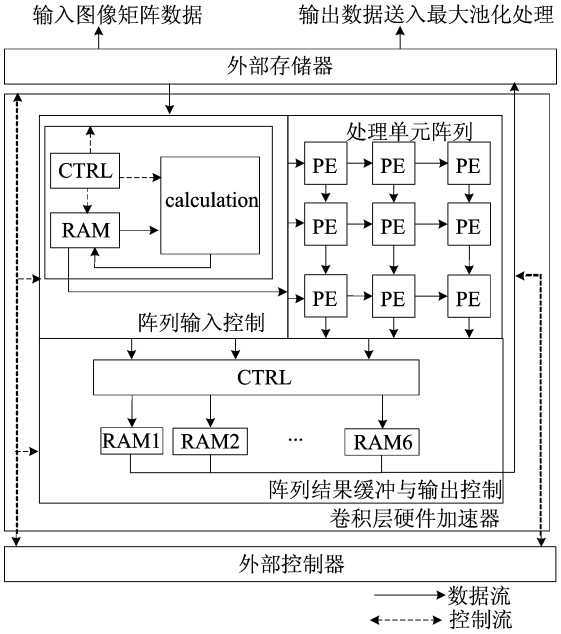

图 1　CNN 加速器全局架构图

卷积层硬件加速器由数据存储单元、特征张量运算单元和数据流控制单元 3 个部分组成，各单元功能如下：

1）数据存储单元。包括存储所有参数和特征张量数据的外部存储器；存放运算阵列数据的寄存器，含输入特征张量寄存器、权重系数寄存器和缓存运算中间变量的寄存器；存放运算结果的 RAM。

2）运算单元。利用高并行度和高重复度的规律优化卷积运算过程，使用 MUX 与进位保存加法器（carry save adder，CSA）组合替换乘累加运算，压缩运算量，并简化电路结构。

3）数据流控制单元。访问外部存储器，完成原始权重的读取和结果回写；管理计算过程中输入特征张量和权重系数的阵列载入。

1.2 卷积运算优化

本文设计的卷积核二维张量是固定的 3×3 结构,其权重系数有 29 种,本文提出结构分组、结果累加方案,实现以卷积核 3×3 结构的输入特征张量为窗口,以 1×3 结构输入特征张量为向量,窗口以步长 1 向右向下滑动,即可将 3×3 结构的权重系数按行分解为 3×1×3 结构,构造出的处理单元 PE 阵列结构如图 2 所示。

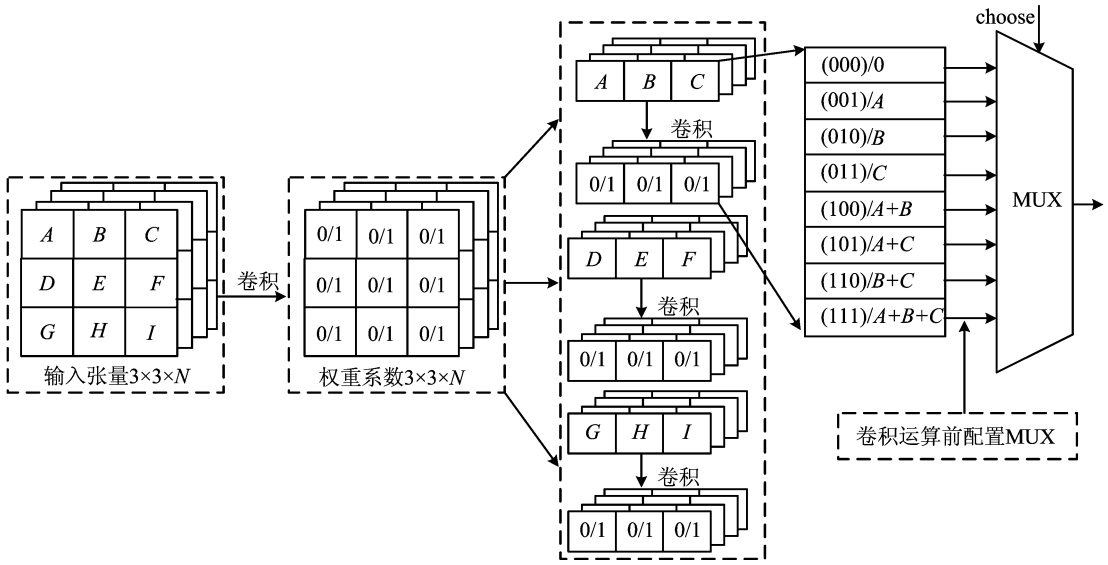

图 2 PE 阵列结构图

以图 2 中第 1 行输入张量(A,B,C)和第 1 行权重系数为例,其卷积计算的结果共有 0、A、B、C、$A+B$、$A+C$、$B+C$、$A+B+C$ 8 种,此时可以将乘法计算转换为加法和选择操作,也就是可以用1 组MUX 和加法器完成 1 次卷积运算,大幅度简化了卷积硬件计算单元的资源,提高了执行速度。

标准算法中输出特征张量需要 27 次乘法和 27 次累加计算,而本文提出的卷积优化算法只需要 9 次数据选择和 9 次累加计算,本文的每个 PE 单元以增加 24 个 10 位寄存器为代价,获得省掉全部乘法器和减少 67%加法器资源的效果。

1.3 处理单元设计

在通过选择法实现滤波器间卷积和并行计算的基础上,利用阵列级联实现张量层间求和运算,并结合一维输入向量的滑动以脉动流水方式完成张量通道平面内的扫描覆盖。由卷积运算优化算法设计的处理单元包括:

1) 基本单元(basic element,BE)。3 bit 向量卷积和查询选择单元,根据 choose 值使用一个类似 8 选 1 的 MUX 完成一个向量卷积任务的设定。

2) 归并加法部件。部分积压缩阵列使用 2 个进位保存加法器(carry save adder,CSA)和 1 个全加器实现 4 个输入数据的累加求和。4 个输入数据由前一级单元的输出与 3 个通道向量卷积和组成。处理单元内部结构及其级联方式如图 3所示。

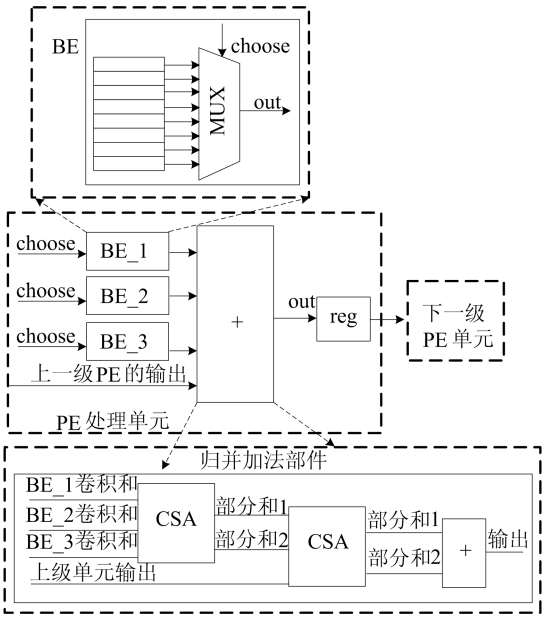

图 3 处理单元内部结构及其级联方式

1.4 处理单元阵列设计

处理单元阵列的功能是实现输入特征张量数据按列方向执行脉动计算,相邻 PE 张量输入在时序上差 1 个周期。阵列输入控制单元通过权重预配置和输入特征张量加载动态调度计算阵列流水化工作。处理单元计算阵列结构如图 4 所示。

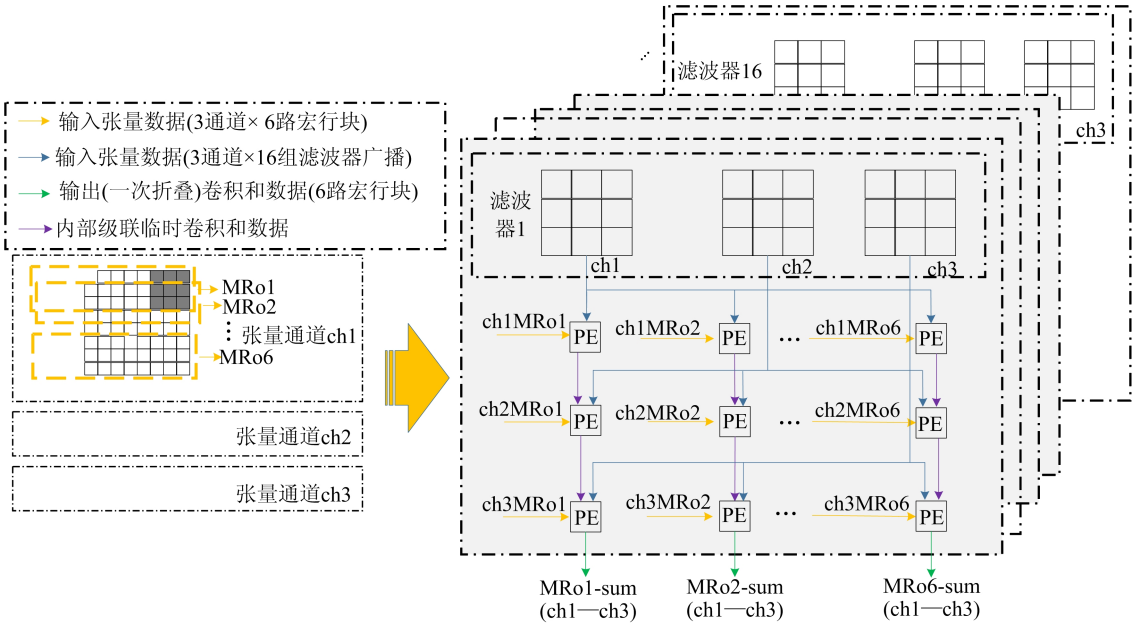

图 4　处理单元阵列结构图

本文的 CNN 加速器为一个由 288(6×3×16)个 PE 组成的立方体计算阵列结构，具体组成如下：

1) X 轴向阵列。该阵列共 6 行，根据 Tiny-YOLOv3 网络中各卷积层 1 行向量向右滑动次数，用 3×3 结构的滤波器滑动 1 行张量需 $6k$ 次($k\in[1,32]$，$k\in\mathbf{N}$)，考虑到阵列的并行度和工作效率，本文设置 6 个通道为 1 组基本操作。例如，第 8 卷积层的单通道输入张量结构为26×26，经过填充操作后，1 行向量需要滑动 26 次。将每次滑动结果按 6 次为 1 组载入计算阵列，共需载入 5 次完成 1 行向量的滑动。

2) Y 轴向阵列。因为 1 个 PE 内部包含 3 个 BE 单元，所以 1 次可以完成 3×3 张量的卷积计算，根据各卷积层的通道数，本文选取 3 列构成计算阵列结构。同样以第 8 卷积层为例，其 128 个通道按每 3 个 1 组分成 43 组，每组计算完成后将结果写回第 1 列进行累加，直至全部通道计算完毕。

3) Z 轴向阵列。该阵列共 16 层，每层表示 1 组滤波器权重系数，其卷积过程中滤波器组与组之间无数据交换。设置该 16 层结构以 16 个为 1 组，将滤波器分组操作。

处理单元阵列内按脉动形式组织数据流，处理单元阵列的输入张量与权重卷积运算示意图如图 5 所示。

由图 5 可知，从 T_1 时刻第 1 组输入张量流入计算阵列到 T_8 时刻最后一组输出张量流出，$8T$ 时间内完成了 9 次选择和 6 次累加计算。

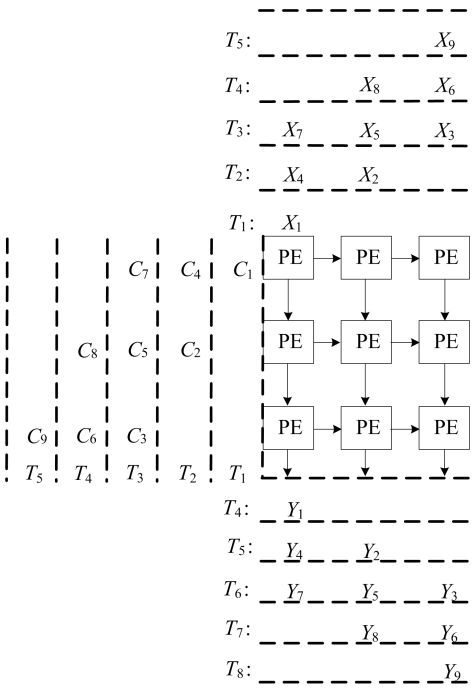

图 5　输入张量与权重卷积运算示意图

1.5　阵列数据流控制

由 1.2 节可知，每个 3×3 张量窗口需与多个滤波器的权重系数进行卷积，因此可以预先计算出 1 行输入特征张量和多个滤波器权重系数的所有结果，并将其缓存在寄存器中，再用查表代替后继运算。由 288 个 PE 构成的立方体结构脉动阵列在计算时将选择信号向右传播，输入特征张量向下传播，每一列的计算结果经过累加后最终得出输出特征张量。阵列数据流示意图如图 6 所示。

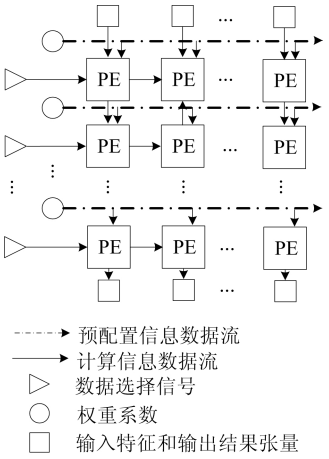

图 6 阵列数据流示意图

本文设计的处理单元阵列执行 1 次通道为 6 个特征张量与 12 个权重进行卷积运算时的阵列内权重系数配置流程如图 7 所示。

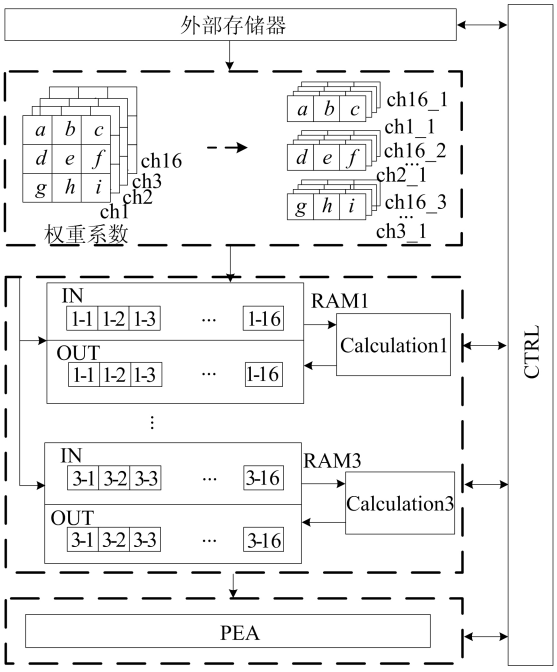

图 7 权重系数加载示意图

权重系数加载至计算阵列的过程如下:

1) 将 3×3 结构的权重系数按行分成的 3×1×3结构,按规则存储在 RAM 中。

2) 计算出 1 组权重系数,然后将其搬运回 RAM。

3) 在卷积运算前配置到相应的 PE 单元内部 MUX 的 8 个寄存器中。

2 系统性能评估

将本文 CNN 加速器部署在 Xillinx 公司的 Zynq UltraScale+MPSoC 系列 XCZU3CG 型号芯片上,对 CNN 加速器硬件设计性能和硬件资源消耗进行验证。

2.1 实验开发流程

CNN 加速器部署系统实验分别在 Vivado 2019.2 和 SDK 开发工具 Vitis 上完成硬件平台搭建和系统软件编写。使用 Verilog 语言完成模块电路设计并集成到 Zynq 开发板上,实验数据为 208×208×16 个的输入特征激活值、32×3×3×16个滤波器权重以及 208×208×32 个特征激活值。

2.2 实验结果

CNN 硬件加速器综合后的资源使用情况见表 1 所列。

表 1 优化后加速器核整体资源使用情况

硬件资源	占用资源数	总资源数	使用率/%
LUT	29 146	70 560	41.31
FF	89 274	141 120	63.26
DSP	0	360	0

本文 CNN 加速器与其他神经网络硬件加速器的性能对比见表 2 所列,可以看出加速器工作频率为 200 MHz,算力为 518.4 GOPS。

表 2 本文 CNN 加速器与其他神经网络硬件加速器的性能对比

数据类别	加速器				
	文献[7]	文献[8]	文献[9]	文献[10]	本文
实验平台	5CGXF	VX690T	VC707	XCZU3CG	XCZU3CG
CNN 模型		AlexNet	Tiny-YOLOv2	Tiny-YOLOv3	Tiny-YOLOv3
工作频率/MHz	100	150	200	250	200
数据位宽/bit	16	16	6	8	8
LUT	89 423*(79.00%)	175 000(40.00%)	86 000(28.33%)	10 793(15.30%)	29 146(41.31%)
FF		202 000(23.00%)	60 000(9.88%)	22 286(15.79%)	89 274(63.26%)
DSP 资源	780(86.70%)	1 376(38.00%)	168(6%)	0	0
算力/GOPS	317.86	570.00	464.47	48.00	518.40

注:* 资源为 ALM。

文献[10]加速器采用二维脉动流水串行的方式完成卷积运算，而本文 CNN 加速器采用三维并行的方式完成卷积运算，相较之下性能提升了 9 倍以上；本文 CNN 加速器与文献[7]加速器均采用并行流水架构，不同之处在于本文以加法器代替了全部乘法器来执行卷积运算，相较之下速度提高了 1 倍，使性能提高了 63%；与文献[8]加速器相比，本文 CNN 加速器在没有使用 DSP 资源的前提下达到了与之相近的性能；与文献[9]加速器相比，本文加速器在 DSP 资源的使用量与性能两方面均占优。

3　结　　论

本文针对卷积运算优化问题，对卷积计算单元和阵列进行优化，计算单元使用类似 8 选 1 数据选择器和 CSA 加法器代替乘累加完成卷积运算，计算阵列采用了三维并行的方式提高运算速度。在 Zynq 平台上对本文设计的 CNN 加速器整体性能进行了测试，结果表明，在未使用 DSP 资源的情况下，该加速器可以快速地进行卷积运算。

[参　考　文　献]

[1] 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. [S. l.]：IEEE，2016：779-788.

[2] REDMON J，FARHADI A. Yolov3：an incremental improvement [EB/OL]. (2019-04-08). https://arxiv.org/abs/.

[3] EETHA S，SRUTHI P K，PANT V，et al. TileNET：hardware accelerator for ternary Convolutional Neural Networks [J]. Microprocessors and Microsystems，2021，83(11)：104039.

[4] SHARMA A，SINGH V，RANI A. Implementation of CNN on Zynq based FPGA for real-time object detection[C]//2019 10th International Conference on Computing，Communication and Networking Technologies (ICCCNT). [S. l.]：IEEE，2019：1-7.

[5] WANG W，ZHU M. An efficient and low-cost FPGAs-accelerated CNN-based edge intelligent garbage classification system on Zynq[C]//2021 International Joint Conference on Neural Networks (IJCNN). [S. l.]：IEEE，2021：1-8.

[6] ZHANG X X，MA H Z，WEI S Y，et al. Design of day-lily robot recognition system based on ZYNQ[C]//2021 International Conference on Networking，Communications and Information Technology (NetCIT). Manchester：IEEE，2021：154-157.

[7] 秦华标，曹钦平. 基于 FPGA 的卷积神经网络硬件加速器设计[J]. 电子与信息学报，2019，41(11)：2599-2605.

[8] SHEN J，YOU H，WANG Z，et al. Towards a uniform template-based architecture for accelerating 2D and 3D CNNs on FPGA[C]//The 2018 ACM/SIGDA International Symposium. [S. l.]：ACM，2018：97-106.

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

[10] 胡永阳. 目标检测网络硬件加速研究与实现[D]. 合肥：合肥工业大学，2022.

（责任编辑　胡亚敏）

(上接第 903 页)

[6] PRATIKANTA M，ATANU B，MOUSAM G. FPGA-based real-time implementation of quadral-duty digital-PWM-controlled permanent magnet BLDC drive[J]. IEEE Transactions on Mechatronics，2020，25(3)：1456-1467.

[7] 肖宜辉，宋保业，许琳. 一种无刷直流电机模糊比例积分控制器的设计[J]. 科学技术与工程，2020，20(19)：7750-7755.

[8] 祝相泉，黄海龙，田昊. 无刷直流电机模糊 PID 控制[J]. 辽宁工业大学学报(自然科学版)，2020，40(1)：22-25.

[9] 刘甫，曾国辉，黄勃，等. 基于改进模糊控制的无刷直流电机控制系统[J]. 制造业自动化，2021，43(10)：64-67，118.

[10] 孙兆龙，钱翰宁，刘振田，等. 基于 ARM+FPGA 的永磁无刷直流电机控制智能方法[J]. 海军工程大学学报，2023，35(1)：93-98，105.

[11] 尹洪桥，易文俊，贾芳，等. 基于单神经元神经网络的无刷直流电机控制系统仿真[J]. 科学技术与工程，2021，21(7)：2747-2753.

[12] 宋丽君，王燕. 一种无刷直流电机模糊自适应控制方法[J]. 制造技术与机床，2022(4)：145-148.

[13] SAMAHY A A，SHAMSELDIN M A. Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control[J]. Ain Shams Engineering Journal，2018，9(3)：341-352.

[14] 韩团军. 高精度无刷直流电机模糊控制系统的研究及 FPGA 实现[J]. 现代电子技术，2018，41(9)：175-178.

[15] 葛佳航. 基于 FPGA 的直流无刷电机无位置传感器控制系统设计研究[D]. 哈尔滨：哈尔滨理工大学，2018.

[16] 卿金晖，胡黄水，王宏志. 基于 FPGA 的 BLDCM 模糊 PID 控制器设计[J]. 长春工业大学学报，2021，42(2)：168-174.

[17] 杨兴旺. 基于 FPGA 的直流电机转速控制研究与设计[D]. 长春：长春工业大学，2020.

（责任编辑　胡亚敏）