

DOI:10.19344/j.cnki.issn1671-5276.2026.04.021

边缘环境下 FPGA 目标检测系统的优化设计研究

钱淑冰,王凯,张旭峰,梁家伟,李雨蔓

(南京航空航天大学,江苏 南京 210016)

摘要:针对边缘环境对设备的功耗、实时性要求较高,且嵌入式设备的资源有限等条件,提出一种基于现场可编程门阵列(FPGA)的轻量化YOLO目标检测系统优化设计。在计算密集的IP设计中,通过并行计算、层间融合来提高推理速率,并采用参数化的虚拟重构方法来实现硬件加速过程中的数据控制调度、模块分时复用,从而最大化实现系统的吞吐量。实验结果表明:在ZYNQ-7020验证平台上的资源使用量和功耗显著降低,系统的识别准确率和实时性能良好。

关键词:边缘环境;现场可编程门阵列;轻量化YOLO目标检测系统;并行计算;控制调度参数化

中图分类号:TP183 **文献标志码:**B **文章编号:**1671-5276(2026)04-0110-06

Research on Optimized Design of FPGA-Based Object Detection System in Edge Environments

Qian Shubing, Wang Kai, Zhang Xufeng, Liang Jiawei, Li Yuman

(Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: In response to the high demands on power consumption and real-time performance in edge environments, as well as the limited resources of embedded devices, this paper proposes an optimized design of a lightweight YOLO object detection system based on Field-Programmable Gate Array (FPGA). In the computationally-intensive IP design, parallel computing and inter-layer fusion are employed to improve the inference rate. Additionally, a parameterized virtual reconfiguration method is adopted to achieve data control scheduling and module time-division multiplexing during hardware acceleration, thereby maximizing system throughput. Experimental results show that the resource utilization and power consumption on the ZYNQ-7020 validation platform are significantly reduced. The system exhibits good recognition accuracy and real-time performance.

Keywords: edge environment; Field-Programmable Gate Array (FPGA); lightweight YOLO object detection system; parallel computing; parameterized control scheduling

0 引言

近年来,边缘环境下的目标检测系统在许多实际应用场景中发挥着重要作用,例如自动驾驶、智能家居、智能监控等,它为社会生活带来了更多的便利与安全。目标检测算法中,YOLO^[1]有物体识别、定位的作用,比R-CNN^[2]的检测速度更快、通用性更强,准确率也并不逊色。目前,YOLO结合新兴的网络设计亮点和自身的需求,不断提高其检测的准确性和速度,如YOLOv3^[3]、YOLOv4^[4]、PPYOLO^[5]等。而考虑边缘计算的应用需求,许多研发者提出的轻量化YOLO网络的准确度损失非常低^[6-8]。因此,以轻量化YOLO为探索对象对边缘端应用开发具有重要的意义。

现场可编程门阵列(field-programmable gate array, FPGA)是一种高性能且低功耗的可自定义

芯片,具有极高的可编程性,因此FPGA常被作为加速器件^[9-22]。文献[23]使用GEMM优化卷积计算和收缩阵列结构,减小设计尺寸;文献[24-26]使用高级综合工具(high-level synthesis, HLS)实现。但系统难以实现最佳性能且功耗难以保证。对此,本文从边缘环境出发,使用底层的硬件描述语言,设计了一个针对轻量化YOLO的低功耗、高性能FPGA目标检测系统架构,以及参数化的加速单元和分时复用方案,并与现有的相关设计进行了资源消耗和性能的比较。

1 方法论述

YOLO是直接把整个图片分割成多个预测框,每个预测框都要逐步过滤,最终留下各预测框中的目标置信度、不同类别概率及其以预测框中心为依据的相对坐标和以整个图像为依据的相对

第一作者简介:钱淑冰(1998—),女,江苏无锡人,硕士研究生,研究方向为智能仪器及自动化技术,2766389169@qq.com。

高度与宽度。另外,为了满足物体尺寸不同的情况,YOLO 发展了多尺度预测的设计,通过特征金字塔网络 (feature pyramid network, FPN) 结构 (图 1),保留了更丰富的信息,并输出不同分割情况的预测结果。因此,YOLO 的层间特征矩阵的尺寸变化较多。

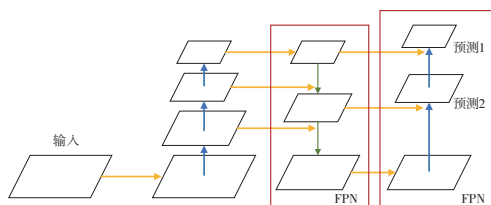


图 1 特征金字塔网络结构

YOLO 系列推理网络主要包括卷积层、归一化层、激活函数层、池化层、上采样层等。卷积层、归一化层和激活函数层往往会作为一个联合设计,激活函数主要使用 LeakyReLU。池化层以最大池化为主,上采样层有复制式和补零式,如图 2 所示。另外,YOLO 还有后处理来对网络输出结果进行整合,主要包括坐标和大小的映射处理、置信度阈值比较和各种类型置信度的非极大值抑制 (NMS) 筛选。

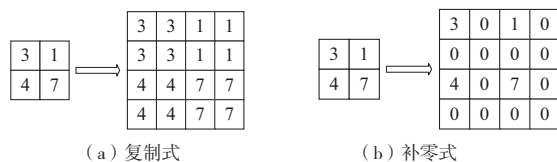


图 2 上采样处理示意图

2 硬件加速器

2.1 系统架构

本文设计的轻量化 YOLO 目标检测系统的框架如图 3 所示。整个系统分为 3 个部分,ARM 主导的控制端,FPGA 主导的加速端,AXI 总线主导的通信部分。处理系统 (processing system, PS) 端与外设沟通,初始化目标检测系统,采集图像信息并整理输入像素点,等待 FPGA 加速结束后,进一步处理加速结果获得目标检测信息最终显示到检测屏上。可编程逻辑 (programmable logic, PL) 端主要实现网络的加速推理过程,采用虚拟重构的方法使硬件在适应不同规模的卷积层并行处理的前提下,实现最大化吞吐量。此外,为减少与 PS 交互产生的延迟时间,PL 端自动控制 AXI 总线与片外存储器 DDR 进行数据交互,并自主实现虚拟重构。

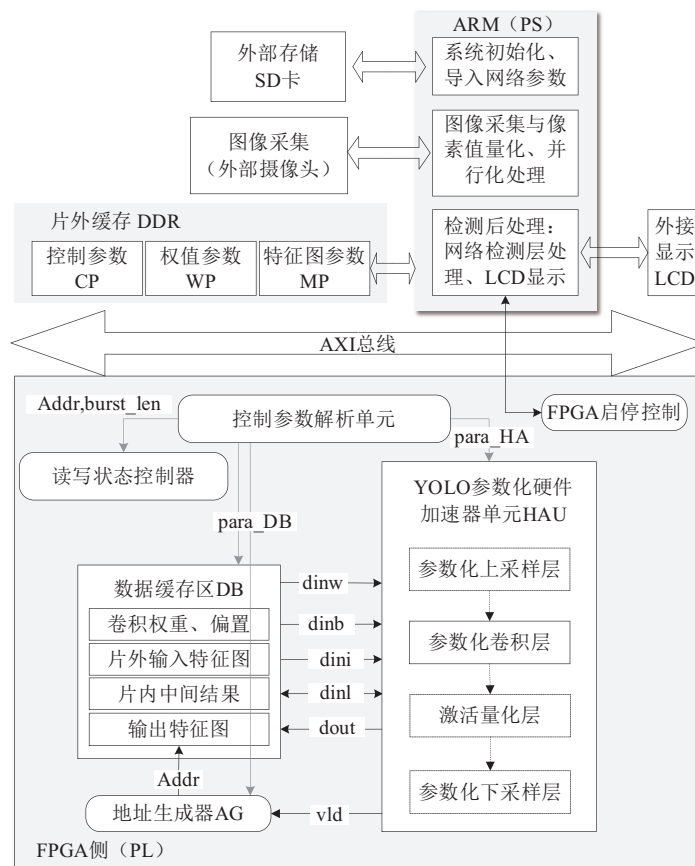


图 3 目标检测系统架构

2.2 卷积运算单元

参数化卷积层是基于权重复用、输出复用的计算方案设计的。考虑硬件资源限制,卷积运算单元由4个部分构成,32个3×3卷积窗口计算算子、可控的多层加法器树算子、可使用的偏置加法

算子、输出复用辅助算子(图4)。由此,构成一个以卷积核内部全并行为计算单元、灵活分配多输入输出通道方向并行数比例的卷积计算结构,通过参数化的控制手段,使不同卷积层的计算并行度提高,实现计算资源的利用率最大化。

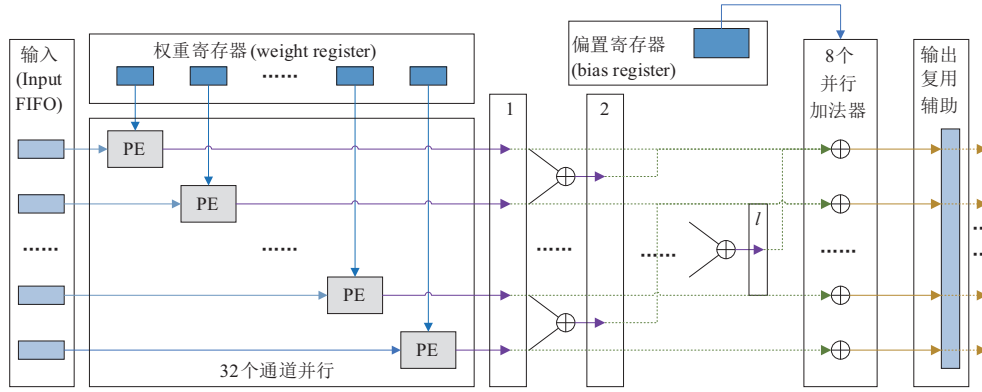


图4 卷积运算单元

1) 并行的卷积计算算子。该算子由并行的4个输出通道可调的处理单元(PE)构成,负责卷积计算的主体,也是占用计算资源的主要模块。从数据缓存区DB的卷积权重、偏置区中提取需使用的权重组和对应的偏置到各自的寄存器中,直到输入特征图的数据流进入卷积模块,依次同对应的权重进行并行运算。

一个PE最多可实现8个通道的卷积并行计算,由8个3×3的乘累加器(MAC)构成,因为轻量化YOLO中大多卷积计算中的输入通道数和输出通道数都是8的倍数,8个乘累加操作批处理单元易于调控。此外,由于RGB图像首次进入卷积层的输入通道数为3,将PE内部进一步划分为2组3个MAC和1组2个MAC,并各有1个加法器,使3组各只有1个输出。针对此3组输出,设计输出通道选择器,可分别输出2组3输入通道并行计算的结果、1组8输入通道并行计算的结果,如图5所示。

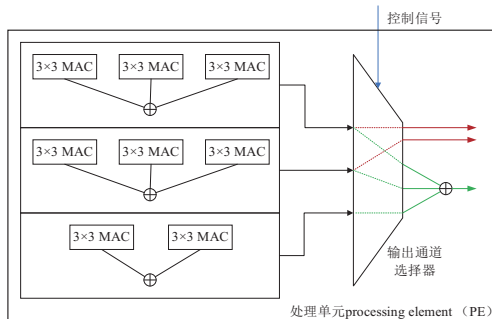


图5 输出通道可选的处理单元

2) 加法器树算子和偏置算子。这两者都由加法器构成,用于优先处理输入通道方向的计算,满足输出复用方案,以减少中间结果所需的存储空间。加法器树算子是以2层加法器组成的,输入的并行度为8,输出的并行度可为1、2、4、8。偏置算子是将加法器树的输出和对应的偏置相加,最大并行度为8。最后,输出复用辅助算子。这一算子是针对输入通道数较大,卷积模块无法将同一输出通道的所有输入通道方向数据一次性处理完成。因此先保存未处理完全的片内中间结果,并通过加法器辅助卷积计算结果的生成。推理过程中,这3个算子的调用情况都将根据输入输出通道分配比进行适应性变化。

2.3 层间融合

本文针对存在计算要求的IP核,从数学表达式出发,进行了层间融合优化,包括卷积层与归一化层融合、激活函数层与定点量化层融合,这可有效减少数据处理步骤,提高计算速度。

卷积计算公式如下

$$Y_{o,w,h} = \left(\sum_{i=1}^{in} \sum_{n=1}^k \sum_{m=1}^k W_{m,n,i,o} \cdot x_{(w+m),(h+n),i} \right) + b_o \quad (1)$$

式中: $Y_{o,w,h}$ 为第 o 个输出通道的第 w 行第 h 列的输出数据; in 为输入通道数; k 为卷积核的大小; $W_{m,n,i,o}$ 为对应位置的权重; $x_{(w+m),(h+n),i}$ 为对应位置的输入数据; b_o 为第 o 个输出通道的偏置。

批量归一化层的数学表达式如下

$$y = \gamma x + \beta \quad (2)$$

式中 γ 与 β 为常数。可知,卷积计算与归一化计算都为线性表达式,两者可以合并为一个线性表达式,使用一个卷积计算完成两个层的处理。

YOLO 中的激活函数以 LeakyReLU 为主,表达式如下

$$L_{\text{LeakyReLU}}(x) = \begin{cases} ax, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3)$$

定点量化,是在边缘设备上实现神经网络硬件加速的不可分割的一部分。数据定点化可以减少很多资源且对结果的准确性影响不大。数据量化前后的关系为式(3)。因为 FPGA 进行除法运算所消耗的资源较多,于是引入了 s_{cale} 、 s_{hift} 两个参数来表示量化的系数 S_{CALE} 。 s_{cale} 通过乘法器实现, s_{hift} 则通过数据位移实现,消耗的资源更少。

$$x' = \frac{x}{S_{\text{CALE}}} + z_{\text{ero}} - p_{\text{oint}} = s_{\text{cale}} \cdot 2^{-s_{\text{hift}}} \cdot x + z_{\text{ero}} - p_{\text{oint}} \quad (4)$$

激活函数位于定点量化之前,只需使激活函数的结果代入定点量化的输入,同样以 s_{cale} 、 s_{hift} 两个参数转换,即可完成两层的融合。

2.4 控制调度参数化

本文利用 FPGA 可编程的特性,设计了参数化的硬件推理架构,以满足不同型号的 FPGA 和不同版本的轻量化 YOLO 网络模型部署需求。然而,网络内部结构复杂、处理规模变化大,在 PL 端内置控制调度算子需要大量的逻辑资源,而通过 PS 调控需要交互的时间难以保证。因此,本文设计了参数化的控制调度方案,并开发了定制的控制参数生成算法。PL 端只需依次读取该控制参数,并根据参数内容进行推理架构的虚拟重构,即可实现整个网络的推理。

本文的控制调度主要涉及两个方面:数据流读写、数据流的推理处理。数据流读写采用 AXI 总线通信协议,连接至片外缓存 DDR,需要的信息包括图 3 中控制参数 CP、权值参数 WP、特征图参数 MP 各对应数据块的 DDR 起始地址与突发长度。为避免读写数据顺序混乱,PL 侧设计的读写状态控制器使用特定的状态机,如图 6 所示,先读取 CP,再根据 CP 中的起始地址和读写长度信息依次读取 WP、MP,当该阶段的加速完成后向 DDR 写入处理完的 MP 数据。不断循环往复直至 FPGA 停止加速。

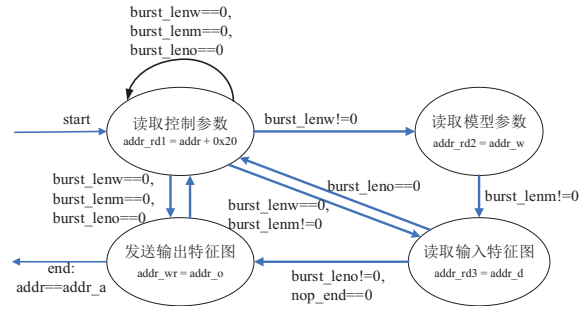


图6 读写状态控制器的状态机

数据流的推理处理则是在虚拟重构的基础上进行设计的。除了上采样层、激活量化层、下采样层的启停,各处理模块包括硬件加速单元 HAU 内各模块、数据缓存区 DB、地址生成器 AG 都需要根据特征图的处理规模而阶段性重新配置。主要参数有卷积步长、补零需求、输入特征块的尺寸、输入输出通道数等,由控制参数解析单元将压缩好的控制参数解析出来,并分别传递至 HAU、DB、AG,只需一个时钟周期重构为处理下一轮数据流对应的硬件结构。

此外,图3中的 FPGA 启停控制也通过控制参数是否已读尽来判断。PS 向 PL 传递加速开始信号后,当前已读取控制参数数目复位至 0,并启动其他模块。若控制参数已读完,当该阶段的加速过程结束并写入操作完成,将自动关闭 PL 侧的所有操作。

控制调度参数生成的前提为已知 YOLO 网络描述信息、卷积核处理并行度、片上最大存储空间。生成算法先根据并行度分配 DB 中各类数据的存储空间大小,根据 YOLO 网络中各层的处理规模确定各加速阶段的数据块大小以生成数据流读写参数,结合网络结构生成各数据块的推理处理参数。将每个阶段的参数压缩为 4×64 bit 的大小,以减少控制参数的传输时间。

3 实验结果

3.1 实验对象

本文使用 ZYNQ 板卡作为系统的验证平台,内嵌集成双核 Cortex-A9 ARM 和 Xilinx XC7Z020 FPGA 的异构芯片。硬件开发使用 Vivado 2019.2 开发环境,软件开发环境为 Xilinx SDK2019.2。

本文以 YOLOv4-Tiny 目标检测算法作为系统的实现对象。另外为减少资源的消耗,对原网络进行了 8bit 量化训练。经过损失评估,如表 1 所示,到达了网络检测标准,确定使用该网络

模型。

表 1 损失评估对比

比较对象	原网络(32 float)	未训练(8 bit)	训练后(8 bit)
损失评估(loss)	0.047 2	5.245 9	0.049 5

3.2 实验结果

在工作频率为 66.67 MHz 下,该目标检测系统的资源使用情况如表 2 所示。其中,BRAM 利用率最高,需要缓存大量数据。而,LUT 与 DSP 的利用率较高且相对平衡,这是因为两者一起承担了所有的乘法运算,并且在设计中对资源进行了预估后划分。

表 2 资源使用情况

资源	资源总数	使用数	利用率/%
LUTs	53 200	25 569.0	48.06
Flip flops	106 400	13 394.0	12.59
DSPs	220	145.0	65.91
BRAMs	140	90.5	64.64

与其他 FPGA 目标检测加速端的性能对比如表 3 所示。在资源和功耗方面,本文的优点较为突出,功耗为 1.95 W,能效为 19.70 GOPS/W,利于在边缘环境下使用。

表 3 本设计与现有成果对比

项目	文献[6]	文献[7]	文献[8]	本文
硬件平台	Zynq-z2	Ultra96 V2	Zedborad	Zynq-z2
网络结构	YOLOv4	YOLOv3-Tiny	YOLOv3-Tiny	YOLOv4-Tiny
频率/MHz	100	250	667	100
BRAMs	115	248	185	90.5
DSPs	174	242	160	145
LUTs	23.2k	27.3k	25.9k	25.6k
FFs	45.8k	38.5k	46.7k	13.4k
功耗/W	10.36	4.26	3.36	1.95
能效/GOPS/W	18.26	7.40	3.11	19.70

4 结语

本文基于边缘应用场景,提出了一种基于FPGA的轻量化YOLO目标检测系统设计方案。设计了一种系统架构来减少处理器的延时操作,并能完成整个目标检测过程。设计了一种并行卷

积运算单元来提高计算资源利用率,使用了层间融合提高计算效率,节省资源。设计了一种参数化的控制调度方法,降低系统设计的复杂度并开发了控制参数的生成算法以提高该系统的通用性。此外该控制方法对实现纯FPGA的YOLO目标检测系统也有一定的借鉴作用。本文在工作频率为66.67 MHz情况下,功耗为1.95 W,实现能效为19.70 GOPS/W。

参考文献:

- [1] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection [C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, 2016:779-788.
- [2] Lee D, Yoon S, Lee J, et al. Real-time license plate detection based on faster R-CNN[J]. KIPS Transactions on Software and Data Engineering, 2016, 5 (11): 511-520.
- [3] Redmon J, Farhadi A. YOLOv3: an Incremental Improvement [PP/OL]. arXiv preprint arXiv: 1804.02767, 2018-04-08[2024-03-25]. <https://doi.org/10.48550/arXiv.1804.02767>.
- [4] Bochkovskiy A, Wang C Y, and Liao H Y M. YOLOv4: optimal Speed and Accuracy of Object Detection [PP/OL]. arXiv preprint arXiv:2004. 10934, 2020-04-23 [2024-03-25]. <https://doi.org/10.48550/arXiv.2004.10934>.
- [5] Long X, Deng K, Wang G, et al. PP-YOLO: an Effective and Efficient Implementation of Object Detector [PP/OL]. arXiv preprint arXiv: 2007. 12099, 2020-07-27[2024-03-25]. <https://doi.org/10.48550/arXiv.2007.12099>.
- [6] Kumar A, Kalia A, Sharma A, et al. A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(6): 6783-6796.
- [7] 马宏兴,董凯兵,王英菲,等. 基于改进YOLO v5s的轻量化植物识别模型研究[J]. 农业机械学报, 2023, 54(8): 267-276.
- [8] Cao J S, Bao W S, Shang H X, et al. GCL-YOLO: a GhostConv-based lightweight YOLO network for UAV small object detection [J]. Remote Sensing, 2023, 15(20): 4932.
- [9] Zhang X F, Wang J S, Zhu C, et al. DNN Builder: an automated tool for building high-performance DNN hardware accelerators for FPGAs [C]//Proceedings of the International Conference on Computer-Aided Design. San Diego California: ACM, 2018: 1-8.

- [10] Rivera-Acosta M, Ortega-Cisneros S, Rivera J. Automatic tool for fast generation of custom convolutional neural networks accelerators for FPGA[J]. *Electronics*, 2019, 8(6):641.
- [11] Colangelo P, Segal O, Speicher A, et al. Automated hardware and neural network architecture co-design of FPGA accelerators using multi-objective Neural Architecture Search [C]//2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin). Berlin, Germany: IEEE, 2020: 1-6. DOI:10.1109/icce-berlin50680.2020.9352153.
- [12] Shahshahani M, Sabri M, Khabbazi B, et al. An automated tool for implementing deep neural networks on FPGA[C]//2021 34th International Conference on VLSI Design and 2021 20th International Conference on Embedded Systems (VLSID). Guwahati, India: IEEE, 2021:322-327.
- [13] Guo K Y, Sui L Z, Qiu J T, et al. Angel-eye: a complete design flow for mapping CNN onto embedded FPGA[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, 37(1):35-47.
- [14] Mousoulitis P G, Petrou L P. CNN-grinder: from algorithmic to high-level synthesis descriptions of CNNs for low-end-low-cost FPGA SoCs[J]. *Microprocessors and Microsystems*, 2020, 73:102990.
- [15] Belabed T, Coutinho M G F, Fernandes M A C, et al. User driven FPGA-based design automated framework of deep neural networks for low-power low-cost edge computing[J]. *IEEE Access*, 2021, 9:89162-89180.
- [16] 张萌, 张雨, 张经纬, 等. NN-EdgeBuilder: 面向边缘端设备的高性能神经网络推理框架[J]. *电子与信息学报*, 2023, 45(9):3132-3140.
- [17] 黄永福, 李亚柯. 基于FPGA的小型无人机通信干扰系统设计与实现[J]. *机械制造与自动化*, 2024, 53(5):122-125.
- [18] Zhang D Z, Wang A B, Mo R C, et al. End-to-end acceleration of the YOLO object detection framework on FPGA-only devices [J]. *Neural Computing and Applications*, 2024, 36(3):1067-1089.
- [19] Montgomerie-Corcoran A, Toupas P, Yu Z W, et al. SATAY: a streaming architecture toolflow for accelerating YOLO models on FPGA devices[PP/OL]. V1. arXiv (2023-09-04) [2024-03-25]. <https://doi.org/10.48550/arXiv.2309.01587>.
- [20] Montgomerie-Corcoran A, Yu Z W, Bouganis C S. SAMO: optimised mapping of convolutional neural networks to streaming architectures [C]//2022 32nd International Conference on Field-Programmable Logic and Applications (FPL). Belfast, United Kingdom: IEEE, 2022:418-424.
- [21] Babu P, Parthasarathy E. Hardware acceleration for object detection using YOLOv4 algorithm on Xilinx Zynq platform [J]. *Journal of Real-Time Image Processing*, 2022, 19(5):931-940.
- [22] Li J Y, Chen Y C, Yang X F, et al. Low-latency and energy-efficient FPGA accelerator for sparse neural networks in edge LiDAR-based 3D object detection[J]. *The Journal of Supercomputing*, 2025, 81(15):1423.
- [23] Adiono T, Putra A, Sutisna N, et al. Low latency YOLOv3-tiny accelerator for low-cost FPGA using general matrix multiplication principle [J]. *IEEE Access*, 2021, 9:141890-141913.
- [24] Yu Z W, Bouganis C S. A parameterisable FPGA-tailored architecture for YOLOv3-tiny [C]//Applied Reconfigurable Computing. Architectures, Tools, and Applications. Cham: Springer, 2020:330-344.
- [25] Hosseiny A, Jahanirad H. Hardware acceleration of YOLOv7-tiny using high-level synthesis tools [J]. *Journal of Real-Time Image Processing*, 2023, 20(4):75.
- [26] 裴颂文, 汪显荣. YOLO检测网络的FPGA加速计算模型的研究[J]. *小型微型计算机系统*, 2022, 43(8):1681-1686.

收稿日期:2024-03-25