嵌入式异构智能计算系统并行多流水线设计

赵二虎[1,2],吴济文[1],肖思莹[1],晋振杰[1],徐勇军[1]
(1. 中国科学院计算技术研究所专项技术研究中心,北京 100190;2. 中国科学院大学,北京 100049)

摘 要: 嵌入式智能计算系统因其功耗受限和多传感器实时智能处理需要,对硬件平台的智能算力能效比和智能计算业务并行度提出了严峻挑战. 传统嵌入式计算系统常采用的DSP+FPGA数字信号处理架构,无法适用于多个神经网络模型加速场景. 本文基于ARM+DLP+SRIO嵌入式异构智能计算架构,利用智能处理器多片多核多内存通道特性,提出了并行多流水线设计方法. 该方法充分考虑智能计算业务中数据传输、拷贝、推理、结果反馈等环节时间开销,为不同的神经网络模型合理分配智能算力资源,以达到最大的端到端智能计算业务吞吐率. 实验结果表明,采用并行多流水线设计方法的深度学习处理器利用率较单流水线平均提高约25.2%,较无流水线平均提高约30.7%,满足可见光、红外、SAR等多模图像实时智能处理需求,具有实际应用价值.

关键词: 嵌入式智能计算系统;异构计算架构;神经网络模型;并行多流水线;深度学习处理器
基金项目: 中国科学院技术支撑人才项目;北航杭州创新研究院钱江实验室开放基金(No.2020-Y8-A-023)
中图分类号: TP389.1 **文献标识码:** A **文章编号:** 0372-2112(2023)11-3354-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20230281

Parallel Multi Pipeline Design of Embedded Heterogeneous AI Computing Systems

ZHAO Er-hu[1,2], WU Ji-wen[1], XIAO Si-ying[1], JIN Zhen-jie[1], XU Yong-jun[1]
(1. *Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*;
2. *University of Chinese Academy of Sciences*, *Beijing* 100049, *China*)

Abstract: Due to the limited power consumption and the need for real-time intelligent processing of multiple sensors, embedded AI computing systems desire for higher energy efficiency and more parallel intelligent computing services simultaneously. The digital signal processing architecture DSP+FPGA commonly used in traditional embedded computing systems is not suitable for multiple ANN models inference acceleration. Based on embedded heterogeneous intelligent computing architecture ARM+DLP+SRIO, this paper proposes a parallel multi pipeline design method by taking advantage of the characteristics of multi chip, multi-core and multi memory channels of deep learning processors. Considering the time cost of data transmission, copy, reference and feedback, this method allocates intelligent computing resources for different neural network models to achieve the maximum end-to-end throughput. The experimental results show that the utilization of the deep learning processor using the parallel multi pipeline design method is about 25.2% higher than that of a single pipeline, and about 30.7% higher than that without pipeline. It meets the real-time intelligent processing requirements of visible light, infrared and SAR images, and is valuable for practical applications.

Key words: embedded AI computing systems; heterogeneous computing architecture; neural network model; parallel multi pipeline; deep learning processor (DLP)
Foundation Item(s): CAS Technology Support Talent Program; Hangzhou Innovation Institute, Beihang University (No.2020-Y8-A-023)

1 引言

以深度学习为代表的人工智能技术持续发展,促进嵌入式计算系统向更高阶的嵌入式智能计算系统[1,2]演进,旨在将人工智能赋予更加边缘、机动、专用的物端装置或边缘设备,例如工业机器人、无人机、无人车等. 随着传感器向小型化、高精度发展,无人平台

收稿日期:2023-03-30;修回日期:2023-10-16;责任编辑:王天慧

将集成越来越多的传感器用于环境感知. 如美军“敏捷秃鹫”(Agile Condor)[3,4]无人机智能吊舱同时集成了可见光、红外、合成孔径雷达(Synthetic Aperture Radar, SAR)等目标探测器,我国高空高速侦察无人机[5,6]也配装了多种侦察设备,这就要求嵌入式智能计算系统需具备对多源感知数据的实时智能处理能力.

嵌入式智能计算涉及到数据接入、数据缓存、数据分发、智能前处理、智能推理、智能后处理等环节,虽然依靠神经网络加速芯片(Neural network Processing Unit, NPU)能够提高推理速度,但无法保证端到端吞吐性能,因此我们不能仅仅优化某个特定环节,而应该尽量同时优化系统的各个部分,使得原本串行处理的数据流能够实现并行流水处理. 流水线设计需要软硬件协同才能获得较好性能,尤其在面临多个神经网络模型加速计算需求时,考虑到嵌入式平台的算力不足和能量受限,必须设计并行多流水线,才能最大程度提高系统吞吐量.

2　相关工作

在流水线设计方面,已有的研究大多聚焦在单一计算业务,对于多个计算业务尤其是多个智能计算业务的并行流水线研究还比较少. 文献[7,8]在Cell多核处理器上构建流水执行模型,实现了线程同步流水并行和迭代同步流水并行两方面的优化技术,改进了内存密集型计算的吞吐量. 文献[9,11]从事务、任务、数据流等不同层面设计流水线调度策略和优化方法,并兼顾考虑了计算平台的同构性、异构性和业务分布式特性. 文献[12]在嵌入式异构平台Tegra K1的多核CPU上以流水线方式运行LBP人脸检测算法,并将数据并行任务卸载到GPU来获取加速效果. 文献[13~19]阐述了求解多目标调度问题的常见算法,这些方法大部分需要确定多维度权重值,然而在面临多个智能计算业务时,加上嵌入式计算资源受限,对于多维度权重值的确定来说无疑是“雪上加霜”. 并行多流水线设计重点关注在有限的智能计算资源上如何保证多个智能计算业务的时效性. 近年来快速发展的神经网络加速器[20,21]为多个智能计算业务的并行计算提供了新的技术途径.

3　问题描述

在嵌入式智能计算系统的峰值算力相对固定的条件下,为了最大限度提高智能计算业务吞吐量,除了从算法角度对神经网络模型进行压缩和轻量化之外,还需要重点从系统层面对智能计算业务的全流程进行优化. 以基于神经网络模型的目标智能检测任务为例,在嵌入式智能计算系统中,我们可以将目标智能检测任务分为图像接收与预处理、图像缓存与拷贝、AI模型推理计算、推理结果后处理等阶段,如图1所示.

以航空遥感应用为例,越来越多的探测器或传感器被集成到飞行器平台,以期获取更多维的目标特征信息. 假设有可见光、红外、雷达等n种目标探测手段,针对每个探测手段所获取的目标图像数据,分别训练一个专用的神经网络模型进行目标检测,并且需要对所有的AI模型同时加速计算. 假设n个AI计算任务都按照相同的处理流程依次执行m个处理环节(如图1所示的AI计算任务需要经过4个处理环节),且每相邻2个处理环节之间有一定的缓冲区,那么多个神经网络模型的并行智能加速问题可以抽象为一个有限缓冲区流水线调度问题[13]. 本文不从算法层面去具体求解该问题,而是从方法层面去研究如何设计并行多流水线,并基于实际的嵌入式异构智能计算系统对并行多流水线的计算效率进行评测.

图1　基于神经网络模型的目标智能检测图像批处理流程

流水线是一种时间重叠并行处理技术,其最大吞吐量性能与组成流水线的各流水段的处理耗时有关. 一条流水线的分段越多、各段处理耗时越长,则流水线的吞吐量就越小. 在建立流水线时,需要对耗时较长的流水段(即瓶颈段)进行拆分或设置多条相同的流水段,以提高数据处理并行度. 对耗时较短的流水段,可以将多个流水段组合成一个流水段. 如果采用RapidIO,PCIe等高速数据总线,则传输一张图片的时间开销很小,相比其他环节耗时,传输时延可以忽略,因此我们可以将图像传输、图像预处理组合成一个操作流水段. 而图像缓存、推理计算、推理结果后处理等操作耗时较长,可作为独立的3个操作流水段. 上述每个流水段在系统软件内以独立的进程或线程存在,流水段之间采用同步阻塞队列进行通信. 批处理是图像智能处理的高效率方法,每批(batch)图像的智能计算过程将以串行方式进行处理,单业务多批次图像智能计算过程将以单流水线方式进行处理,多业务多批次图像智能计算过程则采用多流水线方式并行处理.

4　嵌入式智能计算平台

4.1　平台架构

在解决上述问题之前,首先概要介绍一下嵌入式智能计算平台架构[2],如图2所示. 本文开展的面向多个神经网络模型加速的嵌入式智能计算系统并行多流水线设计,均是依托此硬件平台进行设计和测试的.

嵌入式智能计算平台基于ARM+DLP+SRIO异构

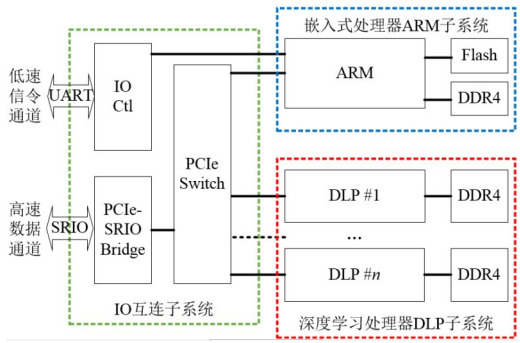


图2 嵌入式智能计算平台架构[2]

智能计算架构. 该架构中,DLP(Deep Learning Processor)是指深度学习处理器,也可称为智能芯片,适用于执行数据密集型计算任务,主要负责神经网络模型中各类算子的运算加速. DLP选用寒武纪MLU100智能芯片[22],集成32个MLUv01加速器内核,设计主频为1 GHz,整形INT8的理论峰值性能为32TOPS,半精度浮点FP16的理论峰值性能为16TFLOPS. MLU100集成了DDR4控制器,设计内存容量为256-bit 8 GB,内存带宽达到102.4 GB/s. ARM是通用嵌入式微处理器,适合执行指令密集型计算任务,主要负责智能计算业务的数据传输、流程调度和DLP驱动管理. ARM选用FT-2000/4芯片[23],片内集成4个FTC663内核和2个DDR4控制器,主频2.2 GHz,峰值算力为38.4 GFLOPS,设计内存容量为64-bit 4 GB,访存带宽51.2 GB/s. ARM与DLP之间通过PCIe总线互连,ARM与外部传感器之间通过高速SRIO总线互连. SRIO(Serial RapidIO)是一种为嵌入式系统开发的串行高速总线,采用包交换互连技术,常用的RapidIO2.0标准[24]单通道信号速率达到6.25 GBaud/lane,最新的RapidIO4.1标准[25]单通道信号速率达到25.781 25 GBaud/lane,适用于板间大带宽数据传输,在可靠性、带宽、可扩展性等方面具有优势. ARM+DLP+SRIO架构兼具了通用计算、并行智能计算、高速数据接入能力,而且在性能功耗比方面较传统x86+GPU架构有优势,为嵌入式环境下多个神经网络模型并行加速提供了平台支撑.

4.2 DLP多核架构

DLP的多核处理器架构和多通道访存模式是支撑多模型并行流水线设计的关键因素. MLU100内部共有32核和4个内存通道,各个核之间通过片上网络(Network on Chip,NoC)与4个DDR控制器相连,抽象的机器模型如图3所示[26]. 片内32核均能够访问外部DRAM上的地址范围,协同工作进行神经网络的推理计算,但是每个核访问不同DDR通道的性能存在差异. 基于上述多核并行计算架构和多通道DRAM访存模式,可以采用数据并行和模型并行这2种并行编程方式. MLU100的软件开发环境为Cambricon NeuWare[27]. 在部署深度学习模型时,可根据实际应用需求在NeuWare中指定所需的处理器核数量,并能够将神经网络模型在拓扑结构、输入输出、模型参数等多个唯独进行划分,使划分后的模型能够同时在多个处理器核上并行地执行,且自动保证多核间的数据同步;同时NeuWare还支持多组处理器核上同时运行多个模型来处理多份输入数据,各组处理器核在计算时共享指令和参数,提高深度学习推理计算的端到端吞吐率.

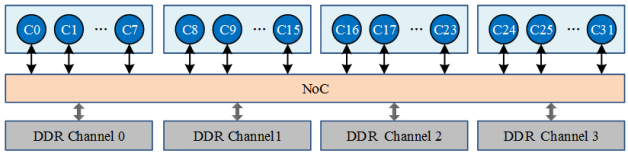


图3 MLU100深度学习处理器的多核抽象机器模型

5 并行多流水线设计

5.1 多流水线原理

考虑从3个方面来进行流水线设计:(1)构建串行流水线,针对单一智能计算业务,沿着数据流向建立串行流水线,缩短从图像接收到结果后处理的平均计算时间(称为makespan);(2)增加数据并行度,充分利用高速数据总线的大带宽、低时延优势,对图像数据进行适当缓存,再送进智能芯片,依托智能芯片的多核架构优势,执行批处理操作,提高处理效率;(3)建立并行多流水线,针对多类型智能计算业务所对应的多个神经网络模型,充分发挥智能芯片的多通道架构优势,支持多个AI模型并行加载到高速缓存区,同时对并行多流水线调度策略进行优化,提高多业务并行处理能力.

为充分利用智能芯片DDR内存多通道结构优势,智能计算软件采用多流水线并行处理机制,为每类图像的神经网络模型创建一条智能推理流水线,每条流水线均由图像接收与预处理、数据缓存与拷贝、AI模型推理计算、推理结果后4个流水段组成. 与单条流水线最大吞吐率相比,4条流水线的最大吞吐率提高了4倍. 智能计算系统多流水线并行处理工作原理如图4所示.

5.2 智能计算流水线设计

如上所述,图像数据在嵌入式智能计算系统中推理计算全过程需要建立起流水线,流水段节点(线程)之间应能同时进行并行计算,进而使系统整体达到最大图像处理吞吐率. 一般认为前向推理计算流水段针对给定的图像像素数据类型和固化的智能算法模型来说较趋于稳定,而数据的传输阶段是有较大优化空间的,也是建立流水线的关键. 因此下面对数据在嵌入式智能计算系统中的传输过程进行详细说明及提出优化思路.

以航空遥感图像处理为例,嵌入式智能计算系统

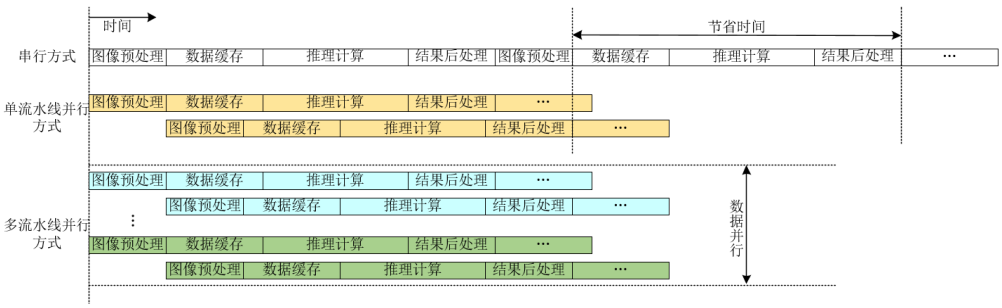


图4　多流水线并行计算工作原理

需要在每个250 ms间隔内处理完10张$W\times H$大小的图像,接着拷贝进智能芯片DLP进行前向推理,再将推理完的结果拷出到嵌入式微处理器ARM上进行与外围设备的信息交互. 由于ARM和DLP处理的数据格式有所不同,因此在数据拷入到DLP时需要进行数据重新摆放、数据通道补齐等操作,然后方可进行拷贝. 数据在嵌入式智能计算系统中的传输过程如图5所示.

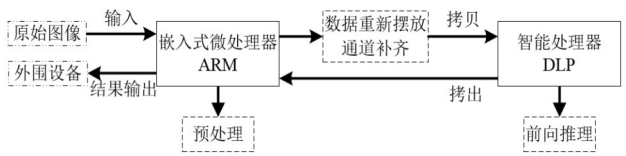


图5　数据在嵌入式智能计算系统中的传输过程

ARM接收到原始数据之后首先对图像进行预处理,然后将数据拷贝到智能芯片进行前向推理,这就涉及预处理时间、拷贝时间以及前向推理时间. 经过验证,针对该实验中所用固化模型前向推理时间也基本稳定在220 ms左右. 但由于数据拷贝时间较长,数据拷贝进智能芯片的时间远大于智能芯片执行时间,这样智能芯片就会有不进行工作的空闲时间,导致资源没有被充分利用. 此时就认为是流水线没有建立起来,数据拷贝时间过长是流水线建立的瓶颈问题,如图6所示. 由于拷贝时间远大于智能芯片进行前向推理的时间,因此即使在使用多线程运行模式的情况下,智能芯片仍然存在资源闲置问题,导致流水线无法建立起来,影响系统整体运行效率.

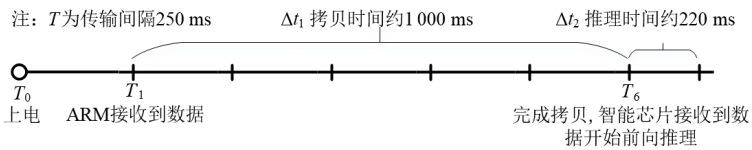


图6　数据拷贝对流水线建立的瓶颈示意图

对数据传输阶段的优化主要从数据的预处理、数据重新摆放、通道补齐操作来着手进行. 在之前的实验测试中,数据预处理、数据重新摆放和通道补齐操作(以下统称为“转数和对齐操作”)是优先放在嵌入式微处理器端进行的. 考虑到ARM性能有限,因此针对流水线无法建立问题的优化思路是选择将数据预处理、转数和对齐操作优先放到智能芯片端进行.

优化前后的数据处理流程对比如图7所示. 优化后将数据预处理、转数和对齐操作都放在智能芯片上进行,能减少ARM的预处理及拷贝时间. 这一优化方法虽然增加了智能芯片的处理任务,但经过验证并不会大幅增加智能芯片的运行时间. 智能芯片仍然能高效运行预处理、转数和对齐以及前向推理操作. 这样通过略微增加智能芯片的片内执行时间来减少嵌入式微处理器拷贝时间,从而提升整个系统的端到端运行效率的方法可以成功建立起流水线,充分利用智能计算资源.

使用优化方法建立起流水线之后的数据处理过程

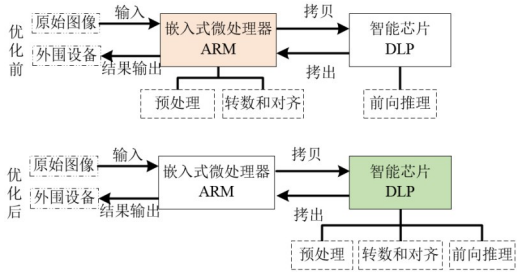


图7　优化前后的数据处理流程

如图8所示. 经过优化后建立起的流水线,拷贝时间明显变少而智能芯片执行时间只有毫秒级的增加. 开启多线程工作模式之后,多个线程之间可以同时运行各自的处理流程,从而建立起流水线模式,提升系统整体的端到端运行效率.

5.3　并行多流水线设计

针对多模图像检测识别模型(如可见光、红外、SAR等)的并行加速计算问题,可以构建并行多流水线,通过对DLP多核计算资源和多模图像计算任务分配,实

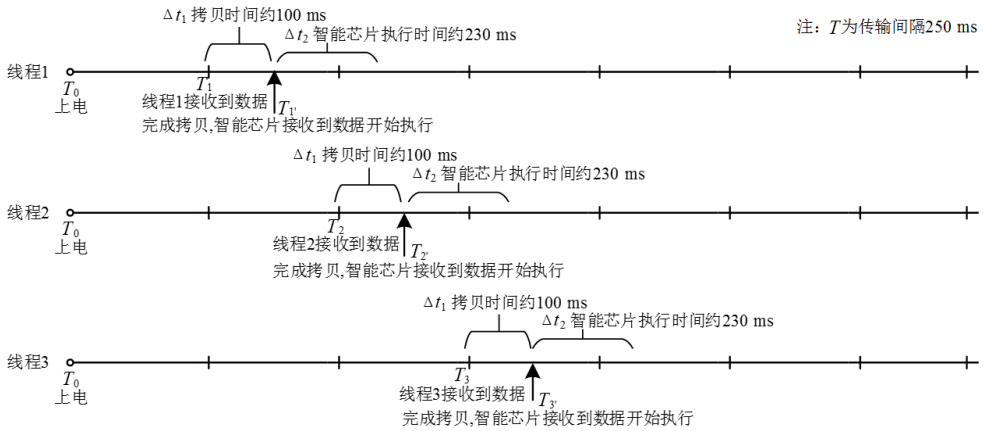


图8 优化后的流水线数据处理过程

现对多个神经网络模型的并行推理计算. 假定嵌入式智能计算系统要处理的红外、可见光、SAR三模图像数据特征如表1所示. 根据各类图像像素大小不同,1张可见光大图可以分割成10张图片切片,1张红外大图刚好分割成1张图片切片,1张SAR图可以分割成4个图片切片.

表1 待处理的多模图像数据特点

属性	可见光	红外	SAR
大图尺寸	$W_v \times H_v$	$W_i \times H_i$	$W_r \times H_r$
成像帧率/fps	20	20	1
图像切片尺寸	$W' \times H'$		
切片帧率/fps	200	20	4
切片总帧率/fps	224		

为了使图像切片$W' \times H'$的智能处理效率达到224 fps,智能处理软件设计需完成多业务并发的流水线编程,包括可见光图像目标检测计算流水线、红外图像目标检测计算流水线、SAR图像目标检测计算流水线. 各条流水线均由数据拷贝阶段(传输)、推理计算阶段(推理)、推理结果后处理阶段(反馈)3个环节组成,所需时间分别标记为t_1, t_2, t_3. 如图9所示. 数据拷贝阶段主要功能是接收缓存的多模图像数据,并将图像数据拷贝输入至检测/识别模型输入层. 推理计算的主要功能是对每个batch图像进行神经网络模型推理计算,记录端到端推理计算时间、智能芯片片内推理计算时间信息;推理计算模块在上电时还完成了模型加载、模型函数抽取、模型输入输出适配、核心数量分配计算,以及为模型运行在智能芯片上开辟空间等. 推理结果后处理阶段主要实现每个batch图像计算结果从DLP芯片拷贝至ARM内存,经过后处理计算,按照约定的协议格式发送至上游主控单元. 针对多传感器图像智能处理设计了并行多流水线,包括但不限于可见光图像目标智能检测流水线、红外图像目标智能检测流水线、SAR图像目标智能检测流水线. 假设3个传感器图像数据会并行输入到系统中,三模图像的类别分别用上标1,2,3来标识. 如果把传输任务分配给ARM内核A,把推理任务分配给ARM内核B,把反馈任务分配给ARM内核C,那么就能以较低系统开销和时延将数据从一个进程传输到另一个进程,并且流水线地执行多个并发业务. 在推理执行过程中,ARM内核B会对外部智能协处理器DLP进行调度,将推理过程进一步拆分为多个神经网络算子,并交由DLP的32个核并行运算.

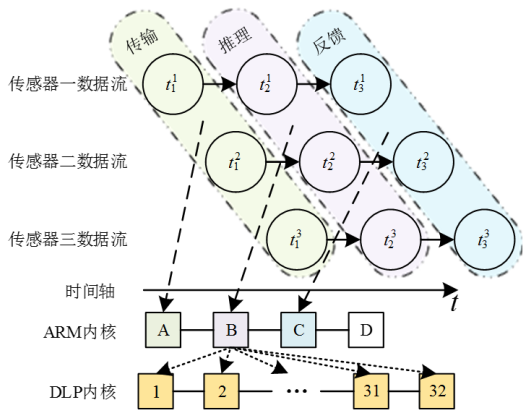


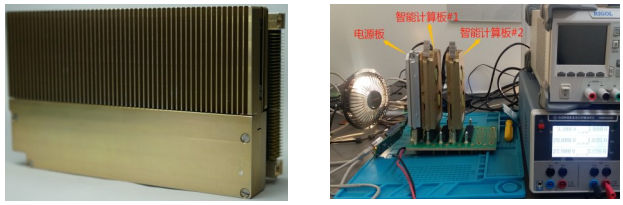
图9 面向异构多核的并行流水线任务映射

6 实验验证

6.1 实验环境

为了测试嵌入式智能计算系统并行多流水线性能,我们搭建了如图10所示的实验环境. 其中,图10(a)所示的嵌入式智能计算板卡是基于图2所示的ARM+DLP+SRIO架构研制. 经实测,一块嵌入式智能计算板卡对$W' \times H'$图像切片的智能处理效率约为65 fps,无法满足表1三模图像总吞吐率224 fps要求. 为此,嵌入式智能计算系统至少需采用双智能计算板卡多流水线并行工作模式,如图10(b)所示. 考虑到可见光是三模图像中数据量最大的传感器,因此将智能计

算板#1的算力全部服务于可见光的目标检测任务，智能计算板#2负责余下可见光以及全部红外、SAR的目标检测任务.



(a) 嵌入式智能计算板卡　　(b) 双智能计算板工作场景

图10　嵌入式智能计算系统实验环境

6.2　算力分配

智能计算板#1的算力分配如图11所示，其全部算力用于可见光图像目标检测，4个DDR通道并行加载4个可见光目标检测模型，每路模型输入参数$N\times C\times W\times H$设为$7\times 1\times W'\times H'$，计算任务是每个流水周期处理3帧可见光图片，帧号为$\{5n-4,\ n>0\}$，$\{5n-3,\ n>0\}$，$\{5n-2,\ n>0\}$，合计30个图像切片. 由于当前4个DDR通道一次批处理的图像切片数量为28，考虑到相邻两帧大图之间存在重叠区域，我们选择丢弃可见光第$\{5n-2,\ n>0\}$帧的第1和第2个图像切片，以满足3张可见光大图的实时性处理要求.

智能计算板卡#2的算力分配如图12所示，同时服务于可见光、红外、SAR图像目标检测. DDR通道#1~#3各加载1个可见光目标检测模型，每路模型输入参数$N\times C\times W\times H$设为$6\times 1\times W'\times H'$，计算任务是每个流水周期处理2张可见光大图，帧号为$\{5n-1,\ n>0\}$，$\{5n,\ n>0\}$，共20个图像切片. 由于当前3个DDR通道一次批处理的图像切片数量为18，考虑到相邻两帧大图存在重叠区域，我们选择丢弃可见光第$\{5n-2,\ n>0\}$帧的第1和第2个图像切片，以满足2张可见光大图的实时性处理要求. DDR通道#4同时加载1个红外目标检测模型和1个SAR目标检测模型，红外模型输入参数$N\times C\times W\times H$设为$5\times 1\times W'\times H'$，SAR模型输入参数$N\times C\times W\times H$设为$4\times 1\times W'\times H'$，计算任务是每个流水周期处理5帧红外大图和1帧SAR大图，共9个图像切片.

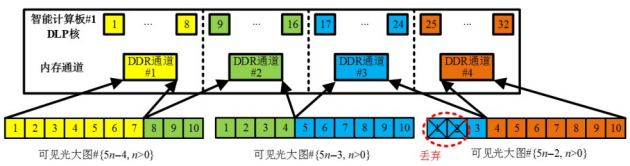


图11　智能计算板#1的计算任务及多核资源分配

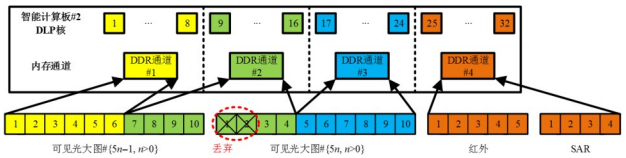


图12　智能计算板#2的计算任务及多核资源分配

6.3　并行多流水线测试结果

基于上述算力分配方法，设计了并行多流水线方案，如图13所示. 对于可见光图像的目标智能检测，智能计算板#1需要在11个成像周期(550 ms)内完成3帧可见光图像的传输、拷贝和推理；智能计算板#2需要在5个成像周期(250 ms)内完成2帧可见光图像的传输，在13个成像周期(650 ms)内完成2帧可见光图像的拷贝和推理. 对于红外图像的目标智能检测，智能计算板#2需要在5个成像周期(250 ms)内完成5帧红外图像的传输，在13个成像周期(650 ms)内完成5帧红外图像的拷贝和推理. 对于SAR图像的目标智能检测，智能计算板#2只要在20个成像周期(1 s)内完成1帧SAR图像的传输、拷贝和推理即可. 由于SAR图像帧率低，数据量较小，我们按照8个成像周期(400 ms)完成1帧SAR图像的传输、拷贝和推理来设计.

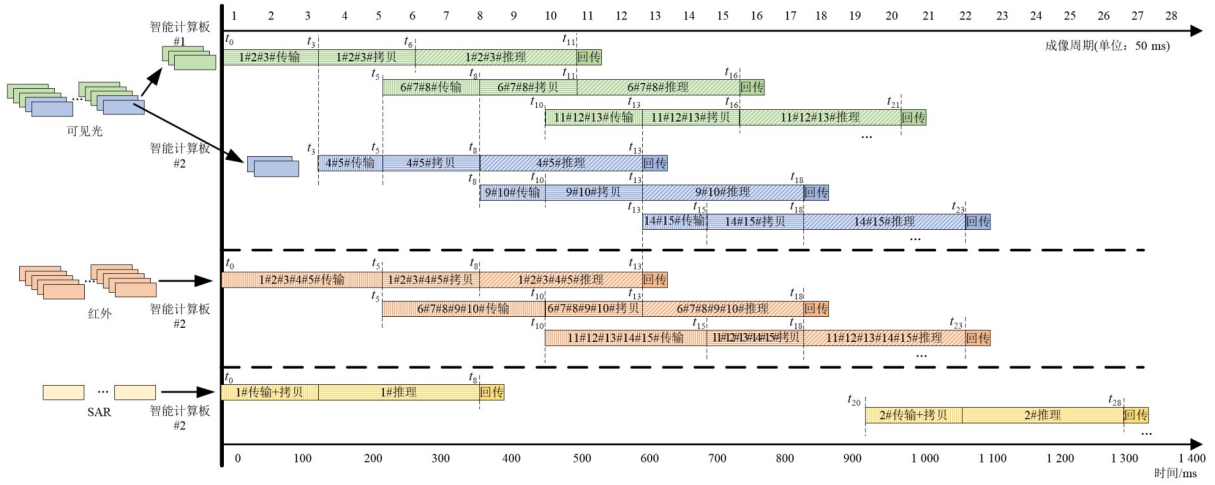


图13　并行多流水线设计方案

测试可见光、红外、SAR图像目标检测流水线从上电到流水线建立前后各阶段的时间开销，我们用T_1表示从板卡上电到ARM内存中缓存够1个batch大图的时间开销；用T_2表示将1个batch图像从ARM数据缓存队列拷贝至DLP存储空间的时间开销；用T_3表示1个batch图像在DLP上推理计算的时间开销；用T_4表示将DLP上1个batch图像的推理计算结果拷贝至ARM存储空间的时间开销；用T表示流水线建立总的时间开销，即$T=T_1+T_2+T_3+T_4$. 经过多次批处理测试，获得各个环节的平均时间开销如表2所示.

表2 并行多流水线的各流水段时间开销 单位：ms

AI算力	传感器	ARM数据接收T	ARM→DLP数据拷贝T	DLP模型推理T	DLP→ARM推理结果反馈T	总计T
智能计算板#1	可见光{5n−4,5n−3,5n−2 | n>0}	150	71	237	<1(可忽略)	458
智能计算板#2	可见光{5n−1,5n | n>0}	100	71	236	<1(可忽略)	407
	红外	250	72	226	<1(可忽略)	548
	SAR	150	98	44	<1(可忽略)	292

将图13和表2进行对比，可以看出多流水线的实测效果优于设计方案，如图14所示. 可见光#1的设计流水线周期为551 ms，实测流水线周期为459 ms；可见光#2的设计流水线周期为501 ms，实测流水线周期为408 ms；红外的设计流水线周期为651 ms，实测流水线周期为549 ms；SAR的设计流水线周期为401 ms，实测流水线周期为293 ms. 可以看出，各个传感器的实测流水段时间开销均小于或等于设计方案，直观上能够满足航空侦察图像智能目标检测实时处理需求，下面对其进行量化分析.

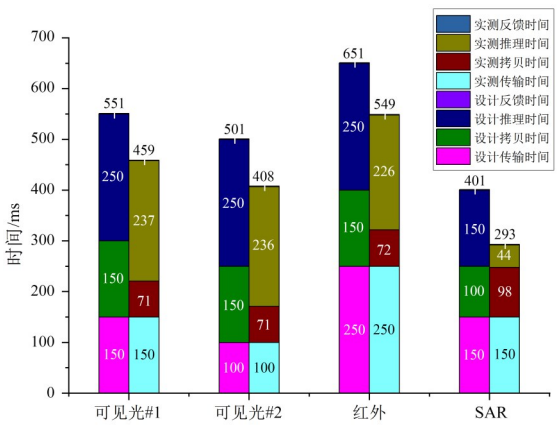


图14 并行多流水线设计方案与实测结果对比

6.4 实时性分析

如表2所示，可将光#1的流水线周期为T_3=237 ms，每个流水周期能处理3张可见光大图(折合30个图像切片)；可将光#2的流水线周期为T_3=236 ms，每个流水周期能处理2张可见光大图(折合20个图像切片)；红外的流水线周期为T_1+T_2=322 ms，每个流水周期能处理5张红外大图(折合50个图像切片)；SAR的流水线周期为T_1+T_2=248 ms，每个流水周期能处理1张红外大图(折合4个图像切片). 由于多模图像处理为并行方式，整个系统的单流水线周期取最大值为322 ms，并在该时间内处理了104张不同类型的图像切片，单张图像切片处理时间为322/104≈3 ms，即333 fps，大于所需要的224 fps(见表1). 基于以上分析可以得出：采用并行多流水线设计方法后，基于ARM+DLP+SRIO的嵌入式智能计算系统实现了可见光、红外、SAR等多模图像实时智能处理需求.

表3列出了典型YOLO模型的端到端吞吐率(折算成1 280×1 280后)，从中可以看出，本文所采用的基于ARM+DLP+SRIO的嵌入式智能计算系统算力条件并不突出(64 TOPS<130 TOPS)，但是统一折算后的端到端吞吐率指标达到333 fps，较表中所列其他文献均有一定优势. 由此可见，本文的并行多流水线设计方法能够更好地发挥出硬件平台的智能算力资源，提高多传感

表3 若干典型模型的端到端吞吐率比较

文献	算法模型	硬件平台	int8算力	图像尺寸	吞吐率/fps	吞吐率/fps
文献[29]	YOLOv4	Telsa T4	130 TOPS	32×32	1 085	0.7
文献[3]	CNN	GM107×6	7.5 TFLOPS	32×32	5 000	3.1
文献[32]	YOLOv7	Telsa T4	130 TOPS	1 280×1 280	17	17
文献[12]	LBP	Tegra K1×192	70 TFLOPS	1 920×1 080	29	36.7
文献[34]	YOLOx	Telsa T4	130 TOPS	640×640	396	99
文献[30]	YOLOv5	Telsa T4	130 TOPS	1 280×1 280	175	175
文献[33]	YOLOv8	Telsa T4	130 TOPS	640×640	734	183.5
文献[31]	YOLOv6	Telsa T4	130 TOPS	1 280×1 280	281	281
本文	YOLOv3	MLU100×2	64 TOPS	1 280×1 280	333	333

器图像智能处理实时性.

6.5 DLP利用率分析

下面计算无流水线、单流水线、多流水线3种方法的DLP利用率(用η表示). 无流水线情况下,DLP利用率的计算方法为

$$\eta_{无}=\frac{t_{推理}}{t_{传输}+t_{拷贝}+t_{推理}} \tag{1}$$

依据式(1)和图13,可以计算出无流水线情况下可见光#1、可见光#2、红外的DLP利用率分别为45.5%、50%、38.5%.

单流水线情况下,DLP利用率的计算方法为

$$\eta_{单}=\frac{T_3}{T_\Delta} \tag{2}$$

依据式(2)和表2,可以计算出单流水线情况下可见光#1、可见光#2、红外的DLP利用率分别为51.7%,58.0%,41.2%.

并行多流水线情况下:

(1)可见光#1的各个处理阶段的时间占比如图15所示(因回传时间占比很小,以下分析均忽略此时间).

(2)可见光#2的各个处理阶段的时间占比如图16所示.

(3)红外的各个处理阶段的时间占比如图17所示.

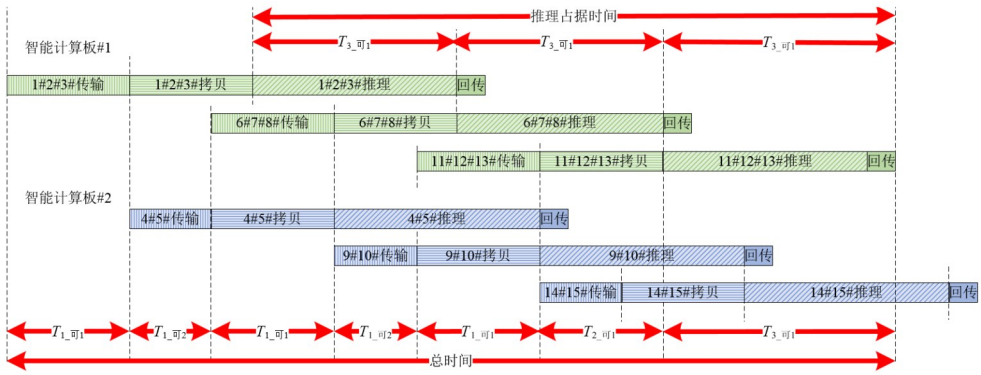


图15　多流水线情况下可见光#1的时间段占比

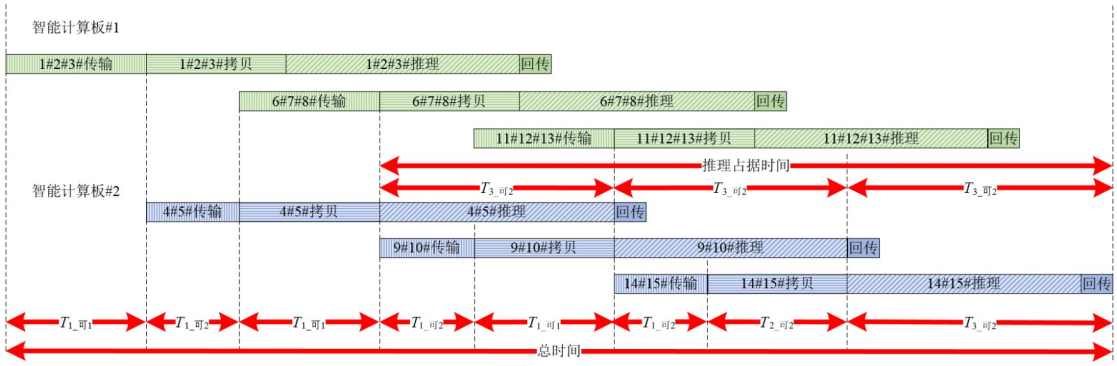


图16　多流水线情况下可见光#2的时间段占比

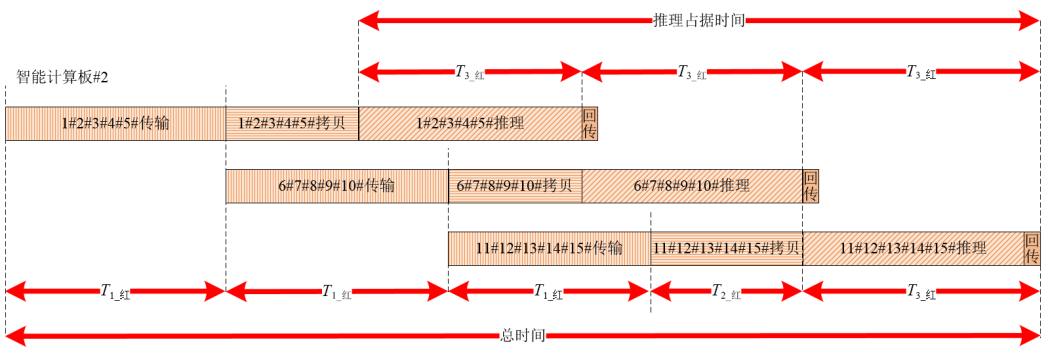


图17　多流水线情况下红外的时间段占比

从图15中可以计算出可见光#1的DLP利用率为

$$\eta_{可1多}=\frac{推理占据时间}{总时间}=\frac{T_{3\_可1}\times 3}{T_{1\_可1}\times 3+T_{1\_可2}\times 2+T_{2\_可1}+T_{3\_可1}}=\frac{237\times 3}{150\times 3+100\times 2+71+237}\approx 74.2\% \tag{3}$$

从图16中可以计算出可见光#2的DLP利用率为

$$\eta_{可2多}=\frac{推理占据时间}{总时间}=\frac{T_{3\_可2}\times 3}{T_{1\_可1}\times 3+T_{1\_可2}\times 3+T_{2\_可2}+T_{3\_可2}}=\frac{307\times 3}{150\times 3+100\times 3+71+236}\approx 87.1\% \tag{4}$$

从图17中可以计算出红外的DLP利用率为

$$\eta_{红多}=\frac{推理占据时间}{总时间}=\frac{T_{3\_红}\times 3}{T_{1\_红}\times 3+T_{2\_红}+T_{2\_红}}=\frac{226\times 3}{250\times 3+72+226}\approx 64.7\% \tag{5}$$

无流水线、单流水线、多流水线3种方法的DLP利用率对比情况如图18所示，从中可以看出，采用并行多流水线后的DLP利用率较单流水线DLP利用率平均提高约25.2%，较无流水线DLP利用率平均提高约30.7%.

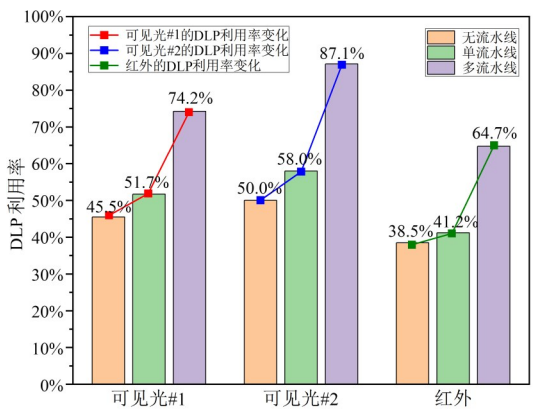


图18 不同流水线方法DLP利用率对比

6.6 应用边界

本文所研究的基于ARM+DLP+SRIO嵌入式异构智能计算架构的并行多流水线设计方法，不仅可应用在航空遥感图像领域，还可以应用于其他领域（如车载、船载、无人系统等）. 该方法的应用边界在于基于ARM+DLP+SRIO的嵌入式智能计算系统的数据拷贝时间远大于智能芯片进行前向推理的时间. 针对该应用边界，笔者经过多次实验证明：未经过优化的基于ARM+DLP+SRIO的嵌入式智能计算系统，除了航空遥感图像应用领域之外，针对其他领域的图像仍存在数据拷贝时间大于智能芯片前向推理时间这种情况. 针对其他领域应用，该方法需要调整的参数主要是图像切片的大小W×H，因为该嵌入式智能计算系统在硬件架构和智能算法确定条件下，最大算力相对固定，每秒处理的图像切片数目也相对固定，所以图像切片的大小直接影响数据拷贝时间和智能芯片前向推理时间.

7 结论

本文研究了嵌入式多业务多模型并行智能加速问题，利用智能处理器多片多核多内存通道特性，提出了并行多流水线设计方法. 该方法综合考虑智能计算业务中数据传输、拷贝、推理、结果反馈等环节时间开销，为不同的神经网络模型合理分配智能算力资源，优化端到端智能计算业务吞吐率. 本文方法对边缘设备多传感器智能处理问题具有一定指导意义. 随着深度学习处理器及智能计算硬件[28]的发展，未来在嵌入式智能计算资源分配和端到端数据流模型方面还有继续优化空间，值得进一步研究.

参考文献

[1] 陈云霁, 李玲, 李威, 等. 智能计算系统[M]. 北京: 机械工业出版社, 2020.
CHEN Y J, LI L, LI W, et al. AI Computing Systems[M]. Beijing: China Machine Press, 2020. (in Chinese)

[2] 赵二虎, 吴济文, 查晶晶, 等. 基于ARM+DLP+SRIO的嵌入式智能计算系统研究[J]. 电子学报, 2021, 49(3): 443-453.
ZHAO E H, WU J W, ZHA J J, et al. Embedded AI computing system based on ARM+DLP+SRIO[J]. Acta Electronica Sinica, 2021, 49(3): 443-453. (in Chinese)

[3] ISEREAU D, CAPRARO C, COTE E, et al. Utilizing high-performance embedded computing, agile condor, for intelligent processing: An artificial intelligence platform for remotely piloted aircraft[C]//2017 Intelligent Systems Conference (IntelliSys). Piscataway: IEEE, 2017: 1155-1159.

[4] Barnell M, Raymond C, Capraro C, et al. High-performance computing (HPC) and machine learning demonstrated in flight using agile condor[C]//2018 IEEE High Performance Extreme Computing Conference (HPEC). Piscataway: IEEE, 2018: 1-4.

[5] 段海滨, 申燕凯, 赵彦杰, 等. 2019年无人机热点回眸[J].

科技导报, 2020, 38(1): 3-5.
DUAN H B, SHEN Y K, ZHAO Y J, et al.Review of technological hotspots of unmanned aerial vehicle in 2019[J]. Science & Technology Review, 2020, 38(1): 3-5.(in Chinese)

[6] 陈鹏, 宋愿赟, 李文静, 等.临近空间高速侦察与监视载荷技术研究综述[J].战术导弹技术, 2021, (1): 7-12.
CHEN P, SONG Y Y, LI W J, et al.Review of high speed reconnaissance and surveillance payload technology in near space[J].Tactical Missile Technology, 2021(1): 7-12. (in Chinese)

[7] KESSLER C W, KELLER J.Optimized mapping of pipelined task graphs on the Cell BE[C]//Proceedings of the 14th International Workshop on Compilers for Parallel Computers (CPC-2009). Piscataway: IEEE, 2009: 1-7.

[8] 曹倩, 胡长军, 李士刚.Cell 异构多核处理器上流水并行优化技术[J].计算机应用研究, 2011, 9(28): 3344-3347.
CAO Q, HU C J, LI S G. Cell heterogeneous multicore pipeline parallel optimization techniques[J]. Application Research of Computers, 2011, 9(28): 3344-3347. (in Chinese)

[9] 李士刚, 胡长军, 王珏, 等.异构多核上多级并行模型支持及性能优化[J]. 软件学报, 2013, 24(12): 2782-2796.
LI S G, HU C J, WANG J, et al. Support for multi-level parallelism on heterogeneous multi-core and performance optimization[J]. Journal of Software, 2013, 24(12): 2782-2796. (in Chinese)

[10] 杨平平, 岳春生, 胡泽明.异构信号处理平台中层次性流水线调度算法[J]. 计算机工程, 2018, 44(11): 83-89.
YANG P P, YUE C S, HU Z M. Multi-level pipeline scheduling algorithm in heterogeneous signal processing platform[J]. Computer Engineering, 2018, 44(11): 83-89. (in Chinese)

[11] 于俊清, 张维维, 陈文斌, 等. 面向多核集群的数据流程序层次流水线并行优化方法[J]. 计算机学报, 2014, 37(10): 2071-2083.
YU J Q, ZHANG W W, CHEN W B, et al. Multi-level pipelining parallelism for dataflow programs on multi-core cluster[J]. Chinese Journal of Computers, 2014, 37(10): 2071-2083. (in Chinese)

[12] OH C, YI S, YI Y.Real-time face detection in full HD images exploiting both embedded CPU and GPU[C]//2015 IEEE International Conference on Multimedia and Expo (ICME). Piscataway: IEEE, 2015: 1-6.

[13] 韩玉艳, 攻敦为, 桑红验, 等. 基于进化优化的多目标批量流水线调度[M]. 北京: 科学出版社, 2018.
HAN Y Y, GONG D W, SANG H Y, et al. Multi-Objective Batch Pipeline Scheduling Based on Evolutionary Optimization[M]. Beijing: Science Press, 2018. (in Chinese)

[14] DEFERSHA F M, CHEN M. Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem[J]. International Journal of Advanced Manufacturing Technology, 2012, 62: 249-265.

[15] LI J Q, PAN Q K, LIANG Y C. An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems[J]. Computers & Industrial Engineering, 2010, 59(4): 647-662.

[16] PAN Q K , TASGETIREN M F , SUGANTHAN P N , et al. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem[J]. Computer Engineering & Applications, 2011, 181(12): 2455-2468.

[17] MARIMUTHU S , PONNAMBALAM S G , JAWAHAR N .Threshold accepting and ant-colony optimization algorithms for scheduling m-machine flow shops with lot streaming[J]. Journal of Materials Processing Technology, 2009, 209(2): 1026-1041.

[18] TSENG C T, LIAO C J. A discrete particle swarm optimization for lot-streaming flowshop scheduling problem[J]. European Journal of Operational Research, 2008, 191(2): 360-373.

[19] 桑红燕, 潘全科, 武磊, 等. 批量流水线调度问题的混合差分进化算法[J]. 计算机工程与应用, 2010, 46(21): 47-50.
SANG H Y, PAN Q K, WU L, et al. Effective hybrid differential evolution algorithms for lot-streaming flowshop scheduling problem[J]. Computer Engineering and Applications, 2010, 46(21): 47-50. (in Chinese)

[20] 韩栋, 周聖元, 支天, 等. 智能芯片的评述和展望[J]. 计算机研究与发展, 2019, 56(1): 7-22.
HAN D, ZHOU S Y, ZHI T, et al. A survey of artificial intelligence chip[J]. Journal of Computer research and development, 2019, 56(1): 7-22. (in Chinese)

[21] 尹首一, 郭珩, 魏少军. 人工智能芯片发展的现状及趋势[J]. 科技导报, 2018, 36(17): 45-51.
YIN S Y, GUO H, WEI S J. Present situation and future trend of artificial intelligence chips[J]. Science & Technology, 2018, 36(17): 45-51. (in Chinese)

[22] 中科寒武纪公司. MLU100 简介[EB/OL]. (2019-07-11)[2023-03-30]. https://forum.cambricon.com/.

[23] 天津飞腾公司. FT-2000/4 系列处理器数据手册[EB/OL]. (2022-06-06)[2023-03-30]. https://www.phytium.com.cn/

class/38?page=2.
[24] RAPIDIO. RapidIO Specification 2.2[EB/OL]. (2011-05-01) [2023-03-30]. https://rapidio.org/files/RapidIO_Rev_2.2_Specification.zip.
[25] RAPIDIO. RapidIO Specification 4.1[EB/OL]. (2017-07-01)[2023-03-30]. http://rapidio.wpengine.com/wp-content/uploads/2018/06/RapidIO-Specification-4-1.pdf.
[26] 中科寒武纪公司. 寒武纪端云一体人工智能开发平台Cambricon Neuware白皮书[EB/OL]. (2019-08-13)[2023-03-30]. https://www.cambricon.com/.
[27] 中科寒武纪公司. 寒武纪软件开发环境[EB/OL]. (2019-07-11)[2023-03-30]. https://forum.cambricon.com/.
[28] YANN L C. Deep learning hardware: Past, present, and future[C]//Proceedings of 2019 IEEE International Solid-State Circuits Conference (ISSCC). Piscataway: IEEE, 2019: 12-19.
[29] BARNELL M, RAYMOND C, et al. Model quantization and synthetic aperture data analyses increasing throughput and energy efficiency[C]//2021 IEEE High Performance Extreme Computing Conference (HPEC). Piscataway: IEEE, 2021: 1-5.
[30] JOCHER G.YOLOv5 release v6.1[EB/OL]. (2022-08-21) [2023-03-30]. https://github.com/ultralytics/yolov5/releases/tag/v6.1.
[31] LI C Y, LI L L, GENG Y F, et al. YOLOv6 v3.0: A full-scale reloading[EB/OL]. (2023-01-23) [2023-03-30]. https://arxiv.org/abs/2301.05586.
[32] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[EB/OL]. (2022-07-06) [2023-03-30]. https://doi.org/10.48550/arXiv.2207.02696.
[33] JOCHER G. Ultralytics YOLOv8[EB/OL]. (2022-04-18) [2023-03-30]. https://github.com/ultralytics/ultralytics.
[34] GE Z, LIU S T, WANG F, et al. Yolox: Exceeding yolo series in 2021[EB/OL]. (2021-07-18)[2023-03-30]. https://doi.org/10.48550/arXiv.2107.08430.

作者简介

赵二虎 男,1985年生,河北邢台人. 现为中国科学院计算技术研究所高级工程师、硕士生导师,中国科学院大学博士研究生. 主要研究方向为嵌入式智能计算系统、专用计算机硬件、芯片微体系结构等.
E-mail: zhaoerhu@ict.ac.cn

吴济文 男,1987年生,江西上饶人. 现为中国科学院计算技术研究所高级工程师. 主要研究方向为嵌入式智能计算系统的软硬件协同优化.
E-mail: wujiwen@ict.ac.cn

肖思莹 女,1987年生,江西吉安人. 现为中国科学院计算技术研究所工程师. 主要研究方向为嵌入式智能计算系统软件.
E-mail: xiaosiying@ict.ac.cn

晋振杰 男,1991年生,河北邯郸人. 现为中国科学院计算技术研究所助理工程师. 主要研究方向为嵌入式智能计算系统的AI模型优化加速与部署.
E-mail: jinzhenjie@ict.ac.cn

徐勇军 男,1979年生,安徽安庆人. 现为中国科学院计算技术研究所正研级高级工程师、博士生导师、专项技术研究中心主任. 主要研究方向为数据智能.
E-mail: xyj@ict.ac.cn