

针对大小核边缘设备的高效视频分析

李雅馨, 袁景凌, 钟欣

(武汉理工大学计算机与人工智能学院交通物联网技术湖北省重点实验室, 湖北 武汉 430070)

摘要: 视频分析通过从视频流中提取高价值信息, 在智能交通和公共安全等应用中发挥着重要作用。传统云端视频分析尽管具备强大的计算能力, 但海量视频数据的上传会带来高带宽占用和网络延迟。边缘计算通过将视频数据下沉至摄像头附近以降低网络延迟, 仍面临着两大挑战: 1) 逐帧分析导致重复推理, 而现有帧重用方法无法充分利用历史帧的局部相似性; 2) 核心负载不均, 任务在大小核间分配缺乏实时负载感知。为此, 提出一种面向大小核边缘设备的高效视频分析系统 Vable。设计多历史帧块级帧重用机制, 将视频帧划分为细粒度块, 并通过树形存储结构与基于局部敏感哈希 (LSH) 的相似性匹配, 实现跨帧的高效计算结果复用, 从而显著降低冗余推理开销。同时, 提出核心负载感知的列表式有向无环图 (DAG) 分区算法, 通过实时监测大小核负载状态, 动态分配分析任务, 以平衡计算与通信开销, 避免负载失衡导致延迟增加, 实现 Vable 的系统原型, 在两个真实数据集上进行了实验评估。实验结果表明, 在 UA_V₁ 中相较于延迟最高的 WiseFuse, Vable 可将端到端延迟降低 59.23%, 在 UA_V₂ 中相较于 Faastlane, Vable 可将端到端延迟降低 45.83%, 在保持高吞吐量的同时有效降低端到端延迟。

关键词: 视频分析; 视频帧重用; 任务调度; 大小核架构; 边缘计算

中图分类号: TP391

文献标志码: A

DOI: 10.19678/j.issn.1000-3428.0253462

Efficient Video Analytics for Big.LITTLE Edge Devices

LI Yaxin, YUAN Jingling, ZHONG Xian

(Hubei Provincial Key Laboratory of Transportation Internet of Things Technology,

School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan 430070, Hubei, China)

【Abstract】 Video analytics extracts high-value information from video streams and plays a crucial role in applications such as intelligent transportation and public safety. Although traditional cloud-based video analytics offers powerful computational capabilities, uploading massive amounts of video data results in high bandwidth consumption and network latency. Edge computing reduces network latency by processing video data near cameras; however, it still faces two major challenges. First, frame-by-frame analysis leads to redundant inference, and existing frame reuse methods cannot fully exploit local similarities in historical frames. Second, uneven core workload arises because task allocation across big LITTLE lacks real-time load awareness. To address these issues, this study proposes Vable, an efficient video analytics system for big.LITTLE edge devices. It employs a multihistorical frame, block-level frame reuse mechanism, which partitions video frames into fine-grained blocks and employs a tree-based storage structure combined with Locality-Sensitive Hashing (LSH) for similarity matching, enabling efficient cross-frame computation reuse and significantly reduces redundant inference overhead. Additionally, Vable introduces a core workload-aware list-based Directed Acyclic Graph (DAG) partitioning algorithm that dynamically allocates analysis tasks by monitoring the real-time load of big LITTLE, thus balancing computation and communication overhead while avoiding latency increases caused by load imbalance. A prototype of Vable is implemented and evaluated using two real-world datasets. Experimental results demonstrate that Vable reduces the end-to-end latency by 59.23% in UA_V₁ compared to WiseFuse (the highest-latency baseline) and by 45.83% in UA_V₂ compared to Faastlane, effectively lowering the end-to-end latency while maintaining high throughput.

【Key words】 video analytics; video frame reuse; task scheduling; big.LITTLE architecture; edge computing

0 引言

视频分析通过目标检测、运动跟踪和图像分割等任务, 从连续视频流中提取对场景理解具有重要

意义的结构化信息, 已广泛应用于自动驾驶、智能交通管控、公共安全监控等领域^[1-2]。通过分析交通流量、识别异常事件或预测潜在风险, 视频分析在城市治理和智慧交通体系中发挥着关键支撑作用。例

基金项目: 国家自然科学基金(62472332, 62271361)。

作者简介: 李雅馨(CCF 专业会员), 女, 硕士研究生, 主研方向为边缘计算; 袁景凌(通信作者)、钟欣, 教授、博士。

收稿日期: 2025-12-23

修回日期: 2026-03-09

E-mail: yjl@whut.edu.cn

如,在智能交通场景中,视频分析可支持碰撞预警、车牌识别和行人跟踪等功能,有效提升道路安全性和通行效率^[3-4]。

现有视频分析系统多采用云计算范式,即由摄像头或本地设备通过广域网将原始帧或初步处理后的视频数据上传至远程服务器进行集中分析。尽管云端具备强大的计算能力,但每日 TB 级的视频数据传输会消耗大量带宽,显著增加部署与运维成本,并易引发网络拥塞。在山区、隧道或乡村道路等网络条件受限的场景中,链路中断与抖动将进一步恶化分析时延和系统响应能力。带宽瓶颈与低时延需求共同推动了视频分析从云端向边缘侧迁移^[5-6]。

边缘计算通过将处理任务下沉至靠近摄像头的位置,有效降低了网络负载并缩短了端到端延迟。尽管图形处理器(GPU)、神经网络处理器(NPU)等专用加速器在性能上具有优势,但在轻量级边缘环境中,基于通用 CPU 的推理仍然不可或缺,其在系统集成性和软件兼容性方面更具优势^[7]。当前主流边缘处理器多采用非对称多处理器(AMP)架构^[8-9],在同一片上系统(SoC)中集成高性能大核与低功耗小核,以兼顾吞吐量与能效。

然而,在大小核架构的边缘设备上直接部署视频分析仍面临显著挑战。首先,逐帧的视频分析通常涉及高计算量,在资源受限的边缘环境中难以满足延迟要求。降低视频分析计算负载的一个常见方法是帧重用^[10],即复用已计算帧的结果以避免重复推理。现有的帧重用方法通常假设相邻帧具有最高相似度,仅保留最近帧的计算结果用于帧重用^[11-12]。但在实际视频场景中,这一假设并不总是成立,历史帧中可能存在更高的相似度。同时,当前帧的局部可能仍与历史帧高度相似,仅以整帧或背景区域作为重用单位,可能显著减少可重用的规模。此外,随着保留的历史帧增加,由于缺乏合适的存储结构,延迟反而会增加。因此,亟需一种高效的跨帧计算结果重用机制。其次,视频分析的时延不仅取决于视频分辨率和数据规模,还与任务在大小核之间的分配方式密切相关。现有边缘侧视频分析系统的调度策略通常依据核类型或固定划分规则来映射计算任务。这类方法在短时或稳定的场景下能够工作,但在边缘持续运行的视频分析过程中,核心负载会随任务并发、系统负载等因素发生动态变化。若调度决策未能感知核心负载变化,任务可能持续被分配至高效核心,导致负载失衡并增加整体计算延迟。因此,设计一种能够感知核心负载变化进行动态调

度的策略同样至关重要。

针对上述问题,本文提出一种面向大小核边缘设备的高效视频分析系统 Vable,旨在降低视频分析的整体时延,满足辅助驾驶、交通信号控制和安全监控等对时延高度敏感的应用需求。本文的主要贡献包括以下 3 个方面:

1)提出一种多历史帧块级帧重用算法,用于减少视频分析中的冗余计算。该算法从多个历史帧中基于内容相似性选择可复用结果,并通过块级重用方法扩大可重用区域。为适应边缘设备的资源限制,该算法设计了面向块级重用的树形存储结构,并结合基于局部敏感哈希(LSH)的相似性匹配进行高效检索。

2)提出了一种核心负载感知的列表式有向无环图(DAG)分区算法,用于减少视频分析的整体延迟。该算法实测了核心压力与函数计算延迟的关系,并对视频分析流程进行建模,构建了函数在不同核心上执行的代价模型。在调度过程中,该算法监测核心的实时负载状态,基于最早完成时间表动态地对 DAG 进行分区,避免核心负载失衡,并平衡了计算开销与数据通信成本。

3)实现了 Vable 的系统原型,并在 2 个真实的视频数据集上进行了实验评估。与 6 种现有的视频分析系统和调度方案相比,Vable 在保持最高吞吐量的同时,将整体延迟分别降低了 59.23% 和 45.83%,充分验证了所提出的算法与设计的系统在实际大小核边缘设备上的有效性。

1 相关工作

1.1 视频分析系统

早期的视频分析系统主要构建于云计算环境中。JIANG 等^[13]提出了面向大规模视频分析场景的云端系统 Chameleon,依托充足的云端计算资源,在保证分析精度的同时权衡计算开销与处理延迟。ZHANG 等^[14]提出了 VideoStorm,该系统运行于大规模集群之上,通过在线分析器与离线分析器协同工作,以提高分析质量并降低实时处理延迟。然而,这类云端方案通常需要传输大量原始视频数据,带来了显著的通信开销。

为缓解网络带宽压力,边云协作视频分析逐渐成为研究热点。WANG 等^[15]提出任务驱动的边云协同实时视频分析系统 VaBUS,从视频内容和上下文特征出发,仅将高置信度的感兴趣区域上传至云端,从而降低带宽消耗。YANG 等^[16]提出了联合感知视频处理架构 JAVP,通过多尺度复杂度感知

模型预测任务复杂度,并结合自适应资源感知优化算法显著提升系统吞吐量。随着研究的深入,函数即服务(FaaS)架构因其高扩展性和灵活性受到关注。ZHANG 等^[17]提出了基于无服务器流水线的云边协同视频分析系统 CEVAS,论证了 FaaS 在细粒度资源划分和最小依赖性方面的优势,通过在边缘与云端对工作负载进行细粒度拆分,有效降低云端成本和数据传输开销。PENG 等^[18]进一步提出了无服务器高分辨率边云协作视频分析系统 Tangram,支持对感兴趣区域的动态批处理与拼接,并设计了在线 SLO 感知批处理算法,以降低带宽与计算成本。

随着边缘计算能力的提升,以及对低时延和隐私保护需求的增长,纯边缘视频分析系统逐步发展。HUNG 等^[19]提出 VideoEdge,在多层集群中综合考虑多资源约束,通过识别潜在的“帕累托带”缩小配置搜索空间,在资源利用和分析精度之间取得平衡。HU 等^[20]提出了基于双图像现场可编程门阵列(FPGA)的实时边缘视频分析系统 Gemini,利用 FPGA 图像切换时间对中央处理器(CPU)与 GPU 图像预配置,在时间维度上实现异构资源的弹性重用,从而显著提升分析精度。

Vable 针对大小核架构的边缘设备进行优化,通过有效视频帧重用并引入适配大小核特性的调度机制,有效减少总时延,为在大小核边缘架构上部署视频分析提供了一种高效可行的解决方案。

1.2 视频帧重用

视频帧重用旨在减少重复推理,从而提升视频分析的整体效率。XU 等^[21]提出的支持大规模存档视频快速访问的系统 VStore,将分析需求反向传递至检索、存储和摄取环节,加速视频查询速度。DAUM 等^[22]提出基于分块的视频数据存储管理器 TASM,利用现代视频编解码器的分块特性,实现对编码视频的空间随机访问。这两者主要从视频数据管理的角度出发,侧重于离线视频分析场景,通过优化访问路径或存储布局来提升吞吐量,对时延的敏感性较低,难以直接适用于对实时性要求较高的边缘侧视频分析。CICEK 等^[23]提出了一种以重用为中心的通用卷积神经网络加速器,通过挖掘图像块间的相似性,节省了计算时间和能耗。该方法从硬件加速的角度验证了基于内容相似性的重用价值,但其假设算力和存储空间相对充足,使其难以应用于资源有限的大小核边缘侧。

Vable 关注资源受限的大小核边缘设备上的在

线视频分析场景。在低时延的约束下,Vable 考虑了历史帧相似与局部相似、多帧的存储与重用结果的高效检索,从而实现面向低延迟需求的大小核边缘设备视频分析的计算结果重用。

1.3 任务调度

在任务调度方面,LIU 等^[24]提出了面向卷积神经网络 CPU 推理的 NeoCPU,通过静态核选择将计算映射到 SoC 中的不同 CPU 核心。KOTNI 等^[25]提出了无服务器调度系统 Faastlane,将工作流函数作为线程运行于单个容器实例中,以降低函数间交互延迟。MAHGOUB 等^[26]进一步提出了 WiseFuse,该系统根据用户指定的延迟目标,自动融合顺序函数并绑定并行函数,以优化端到端时延。李成华等^[27]提出了一种视频智能分析任务的动态调度方法,将调度问题建模为二维多重背包问题,并通过动态规划协调边缘侧空闲计算资源,实现算力协同并最大化安全价值收益。

与上述方法不同,Vable 用核心工作负载感知的调度策略,根据实时核心负载和性能配置文件,将视频分析函数动态分配至最合适的执行核心,在大核与小核之间平衡计算与通信开销,从而最大限度地降低整体处理时延。

2 背景与动机

2.1 视频分析

视频分析通常采用管道式结构,由一系列级联函数组成,每个函数负责执行特定的视频处理步骤,例如感兴趣区域选取、候选框生成和特征提取等。根据不同应用需求,这些函数可以灵活组合与重排,从而构建适配多种应用场景的分析管道,实现高效且可扩展的视频分析。

2.2 大小核上的计算延迟恶化

为刻画大小核架构下计算延迟的变化特性,本文选取一个具有代表性的资源敏感型视频分析函数,并将其部署在主频为 2.54 GHz 的大核 Cortex-A77 上执行。同时,使用 stress-ng 工具向该核心施加不同程度的压力,以模拟核心负载的动态变化。需要指出的是,特征提取本身产生的计算负载也被计入核心的整体压力。

如图 1 所示(彩色效果见《计算机工程》官网 HTML 版,下同),特征提取的计算延迟随着核心压力的增加而持续上升。在压力较低阶段,延迟增长相对平缓;当压力超过某一阈值后,延迟迅速攀升,呈现出近似指数式的增长趋势。本文将该阈值称为“转折点”,用于表征核心压力开始引发性能急剧下

降的临界位置。图 1 同时给出了其余视频分析函数的统计结果,其计算延迟同样随核心压力增加而表现出相似的上升规律。低功耗小核在压力不断增大的情况下也会遭遇类似的性能瓶颈。在大小核架构中处理复杂的视频分析负载时,核心间显著的计算

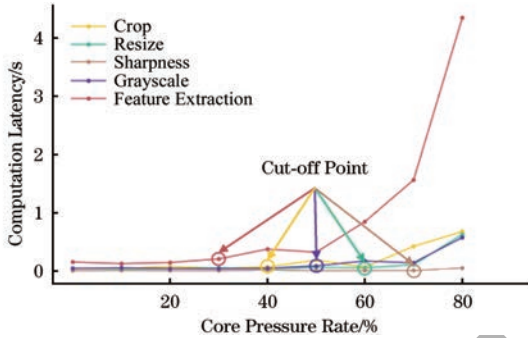


图 1 计算延迟随核心压力上升而增加
Fig.1 Computation latency increases with core pressure

能力差异容易导致调度失衡。若调度策略忽略核心的实时负载状态,仅依据核类型进行粗粒度分配,任务可能会持续集中于某一核心,从而引发计算延迟的急剧恶化。对于时延敏感的边缘视频分析场景而言,由调度不均衡带来的延迟退化是不可接受的。

3 系统架构

Vable 的整体架构如图 2 所示,主要由重用模块和分区模块两部分组成。重用模块负责在视频帧中识别并匹配相似的图像块,以支持计算结果复用;分区模块负责对视频分析管道进行划分与调度。智能摄像头首先将采集的原始视频发送至重用模块,后者将视频帧切分为若干图像块,并插入树结构中进行存储,同时通过基于 LSH 的匹配方法搜索帧内及跨帧的相似块。

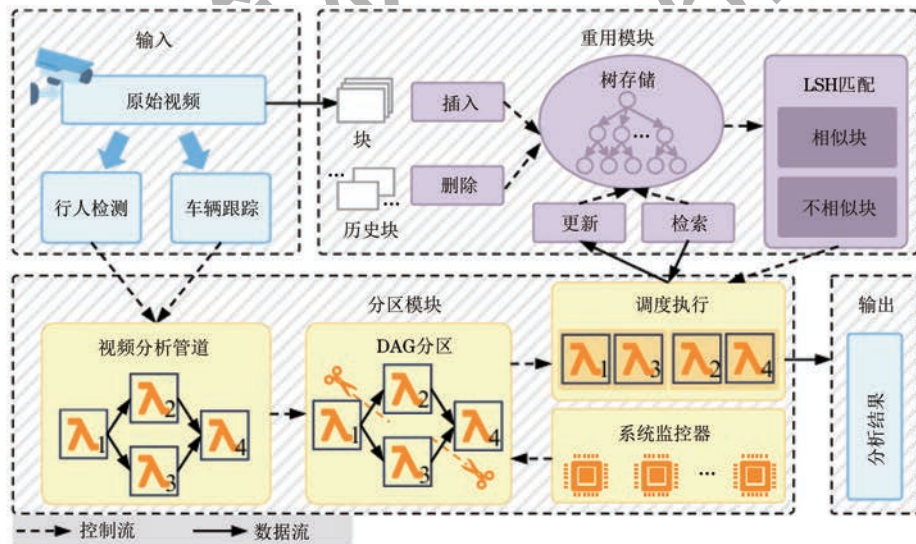


图 2 Vable 系统架构

Fig.2 Architecture of the Vable system

与此同时,摄像头将视频分析需求提交给分区模块。分区模块将该需求解析为对应的视频分析管道,即 DAG,并依据系统监控器中各核心的实时状态对 DAG 进行分区调度。随后,重用模块将获得的块级相似性信息反馈给分区模块,辅助其制定执行决策。

在函数执行阶段,分区模块按照分区结果依次调度并执行 DAG 中的各个函数。对于不相似的图像块,系统进行必要的重计算,并将所有块的计算结果返回至重用模块以更新树结构;对于已匹配的相似块,重用模块直接检索并复用其历史计算结果,并与重计算得到的不相似块结果进行拼接,再传回分区模块作为下一个函数的输入。该过程沿 DAG 逐层推进,直至最后一个函数完成计算,从而完成一次

完整的视频分析流程。

4 视频帧重用优化

为有效解决视频分析中的重复计算问题,本节提出一种视频帧重用优化机制,旨在高效复用相似帧的函数计算结果。传统的帧重用方法通常假设相邻帧具有最高相似性,仅保留相邻帧的计算结果,但这一假设在复杂动态场景中并不总是成立。此外,现有方法往往缺乏合适的存储结构来系统性地保留视频帧及其对应的计算结果。如图 3 所示,所提出的优化机制将视频帧切分为不同粒度的图像块,并构建树结构进行存储,结合基于 LSH 的区域匹配算法在树中高效搜索相似块的函数结果,在减少后续推理计算量的同时保证推理准确性。

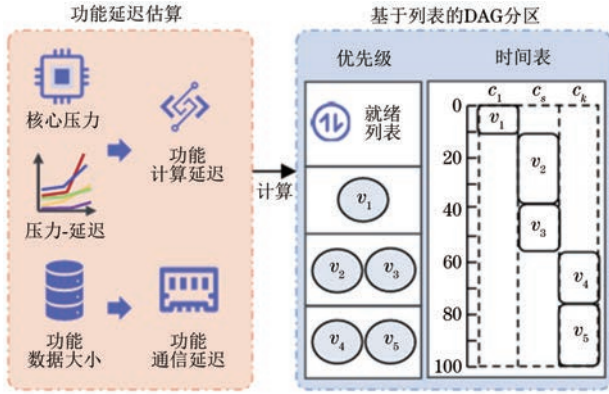


图 4 基于列表的 DAG 分区算法

Fig.4 List-based DAG partitioning algorithm

每个顶点表示管道中的一个视频分析函数,边集 E 中的每条边 $(v_i, v_j) \in E$ 表示函数 v_i 与 v_j 之间的依赖关系。对于存在多个入口或出口节点的 DAG,引入虚拟入口或出口节点,并赋予其零计算权重和零通信代价。记核心集合为 $C = \{c_1, c_2, \dots, c_m\}$,其中 c_k 表示大小核架构中的一个核心。

5.2 问题建模

在视频分析管道的执行过程中,每个函数依次产生计算延迟和通信延迟。函数的计算延迟受核心类型及其当前负载状态的共同影响。核心 c_k 的压力 o_k 可表示为:

$$o_k = 1 - \frac{R_k - r_i}{R_k^t} \quad (3)$$

式中: R_k^t 表示核心 c_k 的总计算资源; R_k 表示其当前可用的空闲计算资源; r_i 表示函数 v_i 所需的计算资源。设 o_k^p 为延迟上升的“转折点”,根据第 2.2 节中实测得到的核心压力-计算延迟曲线,分别用 $t_i^l(c_k)$ 和 $t_i^r(c_k)$ 表示转折点前后函数的计算延迟。当函数 v_i 被分配至核心 c_k 上执行时,其计算延迟 $t_i(c_k)$ 可表示为:

$$t_i(c_k) = t_i^l(c_k) [o_k \leq o_k^p] + t_i^r(c_k) [o_k > o_k^p] \quad (4)$$

不同核心上运行的函数通过 SoC 内部的共享内存进行数据交互。由于通信完全发生在芯片内部,不涉及跨设备网络传输,其通信开销主要由数据量决定。函数 v_i 与 v_j 之间的通信延迟 $t_{i,j}$ 可表示为:

$$t_{i,j} = \alpha |v_i, v_j| \quad (5)$$

式中: α 表示单位数据的传输时间; $|v_i, v_j|$ 表示函数间的数据量。

为保证分配至核心 c_k 的函数总负载不超过其可用资源,需要满足以下资源约束:

$$\sum_{v_i \in V} r_i \leq R_k \quad (6)$$

给定 G 和 C , 视频分析的总时延由 DAG 中的关键路径决定。在满足资源约束的前提下, Vable 的调度目标是 minimized 视频分析的端到端总时延:

$$T = \min \sum_{v_i, v_j \in P} [t_i(c_k) + t_{i,j}] \quad (7)$$

5.3 基于列表的 DAG 分区算法

Vable 设计了一种基于列表的 DAG 分区算法,并根据核心实时状态进行动态自适应调度。由于函数尚未执行,其实际计算延迟 $t_i(c_k)$ 和通信延迟 $t_{i,j}$ 不可直接获得,因此需要采集核心的实时状态,并依据第 5.2 节中的模型对相关延迟进行估算。

在获得延迟估计值后, DAG 分区策略开始执行,其流程如算法 1 所示。

算法 1 基于列表的 DAG 划分算法

输入 计算延迟 $t_i(c_k)$, 通信延迟 $t_{i,j}$

输出 函数划分结果 P

1. 计算每个函数的乐观成本表 (OCT), 并得到其优先级 rank_{oct} ;
2. 初始化空的就绪列表 ready_list ; // 存放所有可调度 // 的函数节点
3. 按照 rank_{oct} 从大到小将可调度函数加入 ready_list ; // 构建基于优先级的调度列表
4. while (ready_list 非空) // 制定时间表
5. 选取 rank_{oct} 最大的函数 v_i ;
6. for each c_k in C
7. 计算 $\text{EFT}(v_i, c_k)$ 并查找 $\text{OCT}(v_i, c_k)$;
8. 计算 $\text{O}_{\text{EFT}}(v_i, c_k) = \text{EFT}(v_i, c_k) + \text{OCT}(v_i, c_k)$;
9. end
10. 选择使 $\text{O}_{\text{EFT}}(v_i, c_k)$ 最小的核心 c_k ;
11. if $r_i \leq R_k$ // 确定资源限制
12. 将函数 v_i 分配至核心 c_k ;
13. else
14. 选择次小 O_{EFT} 对应的核心;
15. end
16. 更新 ready_list ;
17. end
18. 返回 P

该算法首先根据估算的计算延迟和通信延迟构建乐观成本表。OCT 是一个矩阵,其行表示函数,列表示核心,元素 $\text{OCT}(v_i, c_k)$ 表示函数 v_i 在核心 c_k 上执行时,其后继函数至出口节点的最长最短路径。该值从出口节点开始,自底向上递归计算:

$$\text{OCT}(v_i, c_k) = \max_{v_j \in \text{succ}(v_i)} [\min_{c_s, c_s \in C} \text{OCT}(v_j, c_s) + t_j(c_s) + t_{i,j}] \quad (8)$$

式中: $\text{succ}(v_i)$ 表示函数 v_i 的后继函数集合。函数的优先级通过对其所有 OCT 值取平均获得:

$$\text{rank}_{\text{oct}}(v_i) = \frac{\sum_{k=1}^{|C|} \text{OCT}(v_i, c_k)}{|C|} \quad (9)$$

式中: $|C|$ 表示核心的数量。优先级越高的函数将越早被调度。

在制定执行时间表时,算法依次从就绪列表选取函数,并计算其在核心 c_k 上的最早完成时间($\text{EFT}, T_{\text{EFT}}$)。EFT 同时受函数依赖关系和资源竞争约束,其定义为:

$$T_{\text{EFT}}(v_i, c_k) = \max \left[\max_{v_g \in \text{pred}(v_i)} \{T_{\text{EFT}}(v_g) + t_{g,i}\}, T_{\text{Last}}(c_k) \right] + t_i(c_k) \quad (10)$$

式中: $T_{\text{Last}}(c_k)$ 表示核心 c_k 上前一个函数的完成时间。为选择最优执行核心,算法进一步计算乐观最早完成时间(O_{EFT}):

$$O_{\text{EFT}}(v_i, c_k) = T_{\text{EFT}}(v_i, c_k) + \text{OCT}(v_i, c_k) \quad (11)$$

在满足资源约束的条件下,函数将优先分配至 O_{EFT} 最小的核心。该 DAG 分区算法的时间复杂度为 $O(m(e+n)+n^2)$, 其中 e 表示 DAG 中边的数量。

6 实验

6.1 实验设置

为评估 Vable 的性能,本文实验选用海思麒麟 9000 处理器作为边缘设备。该处理器采用大小核架构,由 1 个 3.13 GHz 的 Cortex-A77 核、3 个 2.54 GHz 的 Cortex-A77 核以及 4 个 2.05 GHz 的 Cortex-A55 核组成,配备 8 GB LPDDR5 内存,并运行基于 Ubuntu 22.04 LTS 和 Linux 6.5.0 内核的系统环境。此外,实验租用了一台阿里云基于倚天 710 ARM 架构 CPU 的 16 核实例 g8y,提供最高 64 Gb/s 的网络带宽,用作云端基线的实验平台。

图 5 给出了实验中采用的一个具有代表性的视频分析管道,该管道由多种视频分析函数级联构成,覆盖交通场景中两类典型任务:图像增强与目标检测。为体现管道结构的多样性,实验基于这些函数构建了多种不同深度的级联管道。

本实验选用 2 个不同类别的数据集对 Vable 进行综合评估。

1) UA-DETRAC^[28]。用于多目标检测的标准数据集,包含 100 段真实交通视频,共计超过 14 万帧以及 121 万个手动标注的车辆边界框。

2) UCSD^[29]。行人检测数据集由高位摄像头采集,场景中人群密度从稀疏到高度拥挤不等。

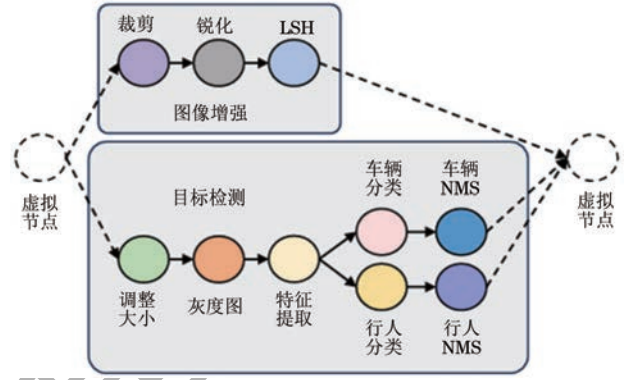


图 5 1 个具有代表性的视频分析管道

Fig. 5 A representative video analytics pipeline

为匹配视频分析管道中目标检测任务的时序特性,本实验将数据集集中的视频划分为多个连续的 10 帧短序列,以模拟资源争用场景,并选取其中一条具有代表性的视频进行统计分析。为提高结果的可靠性,每个数据集分别进行了两组实验,形成 4 个数据集:UA_V₁、UA_V₂ 以及 UCSD_P₁、UCSD_P₂。

本实验将 Vable 与多种纯边缘、纯云及边云协同的视频分析系统或任务调度方案对比,具体包括:

1) VideoEdge^[19]。一种分层式视频分析系统,通过考虑多种资源约束选择模型、参数和部署位置以最大化精度。为保证公平性,本实验实现了其仅在边缘各核心上执行的视频分析变体。

2) CEVAS^[17]。一种边云协同视频分析系统,基于凸包的算法为管道选择合适的边云分区点,以最小化云支出和数据传输开销。

3) Faastlane^[25]。一种纯云任务调度系统,将管道内所有级联函数部署在单一工作节点以减少函数交互延迟,但未考虑单节点负载可能引发的延迟恶化。

4) WiseFuse^[26]。一种纯云调度架构,通过融合串行函数、捆绑相似并行函数并均衡分布不同类型的并行函数,降低总延迟和成本。

5) Rocket^[30]。一种根据系统可用计算资源动态调整模型选择和参数配置的调度方法,在精度与延迟之间取得平衡。

6) E4^[31]。一种边缘视频推理优化系统,通过感知负载和复杂度,动态调整推理路径,降低响应时延。为保证公平性,本实验仅采用其推理路径自适应机制,在固定资源下评估延迟优化。

本实验选取以下 3 个指标,并计算其在不同视频分析管道下的平均值,以全面评估 Vable 与 6 种基线方法的性能表现:1) 吞吐量表示单位时间内稳定处理的视频帧数量,用于反映系统在负载波动条件下进行视频分析的资源利用效率;2) 总延迟指视频分析从输入到输出的端到端执行时间,用于验证

系统的低延迟目标实现效果;3)计算延迟与通信延迟分别代表本地计算与数据传输的开销,用于验证重用算法在减少冗余计算与数据传输方面的有效性,以及调度算法在平衡计算与通信开销中的作用,从而揭示整体性能提升的来源。

6.2 整体表现

图 6 展示了各方法在 4 组数据集上的平均吞吐量。从图 6 可以观察到, Vable 在所有数据集中均取得了最高的平均吞吐量。这表明 Vable 并非依赖特定负载,而是能够在多种视频分析条件下稳定提升系统处理效率。其中在 UA_V₁ 中, Vable 相较于吞吐量最低的 WiseFuse 提升幅度达 145.69%, 相较于次优方案 CEVAS 仍提升了 19.75%。类似的

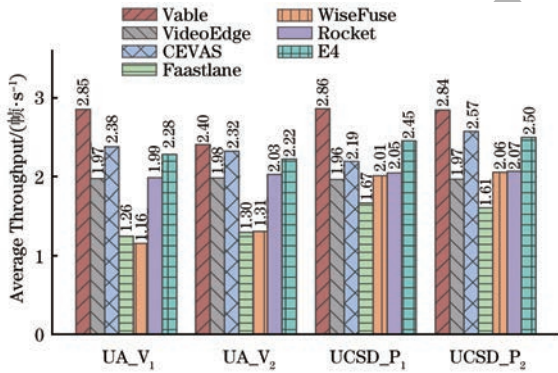


图 6 不同方法在 4 组数据集上视频分析的平均吞吐量

Fig. 6 Average throughput of video analytics for different methods on four datasets

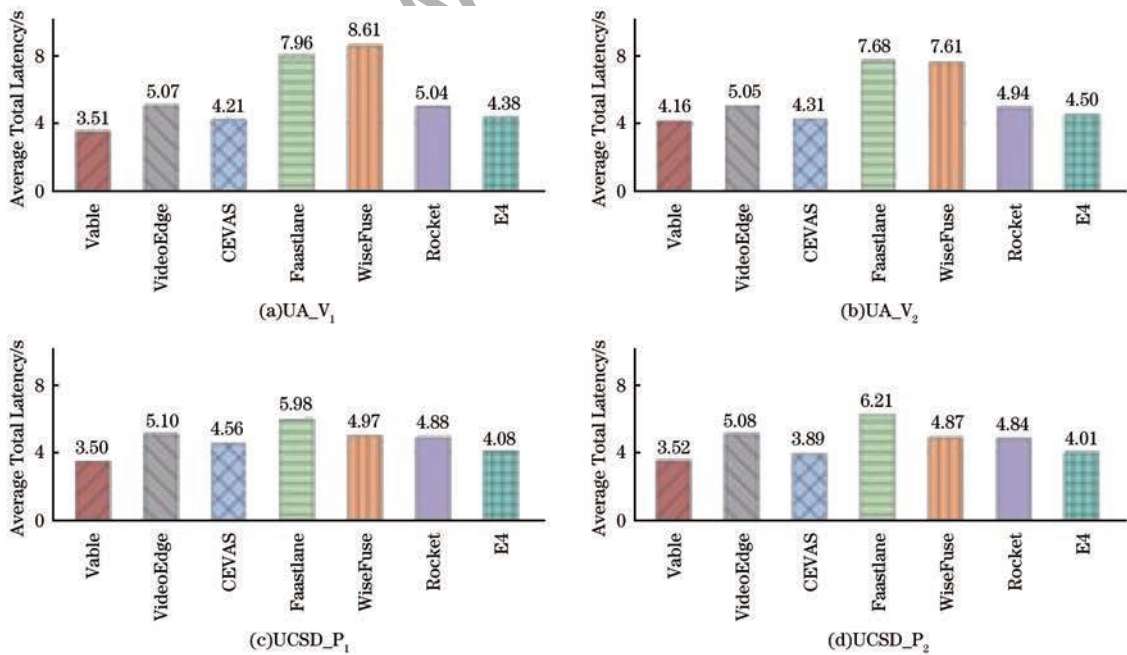


图 7 不同方法在 4 组数据集上视频分析的总延迟

Fig. 7 Total latency of video analytics for different methods on four datasets

为进一步分析不同方法的延迟来源,本实验将总延迟拆分为计算延迟与通信延迟进行对比分

性能差距在其余实验组中均持续存在, Vable 通过其设计机制在保证处理一致性的同时显著提高了系统吞吐量,为低延迟视频分析提供了更具鲁棒性的性能支撑。

图 7 展示了各方法在 4 组数据集上的平均总延迟表现。从图 7 可以看出, Vable 在所有数据集上的总延迟均最低,表明该系统能够在多样化视频内容与负载条件下稳定提供快速响应能力。具体而言,在 UA_V₁ 和 UA_V₂ 中, Vable 的平均总延迟分别为 3.51 s 和 4.16 s,在 UA_V₁ 中相较于延迟最高的 WiseFuse,最大降幅为 59.23%,在 UA_V₂ 中相较于 Faastlane 最大降幅为 45.83%。在 UCSD_P₁ 和 UCSD_P₂ 中, Vable 的延迟分别为 3.50 s 和 3.52 s,相较于延迟最高的 Faastlane,最大降幅为 41.47% 和 43.32%。这一显著差距说明, Vable 能够有效减少纯云方案中由网络传输引入的额外开销。从系统类型角度分析,边缘系统 VideoEdge、Rocket 和 E4 的延迟水平相对稳定,整体处于中上等水平;边云协同系统 CEVAS 虽然在大部分场景下的延迟与 Vable 接近,但存在一定波动。而纯云系统 Faastlane 和 WiseFuse 则表现出较明显的延迟不确定性,反映其受网络传输的高度影响。综合来看,图 7 的结果表明, Vable 能更稳定且有效地降低视频分析的总延迟,更适合对延迟要求较高的视频分析场景。

析。由于各方法的策略制定过程极短,因此对整体延迟影响可以忽略,故不作呈现。如图 8 所示,

Vable 在计算与通信两方面均表现出显著优势,揭示了其具有低延迟性能的主要来源。在计算延迟方面,CEVAS、Faastlane 和 WiseFuse 借助云端算力,整体计算开销低于所有边缘方案。在边缘方案中,Vable 在计算延迟方面表现出显著优势,较 VideoEdge、Rocket 和 E4 分别最高降低了 33.40%、32.26%、和 18.59%。值得注意的是,在 UCSD_P₂ 中,Vable 的计算延迟仅比 Faastlane 高 0.02 s,这表明其通过高效的组织与执行机制有效弥补了边缘算力受限的劣势。在通信延迟方面,Vable 的优势更为突出,其通信延迟在所有数据集中均控制在 0.13~0.20 s。在 4 个数据

集中,最大降幅分别达到 97.55%、96.04%、95.77%和 95.77%,显著降低了通信开销。VideoEdge 和 Rocket 通信延迟虽然较低,但仍明显高于 Vable;E4 在一定程度上同时降低了计算与通信延迟,但整体效果仍不及 Vable。Faastlane 和 WiseFuse 由于需要将完整视频上传至云端,导致通信延迟极高,成为打破其整体性能的主要瓶颈。CEVAS 通过仅上传中间结果,在一定程度上缓解了通信压力。综合来看,Vable 不仅在边缘侧显著降低了计算开销,通过最小化必要的数据传输有效压缩了通信延迟,从而在整体上实现了对端到端延迟的系统性优化。

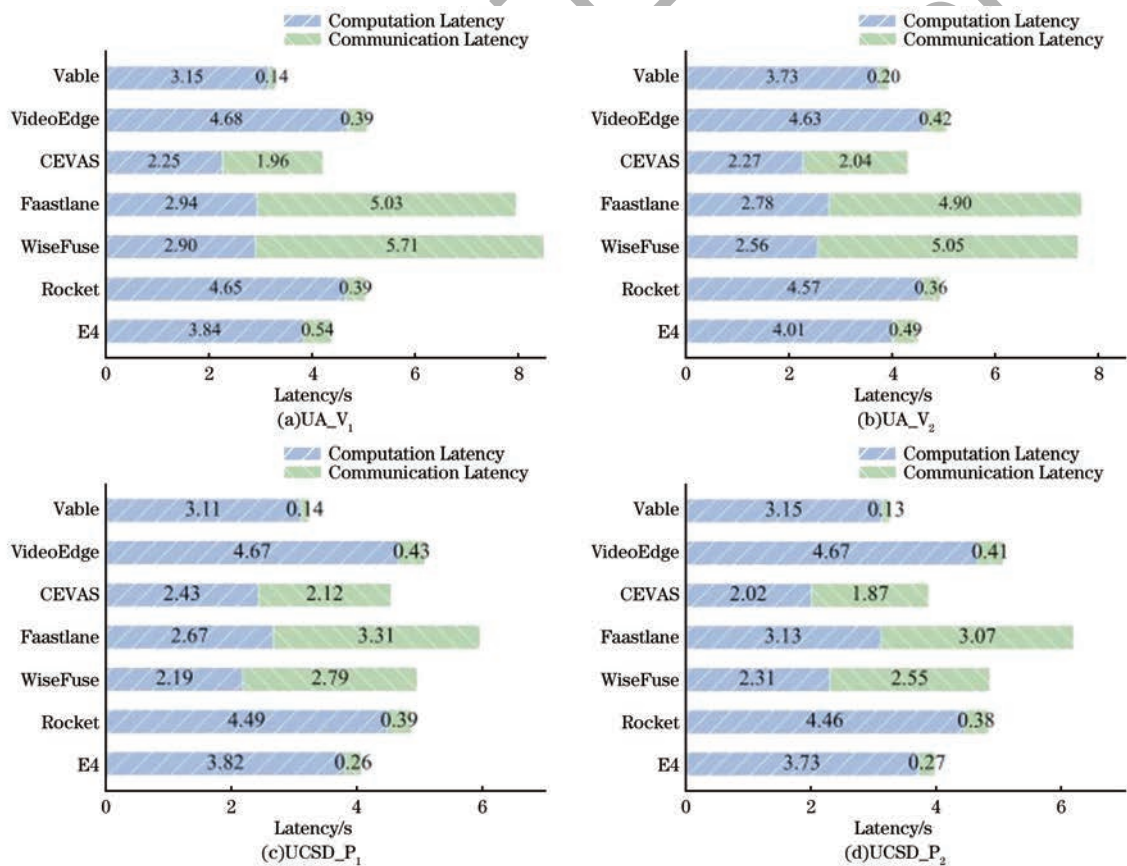


图 8 不同方法在 4 组数据集上视频分析的计算延迟和通信延迟对比

Fig.8 Comparison of video analytics computation latency and communication latency for different methods on four datasets

6.3 算法开销

在第 6.2 节中,通过拆分平均总延迟可以发现,DAG 分区算法的平均开销不足 0.3 s,对视频分析的总延迟影响甚微。由于大小核边缘设备中的核心数通常是一个较小的常数,因此 DAG 分区算法的复杂度主要受其规模的影响。为了评估 DAG 中顶点数量和依赖关系对算法开销的影响,本实验随机生成了不同规模的 DAG,并测试分区算法在这些不同条件下的运行时间。具体来说,本实验逐步增加

DAG 中的顶点数量,同时使用随机数模拟顶点之间的依赖关系,确保依赖关系随着顶点数量的增加而增加。图 9 展示了 DAG 分区算法在不同顶点数量和依赖关系下的运行开销。实验结果表明,当顶点数量不超过 10 个时,算法的运行时间小于 0.01 s,可以忽略不计。当顶点数量增加至 10~20 个时,开销仍保持较低,小于 0.5 s。然而,当顶点数量超过 20 个时,算法开销显著增加,符合预期的时间复杂度增长模式,但仍在可控范围内,最大开销维持在数

秒之内。对于边缘视频分析,请求通常不会过于复杂,这意味着在大多数实际应用中,DAG 分区算法的开销保持在毫秒级,不会造成明显的响应延迟。

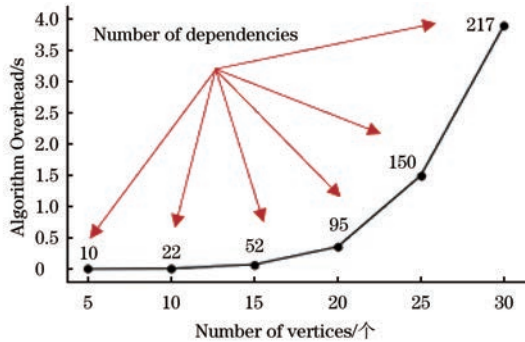


图 9 DAG 分区算法的开销

Fig.9 Overhead of DAG partitioning algorithm

6.4 消融实验

为验证视频帧重用机制与核工作负载感知 DAG 分区策略对 Vable 性能提升的贡献,本文进一步开展了消融实验。图 10 展示了分别移除上述两种设计机制后系统总延迟的变化情况。从图 10 可以看出,去除任一机制都会导致总延迟显著上升。当移除视频帧重用机制时,系统无法复用相邻帧中具有高度相似性的中间结果,导致冗余计算与重复数据传输显著增加,使得计算阶段与通信阶段的开销同步放大,从而使总延迟最大上升 43.02%。当移除核心工作负载感知的 DAG 分区策略时,任务调度退化为对核心异构性的感知不足,多项任务可能被集中分配至同一类核心中,在持续视频流输入的情况下容易引发核心负载失衡,核心达到了压力“转折点”,进而在计算阶段形成瓶颈并放大整体分析延迟,其最大延迟增幅达到 19.09%。上述结果充分表明,视频帧重用优化与核心工作负载感知调度策略分别从减少冗余计算与提升异构资源利用效率的角度,对降低边缘视频分析系统的端到端延迟起到了关键作用。

在讨论 Vable 通过视频帧重用机制降低延迟后,有必要证明它对视频分析系统的精度影响较小。本文进一步在 UA_V₁ 和 UA_V₂ 上采用目标检测领域标准的平均精度均值(mAP)指标进行评价。Vable 的性能变化主要有 2 个原因:1)结果重用引入的近似误差;2)LSH 误匹配带来的误差。表 1 展示了 Vable 和没有使用视频帧重用机制的 mAP 对比结果。与没有使用视频帧重用机制的情况相比,Vable 在 UA_V₁ 和 UA_V₂ 上的 mAP 仅下降了 2.32% 和 5.22%,这种轻微的性能下降是可以接受的,尤其是相比延迟的降低。需要注意的是,该变化

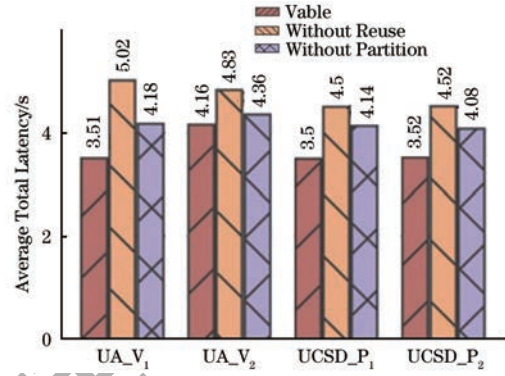


图 10 视频帧重用优化机制和核心工作负载感知

DAG 分区策略对 Vable 总延迟的影响

Fig.10 Impact of video frame reuse optimization mechanism and core work load-aware DAG partitioning strategy on total latency of Vable

幅度是视频帧重用机制对 Vable 性能影响的整体上界,这也侧面反映了 LSH 误匹配对 Vable 整体精度的影响是可控的。

表 1 在 UA_V₁ 和 UA_V₂ 数据集上的检测结果

Table 1 Detection results on UA_V₁ and UA_V₂ dataset %

Method	Dataset	mAP@0.5
Vable	UA_V ₁	71.16
	UA_V ₂	64.33
Without Reuse	UA_V ₁	72.85
	UA_V ₂	67.87

6.5 真实场景的案例分析

为验证 Vable 的普适性与在实际场景中的效果,本节面向真实校园安保巡逻场景开展了一组基于智能终端的视频分析实验。该实验选用谷歌 Pixel 3 智能手机,其搭载骁龙 845 SoC,包含 4 个 Kryo 385 Gold 大核和 4 个 Kryo 385 Silver 小核。实验视频通过设备自带摄像头采集于校园道路场景,分别包含两种典型条件:1)白天光照充足、行人和车辆密集的高目标密度场景;2)傍晚光照不足、交通流量较低的稀疏目标场景。视频拍摄路段长度约为 50 m,单段视频时长约为 20 s。采集视频通过 4 G 网络传输至边缘云集群,以便与基线方案进行对比评估。

表 2 展示了 Vable 与 6 种基线方案在两种场景下的总延迟对比。在目标稀疏的晚间模式下,Vable 的总延迟为 33.64 s,较延迟最高的 Faastlane 降低了 66.2%,相较于次优方案 E4 也降低了 20.21%。在目标密度较高的白天条件下,整体处理负载明显上升,但 Vable 仍能保持稳定的低时延性能,其总延迟较 Faastlane 最高减少了 82%,较次优的 E4 也减少了 25.48%。这一结果表明,即便在高目标

密度导致计算与调度压力显著增加的情况下,Vable 的自适应调度机制仍能够有效协调计算负载,避免延迟随负载增长而急剧恶化。随着视频时长的增加,Vable 与其余方案之间的延迟差距进一步扩大。尤其在真实网络环境下,依赖云端执行的方案均出现明显的延迟累积现象,主要源于持续的远程数据传输开销。综合来看,表 2 结果表明,Vable 所采用的高效帧重用机制与自适应调度策略能够在不同目标密度和运行条件下持续抑制端到端延迟增长,使其更适合无缝集成至智能摄像头、无人机等边缘设备中,支持对时延要求较高的视频分析场景。

表 2 白天和傍晚两种场景下的总延迟对比

Table 2 Comparison of total latency under daytime and after dusk conditions 单位:s

Method	Daytime	After dusk
Vable	82.76	33.64
VideoEdge	125.90	50.63
CEVAS	209.66	88.13
Faastlane	461.56	99.40
WiseFuse	342.65	82.61
Rocket	120.94	48.05
E4	111.06	42.16

7 结束语

本文提出 Vable,一种面向大小核边缘设备的视频分析系统。Vable 通过引入视频帧重用优化机制与核工作负载感知的 DAG 分区算法,有效应对边缘视频分析中的挑战,在提升资源利用效率的同时显著降低端到端延迟。基于两个真实数据集的实验结果表明,Vable 相较于现有先进系统可实现最高 59.23% 的延迟降低。凭借其高效性与可扩展性,Vable 具有良好的工程实用价值,适用于交通等典型的边缘视频分析场景,并为资源受限环境下的视频分析系统设计提供了有益参考。Vable 也为后续研究提供了进一步发展空间。由于重用机制在降低计算开销的同时可能对分析精度产生影响,且边缘设备通常受到能耗约束,因此后续将围绕能耗、延迟与精度间的协同优化展开,从而进一步提升系统在实际部署中的表现。此外,为增强系统对多样化计算资源的适应能力,Vable 还可拓展至更多类型的异构硬件平台,例如融合 CPU、GPU 或 NPU 的边缘设备,以推动其在真实大规模边缘环境中的落地应用。

参考文献

[1] BEWLEY A, GE Z Y, OTT L, et al. Simple online and realtime

tracking[C]//Proceedings of 2016 IEEE International Conference on Image Processing (ICIP). Washington D. C., USA: IEEE Press, 2016: 3464-3468.

- [2] SHELHAMER E, LONG J, DARRELL T. Fully convolutional networks for semantic segmentation [C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2015: 3431-3440.
- [3] 黄玉琦, 杨晓霞, 杨容浩, 等. 面向自动驾驶的多尺度目标检测算法[J/OL]. 计算机工程: 1-13 [2025-12-16]. <https://doi.org/10.19678/j.issn.1000-3428.0252697>.
- [3] HUANG Y Q, YANG X X, YANG R H, et al. Multi-scale object detection algorithm for autonomous driving [J/OL]. Computer Engineering: 1-13 [2025-12-16]. <https://doi.org/10.19678/j.issn.1000-3428.0252697>. (in Chinese)
- [4] 肖进胜, 郭浩文, 谢红刚, 等. 监控视频异常行为检测的概率记忆自编码网络[J]. 软件学报, 2023, 34(9): 4362-4377.
- [4] XIAO J S, GUO H W, XIE H G, et al. Probabilistic memory auto-encoder network for anomaly detection in surveillance video [J]. Journal of Software, 2023, 34(9): 4362-4377. (in Chinese)
- [5] SIPOLA T, ALATALO J, KOKKONEN T, et al. Artificial intelligence in the IoT era: a review of edge AI hardware and software[C]//Proceedings of the 31st Conference of Open Innovations Association (FRUCT). Washington D. C., USA: IEEE Press, 2022: 320-331.
- [6] WANG X B, TANG Z Q, GUO J X, et al. Empowering edge intelligence: a comprehensive survey on on-device AI models[EB/OL]. [2025-11-18]. <https://arxiv.org/pdf/2503.06027>.
- [7] CHENG L, GU Y, LIU Q Z, et al. Advancements in accelerating deep neural network inference on AIoT devices: a survey[J]. IEEE Transactions on Sustainable Computing, 2024, 9(6): 830-847.
- [8] 赵娜, 郝春亮, 翟健, 等. S-Bridge: 性能非对称多核处理器下负载均衡代理机制[J]. 软件学报, 2020, 31(9): 2965-2979.
- [8] ZHAO S, HAO C L, ZHAI J, et al. S-Bridge: CPU load balancing agent for performance asymmetric multicore processors[J]. Journal of Software, 2020, 31(9): 2965-2979. (in Chinese)
- [9] 赵娜, 杨秋松, 李明树. 性能非对称多核处理器下异构感知调度技术[J]. 软件学报, 2019, 30(4): 1164-1190.
- [9] ZHAO S, YANG Q S, LI M S. Heterogeneous-aware scheduling research on performance asymmetric multicore processors[J]. Journal of Software, 2019, 30(4): 1164-1190. (in Chinese)
- [10] GONDIMALLA A, CHESNUT N, THOTTETHODI M, et al. SparTen: a sparse Tensor accelerator for convolutional neural networks[EB/OL]. [2025-11-18]. <https://dl.acm.org/doi/pdf/10.1145/3352460.3358291>.
- [11] HUYNH L N, LEE Y K, BALAN R K. DeepMon: mobile GPU-based deep learning framework for continuous vision applications[EB/OL]. [2025-11-18]. <https://dl.acm.org/doi/pdf/10.1145/3081333.3081360>.
- [12] XU M W, ZHU M Z, LIU Y X, et al. DeepCache: principled cache for mobile deep vision[C]//Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. New York, USA: ACM Press, 2018: 129-144.
- [13] JIANG J C, ANANTHANARAYANAN G, BODIK P, et al. Chameleon: scalable adaptation of video analytics[C]//Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication. New York, USA: ACM Press, 2018: 253-266.
- [14] ZHANG H Y, ANANTHANARAYANAN G, BODIK P, et al. Live video analytics at scale with approximation and delay-tolerance [C]// Proceedings of the 14th USENIX

- Symposium on Networked Systems Design and Implementation. New York, USA: ACM Press, 2017: 377-392.
- [15] WANG H L, LI Q, SUN H Y, et al. VaBUS: edge-cloud real-time video analytics via background understanding and subtraction [J]. IEEE Journal on Selected Areas in Communications, 2022, 41(1): 90-106.
- [16] YANG Z M, JI W, GUO Q, et al. JAVP: joint-aware video processing with edge-cloud collaboration for DNN inference[C]//Proceedings of the 31st ACM International Conference on Multimedia. New York, USA: ACM Press, 2023: 9152-9160.
- [17] ZHANG M, WANG F X, ZHU Y F, et al. Towards cloud-edge collaborative online video analytics with fine-grained serverless pipelines [C]//Proceedings of the 12th ACM Multimedia Systems Conference. New York, USA: ACM Press, 2021: 80-93.
- [18] PENG H S, ZHAN Y F, LI P, et al. Tangram: high-resolution video analytics on serverless platform with SLO-aware batching [C]//Proceedings of the 44th International Conference on Distributed Computing Systems (ICDCS). Washington D.C., USA: IEEE Press, 2024: 645-655.
- [19] HUNG C C, ANANTHANARAYANAN G, BODIK P, et al. VideoEdge: processing camera streams using hierarchical clusters [C]//Proceedings of IEEE/ACM Symposium on Edge Computing (SEC). Washington D.C., USA: IEEE Press, 2018: 115-131.
- [20] HU C, LU R, SANG Q L, et al. An edge-side real-time video analytics system with dual computing resource control [J]. IEEE Transactions on Computers, 2023, 72(12): 3399-3415.
- [21] XU T R, BOTELHO L M, LIN F X Z. VStore: a data store for analytics on large videos [C]//Proceedings of the 14th EuroSys Conference. New York, USA: ACM Press, 2019: 1-17.
- [22] DAUM M, HAYNES B, HE D, et al. TASM: a tile-based storage manager for video analytics[EB/OL]. [2025-11-18]. <https://arxiv.org/pdf/2006.02958>.
- [23] CICEK N M, NING L, OZTURK O, et al. General reuse-centric CNN accelerator [J]. IEEE Transactions on Computers, 2021, 71(4): 880-891.
- [24] LIU Y Z, WANG Y, YU R F, et al. Optimizing CNN model inference on CPUs[C]//Proceedings of 2019 USENIX Annual Technical Conference. Washington D.C., USA: IEEE Press, 2019: 1025-1040.
- [25] KOTNI S, NAYAK A, GANAPATHY V, et al. Faastlane: Accelerating Function-as-a-Service workflows [C]//Proceedings of USENIX Annual Technical Conference. New York, USA: [s.n], 2021: 805-820.
- [26] MAHGOUB A, YI E B, MINOCHA E, et al. WiseFuse: workload characterization and DAG transformation for serverless workflows [J]. Performance Evaluation Review, 2022, 50(1): 57-58.
- [27] 李成华, 石胜涛, 李孝天, 等. 基于边缘算力协同系统的视频智能分析任务动态调度方法 [J]. 电子与信息学报, 2023, 45(12): 4458-4468.
- LI C H, SHI S T, LI X T, et al. Dynamic scheduling method for video intelligent analysis tasks based on edge computing collaborative system [J]. Journal of Electronics and Information Technology, 2023, 45(12): 4458-4468. (in Chinese)
- [28] WEN L Y, DU D W, CAI Z W, et al. UA-DETRAC: a new benchmark and protocol for multi-object detection and tracking [J]. Computer Vision and Image Understanding, 2020, 193: 102907.
- [29] MAHADEVAN V, LI W, BHALODIA V, et al. UCSD anomaly detection dataset [EB/OL]. [2025-11-18]. <http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm>.
- [30] Microsoft Research. Microsoft rocket for live video analytics. [EB/OL]. [2025-11-18]. <https://www.microsoft.com/en-us/research/project/live-video-analytics/>.
- [31] ZHANG Z, ZHAO Y, CHANG M C, et al. E4: energy-efficient DNN inference for edge video analytics via early exiting and DVFS [EB/OL]. [2025-11-18]. <https://arxiv.org/pdf/2503.04865>.

文字编辑 薛晋栋
栏目编辑 宋 圆