中国科学: 信息科学 2023 年 第 53 卷 第 1 期: 46–65
SCIENTIA SINICA Informationis
论文

面向多设备协同场景的实时视频流分析系统

杨铮*, 董亮, 蔡新军

清华大学软件学院, 北京 100084
* 通信作者. E-mail: hmilyyz@gmail.com

收稿日期: 2021–05–22; 修回日期: 2021–08–30; 接受日期: 2022–01–13; 网络出版日期: 2023–01–11

摘要 实时视频流分析在智能监控、智能制造、自动驾驶等场景中具有重要价值, 然而其存在计算负载高、带宽需求大和延迟要求严格等特点, 难以通过传统的本地计算模式或者云计算模式进行部署. 近年兴起的边缘计算范式, 将复杂的计算任务从终端设备上传到物理临近的边缘服务器上, 能够有效解决设备层面的部署问题. 然而, 例如无人机编队飞行、车队自动驾驶和多机器人协同等不断涌现的多设备协同场景, 新增了系统层面的综合性能要求, 包括智能分析的实时准确率、设备之间的性能一致性和系统容纳的设备数量上限. 当前的边缘计算范式对多设备协同场景的优化尚显不足, 未能有效解决设备之间对上传带宽和服务器算力的竞争问题, 所以难以满足这类场景的要求. 本文设计了 MASSIVE 系统, 能够在多设备协同场景中, 全面提升实时视频分析的综合性能. 首先, MASSIVE 系统提出了适合多设备协同场景中度量视频流分析系统综合性能的评价体系. 其次, MASSIVE 系统设计了帕累托改进调度器来计算帕累托最优的系统调度策略, 使得系统在 3 个维度上同时取得了相比已有系统更好的性能表现. 最后, MASSIVE 设计了虚拟流量整形器来保证各个设备在无线网络中按照调度策略上传视频流数据. 实验结果表明, MASSIVE 在多种典型的视频分析任务中, 相比于当前的代表性系统, 至少达到了 122.7% 的实时准确率、1.8 倍的系统容量和更好的系统一致性, 并达到了帕累托最优.

关键词 实时视频流分析, 边缘计算, 多设备协同, 多目标优化, 帕累托最优

1 引言

近年来, 随着深度学习在机器视觉领域的快速进步, 智能终端设备对环境的视觉感知能力不断增强, 促进了基于实时视频流分析的人工智能应用发展, 比如自动驾驶、无人飞行、视觉同步定位与建图、增强现实/混合现实等. 这些智能应用使用深度学习模型对摄像头产生的视频流内容进行自动分析和理解, 从而完成实时识别、分割和预测等复杂任务. 硬件平台的算力是限制深度学习应用在终端设备上的主要障碍, 因此实时视频流分析难以部署在资源受限的终端设备上. 边缘计算范式应运而生, 将终端设备产生的视频流数据上传到边缘服务器 [1], 可以借用其强大的算力实时运行复杂的深度学习

引用格式: 杨铮, 董亮, 蔡新军. 面向多设备协同场景的实时视频流分析系统. 中国科学: 信息科学, 2023, 53: 46–65, doi: 10.1360/SSI-2021-0179
Yang Z, Dong L, Cai X J. Toward cooperative multi-agent video streaming perception (in Chinese). Sci Sin Inform, 2023, 53: 46–65, doi: 10.1360/SSI-2021-0179

模型. 因为视频流数据上传会产生额外的视频传输时延, 增加终端设备获得视频分析结果的延迟, 所以终端设备对外界环境的相应速度随之降低. 如果对视频数据进行压缩, 则可以有效降低传输时延, 但是会损害视频的清晰度, 造成服务器上模型推断的准确率下降. 为了满足智能应用的实时性和准确性需求, 当前工作通过平衡视频传输延迟和模型推断准确率, 比如选择性压缩背景图像, 来提升实时视频流分析的性能 [2].

随着终端设备的互联互通, 人工智能应用逐步向多设备协同场景发展, 建立在终端设备之间的协同工作之上 [3]: 智能工厂中, 智能制造需要依赖于多个智能手臂之间的协同操作 [4]; 无人机蜂群中, 编队飞行依赖于多个无人机之间的协同飞行 [5]; 无人车队中, 车队自动驾驶也依赖无人车之间的协同驾驶 [6]. 在多设备协同场景的视频流分析系统中, 设备之间共享固定的网络带宽和边缘服务器算力, 因此需要设计高效的调度算法, 满足不同设备的性能诉求. 然而, 传统的边缘计算范式适用于提升某个设备的视频分析性能, 缺乏对整个协同系统的优化调度, 在多设备协同场景的表现难以令人满意. 这一方面是因为多设备之间的竞争导致网络带宽和服务器算力的利用效率下降, 引起各个设备的视频流分析性能普遍下降; 另一方面是因为不同设备的视频流分析性能存在差异, 少数 “迟钝” 的设备会造成 “木桶效应”, 拉低整个系统的性能表现. 因此, 本文将设备协同性引入系统设计中, 通过优化调度来提升实时视频流分析的综合性能表现.

本文面临的挑战主要有以下 3 个方面:

• 缺乏协同性角度的系统性能评价体系. 传统的视频流分析系统评价方式针对单个设备, 其具体度量标准主要由分析准确率和响应延迟构成, 无法体现整个多设备协同系统的综合性能.

• 调度策略难以全面提升系统性能. 多设备协同场景中, 设备对系统资源存在激烈的竞争关系, 当前系统缺乏高效的调度策略来全面提升系统的综合性能.

• 无线传输协议不支持优化调度. 多设备协同场景中, 设备与边缘服务器之间主要使用无线网络传输视频流数据, 缺乏对中心化调度策略的支持.

首先, 本文分析了多设备协同场景中实时视频流分析系统的性能要求, 提出了一种多维度评价体系来度量系统综合性能. 评价体系包含 3 个维度: 度量每个设备上视频流分析的实时准确率; 度量设备之间视频分析差异的系统一致性; 度量系统可承载设备数量上限的系统容量. 然后, 本文对多种代表性系统的综合性能进行测量, 通过对比实验结果来分析其性能局限性. 通过多设备协同场景的原理性实验, 本文从系统设计角度揭示性能局限性的来源, 提出针对性的改进思路. 最后, 本文设计了 MASSIVE 系统, 通过协同调度解决多设备的资源竞争和一致性问题, 并使得视频分析性能达到帕累托 (Pareto) 最优. MASSIVE 的核心设计包括两个部分: 帕累托改进调度器根据视频分析任务的特性和系统的资源池 (包括网络带宽、服务器算力等), 计算出满足帕累托最优的调度策略; 虚拟流量整形器保证系统中各个设备在无线网络中按照帕累托改进调度器计算的调度策略上传视频流数据.

总的来说, 本文的原创性贡献主要有以下方面:

• 在多设备协同的实时视频流分析系统中, MASSIVE 是第 1 个达到帕累托最优并解决一致性问题的边缘计算系统.

• 本文首次提出了适用于多设备协同场景的实时视频流分析系统的评价体系, 可以度量整个系统的综合性能.

• 本文设计了帕累托改进调度器来消除多设备的资源竞争, 使得 MASSIVE 在评价体系的 3 个维度上同时取得了相对已有系统更好的性能表现.

• 本文设计了虚拟流量整形器来保证多设备在无线网络条件下按照帕累托改进调度器计算的调度策略进行协同工作.

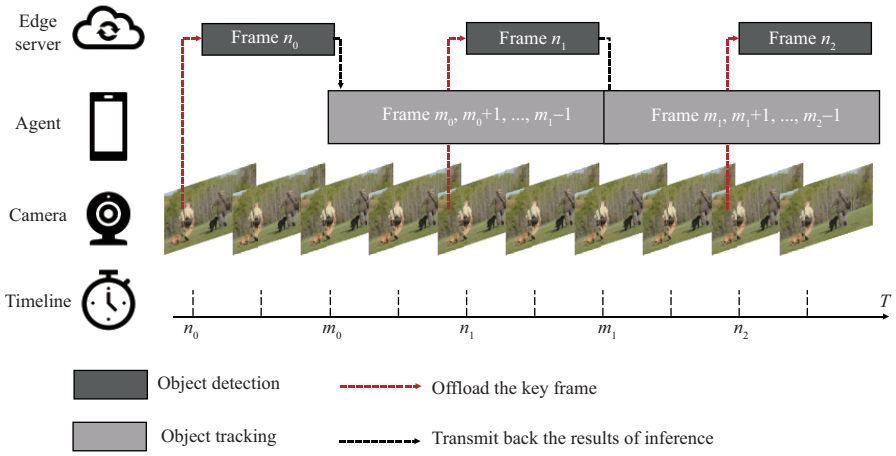

图 1　(网络版彩图) 实时视频流分析系统的边缘计算范式 (以目标检测任务为例)

Figure 1　(Color online) Workflow of a single-agent video streaming perception task (an example of offloading object detection)

• 本文在 30 个实际的典型视频分析任务中测评了 MASSIVE 和其他 4 种代表性系统, 结果证明相比于表现次好的系统, MASSIVE 达到了 122.7% 的实时准确率和 1.8 倍的系统容量以及更好的系统一致性, 并达到了帕累托最优.

本文的后续章节安排如下, 第 2 节对实时视频流分析系统的边缘计算范式进行介绍, 第 3 节对比了代表性系统的性能并分析其局限, 然后提出了 MASSIVE 系统的设计思路, 第 4 节介绍 MASSIVE 的工作流程和关键技术, 第 5 节介绍实验的方法、设置和流程, 第 6 节分析实验结果和评估系统性能, 第 7 节对相关工作进行综述, 最后第 8 节对本文研究内容进行总结.

2　实时视频流分析系统的边缘计算范式

本节介绍实时视频流分析系统的边缘计算范式, 并结合应用场景实例讨论其在多设备协同场景下的综合性能要求.

2.1　基本工作流程

实时视频流分析系统的边缘计算范式, 借助边缘服务器算力运行复杂的视觉分析算法 (包括视觉识别、跟踪和预测等), 帮助算力受限的终端设备获得对外界环境的实时感知能力. 如图 1, 以典型的视觉目标检测算法为例, 整个工作流程包含两条流水线: 一条流水线以某种频率将终端设备上摄像头拍摄的视频关键帧上传给边缘服务器, 在服务器上运行目标检测模型后, 终端设备会收到关键帧的分析结果; 另一条则使用终端设备的算力对比当前视频帧和最近关键帧的内容差异, 普遍利用视觉目标跟踪算法将关键帧上的模型分析结果更新到当前的视频帧上. 最近的研究中, 这种高效的边缘计算范式广泛适用于各种基于像素的视觉分析场景, 比如移动 AR 中的环境感知 [7, 8]、摄像头实时监控 [9, 10] 和机器人视觉定位 [11, 12].

2.2　多设备协同场景的性能指标

在多设备协同场景中, 多个智能设备共享系统资源 (网络带宽和服务器算力), 通过协同工作来完

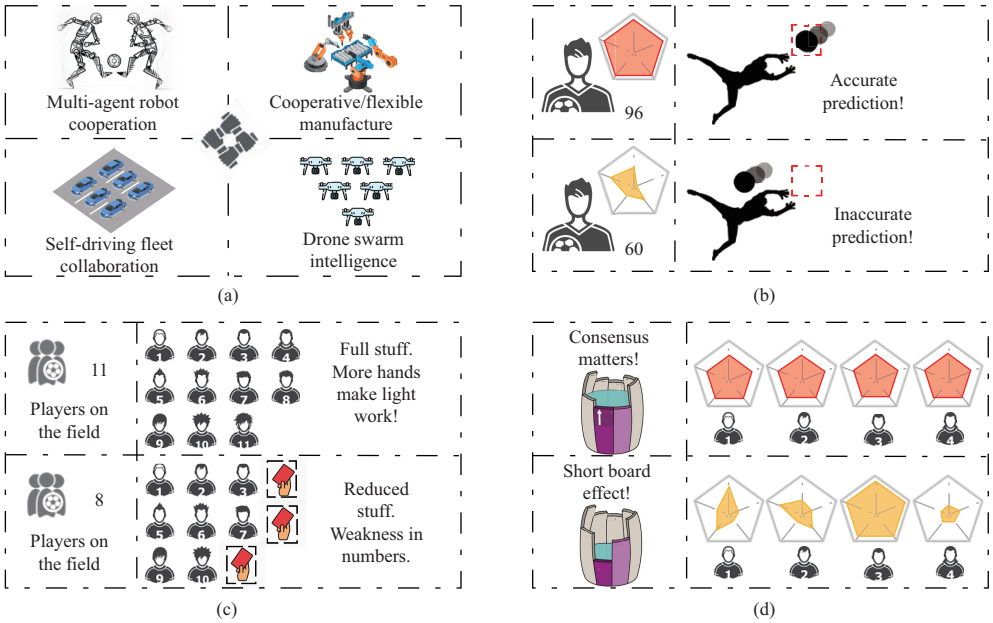

图 2 (网络版彩图) 多设备协同系统及其性能指标

Figure 2 (Color online) Background of cooperative multi-agent systems. (a) Multi-agent scenarios; (b) individual intelligence; (c) group capacity; (d) group consistency

成智能任务. 如图 2(a), 智能制造流水线上的多个机械手臂完成柔性制造任务 [13~15], 类似的还有多机器人系统 [16]、自动驾驶车队 [17] 和无人机蜂群 [18]. 随着设备数量的增加, 系统资源无法满足每个设备最优的资源请求, 设备之间的竞争加剧使得系统资源的利用效率下降, 最终影响协同系统的综合性能表现. 在多设备协同的场景中, 系统的综合性能指标既需要反映个体的准确性, 又需要体现系统的整体协同性. 我们以经典的球队系统为例来定义系统的 3 个综合性能指标: 第一是视频分析的实时准确率, 度量各个设备视频流分析系统的性能表现, 类比球队中各个队员的基础能力值 (如图 2(b)). 第二是系统容量 (如图 2(c)), 度量协同系统中的设备数量上限, 类比比赛中上场的队员数量, 数量多的一方占有明显优势, 但会受到系统资源和调度能力的限制. 第三是系统一致性 (如图 2(d)), 度量不同设备的性能差异, 类比球员的水平差异, 设备性能差异过大引发木桶效应会严重限制系统的协同能力. 多设备协同场景追求更高的实时准确率、更大的系统容量和更好的系统一致性, 但是, 这 3 项指标之间存在相互冲突, 限制了多设备协同的群体智能的发展.

3 系统设计方法

本节定义了系统综合性能的度量公式, 并测量和分析代表性系统的性能表现, 然后设计原理性实验分析代表性系统的设计局限性, 并提出协同性系统的设计思路.

3.1 性能评价体系

当前, 多设备协同场景中的实时视频流分析系统缺乏统一的评价体系, 现有系统的专用评价体系不具备对系统综合性能的评价能力. 本文根据多设备协同场景的特点正式定义系统性能评价体系和公式.

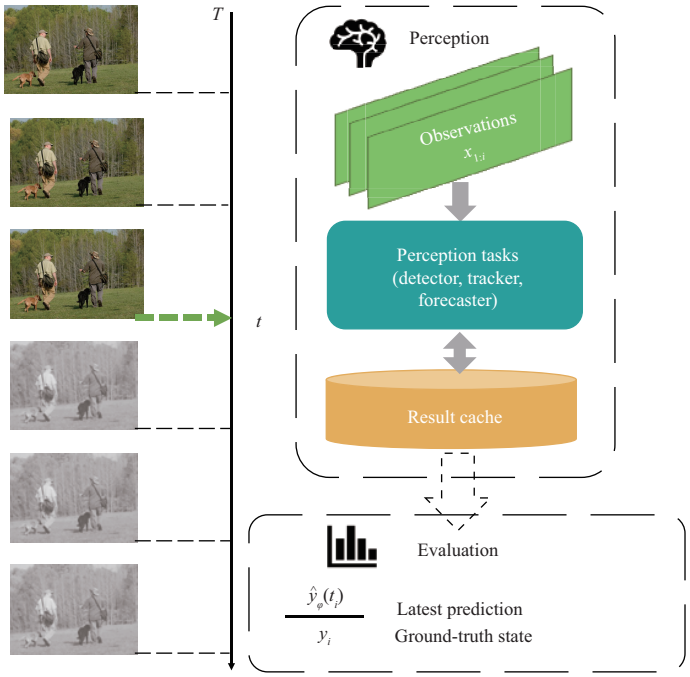

图 3　(网络版彩图) 实时视频流分析任务的评价方法

Figure 3　(Color online) Metrics to evaluate a video streaming perception task

• **实时准确率.** 和离线视频分析不同, 实时视频流分析关注如何持续地反馈外部世界的状态. 传统的评价指标包括分析准确率和延迟两项关键指标, 分别使用一种计算方式进行度量. 我们需要一种融合的指标, 直接反映终端设备在任何时刻感知外部世界的准确性. 本文引入实时准确率, 既继承原分析准确率的计算方法, 又能实时反映终端设备对外界环境的最新感知结果, 并适用于所有基于单帧的视频分析任务. 图 3 说明了使用实时准确率来度量实时视频流分析性能的过程: $\{(x_i, y_i, t_i)\}_{i=1}^{\mathrm{T}}$ 代表一组相机观测、真实世界的状态和对应时间戳. 假设 p 是一个待评价的任务, t 时刻 p 的输入是至今为止的观测 $\{(x_i, t_i)|t_i < t\}$ 以及 t 时刻之前的所有预测的缓存. p 将会在 τ 时刻产生一个预测 $\hat{y}$, 产生的 N 个输出存储为元组 $(\hat{y}_n, \tau_n)_{n=1}^{N}$. 由于 p 的处理延迟的存在, 通常情况下 p 的运行会滞后于视频帧的产生速度, 也就是说 $N \leqslant T$. 为了分析 p, 我们将 t_i 的最新预测和当时的真实世界状态 y_i 进行对比. 首先计算 t_i 时刻的最新预测的下标: $\psi(t) = \arg\max_n \tau_n < t$; 然后, 结合原分析准确率的误差函数 l, 我们正式定义实时准确率的误差计算 L: $L = l((y_i, \hat{y}_{\psi(t_i)})_i^{\mathrm{T}})$. 这个基础的误差函数 L 适用于之前提到的所有实时视频流分析任务.

• **系统容量.** 系统容量反映多设备协同系统的规模, 指在一定资源条件下系统可以支持的处于良好工作状态的终端设备的数量上限. 在定义该指标前需要先定义合理的设备性能阈值来判断设备是否属于良好工作状态, 设备性能阈值的设定则需要根据具体多设备协同任务的性能要求来确定. 为公平起见, 本文将设备在同一系统环境下的最优实时准确率作为标准值, 即环境中只有该设备在运行, 而没有其他设备竞争的情况下, 该运行设备能达到的最优性能表现. 在本文的实验中, 根据视觉分析领域的惯例设定 70% 为设备性能阈值, 认为性能表现高于标准值 70% 的终端设备属于处于良好工作状态. 于是系统容量的定义是在每个终端设备的实时准确率达到标准值 70% 以上的情况下, 系统所能支持的最大终端设备数量.

• **系统一致性.** 系统一致性反映不同终端设备性能表现之间的差异, 定义为不同终端设备的实时准确率的标准差. 系统一致性和终端设备数量有直接的关系, 即随着数量的上升, 设备性能差异更加明显. 在系统资源富余的情况下, 设备之间的性能一致性容易通过调度系统保证 [19]. 如果未经单独说明, 本文中系统一致性的前提是终端设备数量等于系统容量.

3.2 设计空间和性能表现

系统的设计空间包含不同维度的系统设计理念和主要技术, 已有的代表性工作的设计空间主要由以下 5 个基础维度组成.

• 模型压缩与加速. 深度神经网络模型的算力和内存需求较大, 难以在资源有限的嵌入式系统上满足时延敏感类智能应用的需求 [20]. 一个自然的想法是在不显著影响精度的前提下对神经网络进行压缩和加速. 在过去的 5 年, 大量的工作在这一方面取得了显著进展 [21], 具体技术可以分为 4 类: 剪枝与量化 (parameter pruning and quantization) [22,23]、低秩矩阵分解 (low-rank factorization) [24,25]、紧凑卷积滤波器 (compact convolutional filters) [26,27] 和知识蒸馏 (knowledge distillation) [28,29]. 然而这些技术都有本质性的缺陷, 比如模型偏差、结构性限制或者需要额外训练等, 使得表现始终难以比肩原模型 [30].

• 视频配置自适应. 实时视频流分析系统, 会以不同的采样频率在终端设备上抽取视频帧, 压缩成不同的分辨率后上传到边缘服务器, 然后使用对应的模型进行处理. 其中一个分辨率和采样帧率的组合被叫做视频配置 [31], 而不同的视频配置会导致不同的分析精度以及资源消耗. 相关工作 [32,33] 通过将视频配置自适应于不同时间的网络状况和资源情况来提高系统的性能表现.

• 帧间动作预测. 为了缓解资源压力, 系统可以根据视频流的时间相关性预测待检测目标的运动趋势, 从而降低视频帧上传频率. 相关工作 [7,12,34] 在预测到视频内容没有发生突变的情况下, 使用低代价的跟踪模型替代高代价的检测模型来更新对当前世界的预测. 显然, 这类技术无法适用于背景持续变化或者存在高速运动物体的情况, 只适用于固定摄像头或者物体低速运动等场景.

• 帧内内容预测. 为了降低带宽消耗和传输延迟, 系统可以选择性的压缩视频帧, 主要上传神经网络模型敏感的像素或区域. 相关工作 [34,35] 使用结构简单的分类器过滤视频内容, 选择兴趣区域 (region of interest, ROI) 进行上传. 然而, 结构简单的分类器分辨能力模糊, 难以提供对 RoI 的准确预测, 文献 [2] 提出 RoI 预测功能应该由服务器端的深度神经网络来完成. 文献 [11,12] 则提出通过预估不同视频帧对分析任务准确性的影响, 选择性上传更重要的视频帧, 在降低上传视频帧频率的同时尽量避免影响分析任务的性能

• 模型的协同计算. 为了提高深度学习模型的推断效率, 系统可以把计算任务拆分给终端设备和边缘服务器分别完成. 视频分析任务的构成是有向无环图 (direct acyclic graph, DAG) [36,37], 相关工作 [38,39] 提出将模型推断过程拆分成本地和云端分别运行的两部分, 并降低两者之间的数据传输延迟. 然而, 这类方法会产生额外计算和带宽消耗, 相比于深度压缩原视频后直接上传的优化方法缺乏竞争力 [40].

近期的代表性系统通过采用以上 5 种维度的优化方法或者组合来提升系统的整体性能表现, 它们都是以提高单个终端设备的实时准确率为目的, 最终的性能提升效果各有优劣. 正如图 4 右图所示, 我们总结了各种系统在 30 个实际场景下的典型视频流分析任务中的性能表现, 并把它们放在统一的评价体系中进行比较. 实验设置和分析的具体细节可以参照第 6 节. 评价体系中的 3 方面指标存在着明显的矛盾关系, 代表性系统只能在某一或两方面取得相对优势而无法获得全面的性能提升. 这种比较证明当前设计空间中的优化维度存在局限性, 只提高单个终端设备的性能, 而忽视多设备之间的协

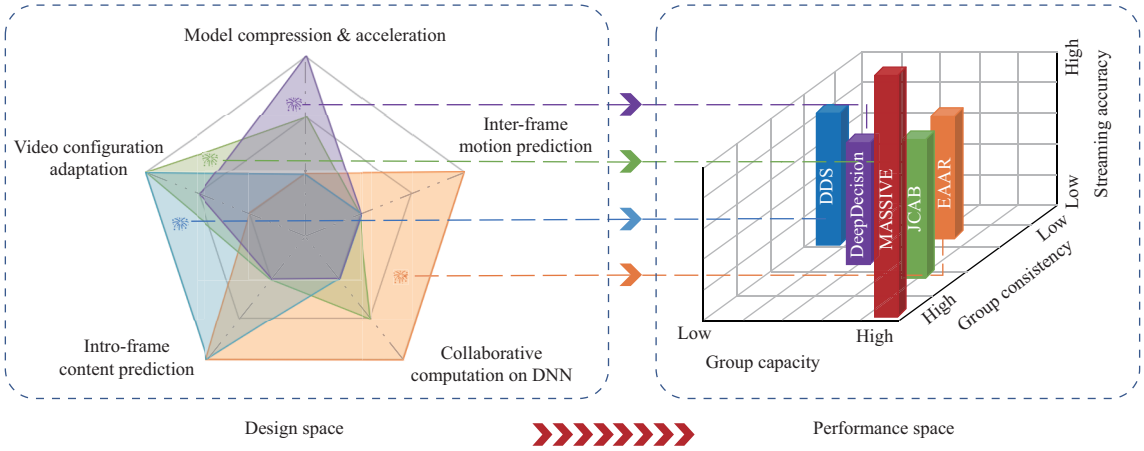

图 4 (网络版彩图) 代表性系统的设计空间和它们的性能对比

Figure 4 (Color online) Performance comparison of state-of-the-art systems and their original intentions in design space

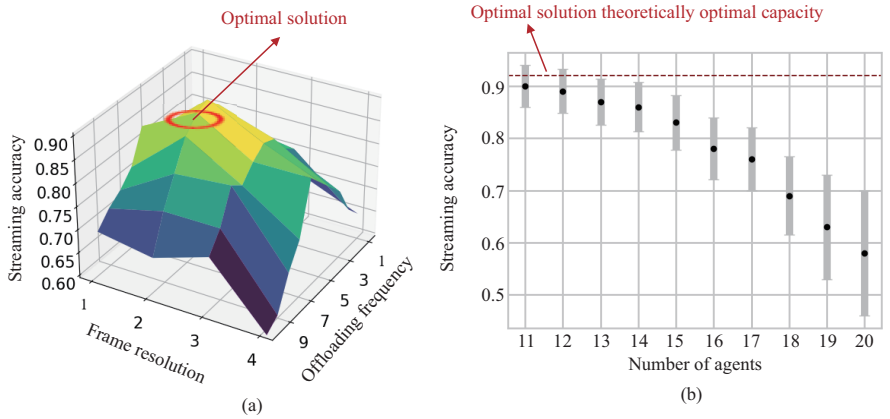

图 5 (网络版彩图) 单设备/多设备系统的性能表现的研究

Figure 5 (Color online) Study on single-/multi-agent systems. (a) Single-agent performance profile; (b) multi-agent performance

同. 本文希望增加系统协同性方面的设计, 来克服已有系统在不同指标之间的矛盾, 设计一个在 3 方面达到更优的, 适用于多设备协同场景的新系统.

3.3 系统设计的局限性

当前系统难以获得综合性能的全面提升, 我们在多设备协同场景下设计原理性实验研究实时视频流分析的性能特性, 试图解释已有系统局限性的来源. 以前文提到的目标检测任务为例, 工作流程和第 2.1 小节中一致. 我们使用 YOLOv3 [41] 作为目标检测模型, KCF [42] 作为目标跟踪模型, 单次目标检测所要上传的视频帧的尺寸 (分辨率) 和两次目标检测之间的间隔 (上传频率) 作为影响实时准确率的两个主要参数. 我们测试了系统在设置不同分辨率 (360, 480, 720, 1080 P) 和不同上传频率 (1~10 fps) 时在自动驾驶任务中的表现 (任务具体说明见第 5 节). 图 5(a) 记录了单个终端设备的场景中, 不同参数设置下该设备的实时准确率变化. 而从图 5(b) 可以发现, 随着应用场景中终端设备数量的增加, 实时准确率的平均值开始逐渐加速下降, 同时终端设备之间的性能差异逐步拉大. 这种情况在博弈论

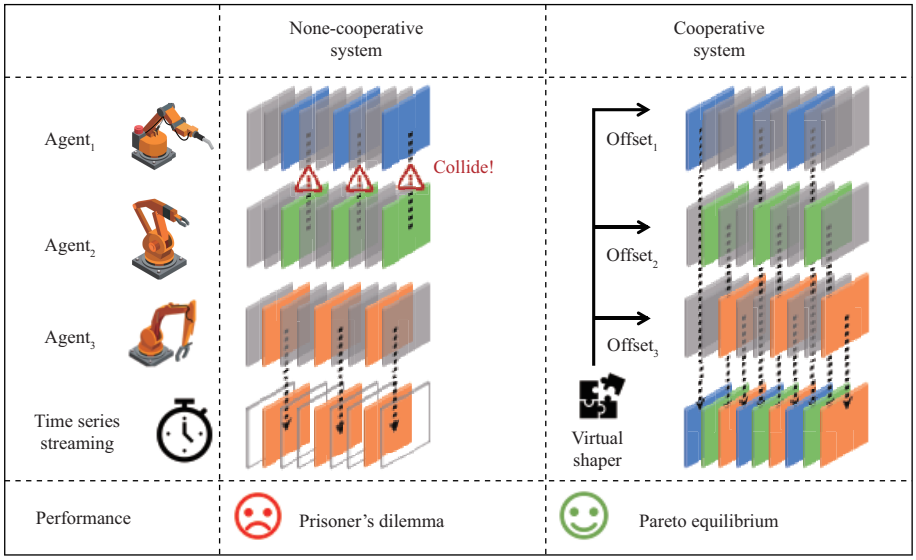

图 6 (网络版彩图) 设想: 虚拟整形器的示意图

Figure 6 (Color online) Insight: the diagram of virtual shaper

中被称作纳什 (Nash) 均衡 [43], 往往会导致 "囚徒困境", 在非协同的多设备系统中较为常见. 而我们期望实现多设备协同, 以达到帕累托最优, 对系统资源进行高效分配和利用. 因此, 本文将协同性设计理念加入设计空间中来设计一种面向多设备协同性场景的实时视频流分析系统.

3.4 协同性系统的设计思路

原理性实验结果表明, 随着终端设备数量增加, 多个终端设备同时需要上传视频帧并进行分析的概率上升, 发生冲突并抢占资源的概率也不断上升. 一个自然的改进思路是错峰地协调各个终端设备上传视频帧的时机, 避免发生各个终端设备争夺系统资源的情况. 图 6 通过可视化不同终端设备上传视频帧的时机, 解释如何将一个非协同性系统改进为协同性系统. 在非协同性系统中, 因为缺乏协同调度, agent_1 和 agent_2 恰好同时上传视频帧并引发了冲突, 结果同时增加了双方的传输时延. 因此本文提出了右图中虚拟整形器的系统调度概念, 它赋予不同终端设备不同的间隔补偿以错开它们的上传时机, 将不同终端设备向边缘服务器发送的多条视频流重组为一条连续而高效的视频流. 这样不仅能避免不同终端设备之间的上传冲突, 还可以提高对网络带宽的有效利用.

4 多设备协同的实时视频流分析系统

本节展示了 MASSIVE 系统的工作流程, 并详细介绍两个关键技术, 包括帕累托改进调度器和虚拟流量整形器.

4.1 系统概述

MASSIVE 是为多设备协同场景设计的实时视频流分析系统, 适用于所有基于单帧的视频分析任务, 能够为多设备协同任务提供高效的视频流分析服务. MASSIVE 的系统设计以视频流分析系统的边缘计算范式为基础, 通过优化调度系统来增强系统在协同性方面的性能. 增强系统协同性的设计主要包含两个关键模块: 一是帕累托改进调度器, 用于计算帕累托最优的系统调度策略; 二是虚拟流量

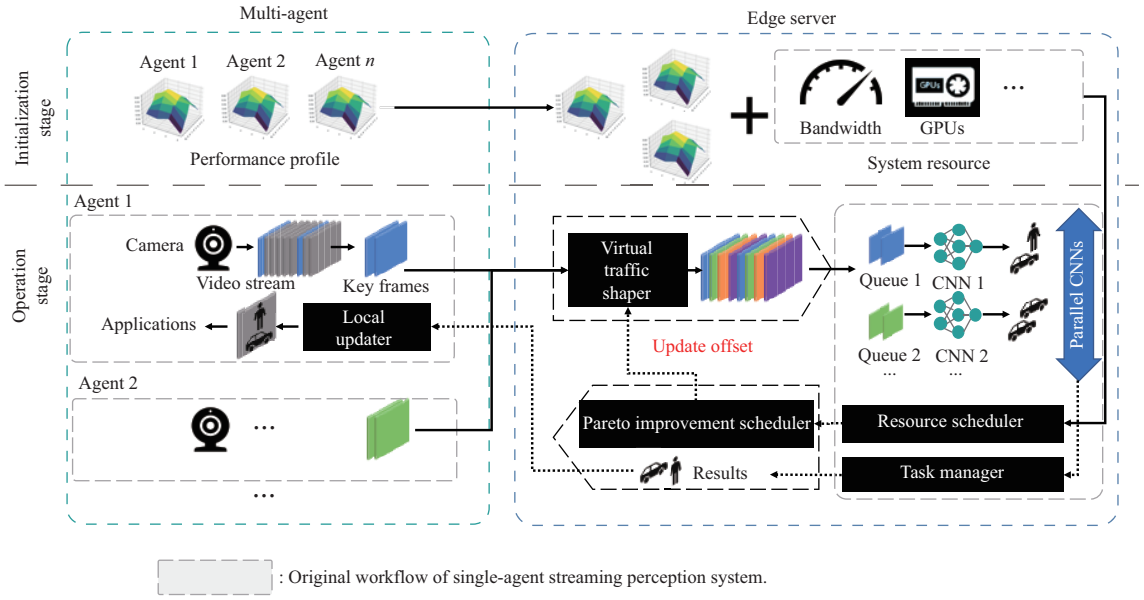

图 7 (网络版彩图) MASSIVE 系统工作流程图

Figure 7 (Color online) System workflow of MASSIVE

整形器, 支持系统在无线网络中执行帕累托最优调度.

MASSIVE 适用于所有基于单帧的视频分析任务, 图 7 以目标检测任务为例, 说明具体工作流程. 在系统初始化阶段, 每个终端设备根据分析任务提供一个概述文件 (profile), 记录不同上传参数 (图像分辨率和上传频率) 和性能表现之间的关系, 并同时保存在终端设备本地和边缘服务器上. 在系统运行期间, 帕累托改进调度器会综合各个 profile, 以及系统资源池来计算系统调度方案, 决定各个终端设备的上传时机和上传参数; 终端设备则基本继承第 2.1 小节中的边缘计算范式的工作流程, 同时配合边缘服务器分配的上传时机和上传参数, 通过虚拟流量整形器上传视频流.

4.2 帕累托改进调度器

帕累托改进调度器以多设备的性能表现为优化目标, 根据概述文件以及系统资源池制定帕累托最优的优化策略.

多设备性能优化问题设定每个终端设备的实时准确率 $\{a_i\}_{i=1}^{N}$ 为一组优化目标, 通过调整优化参数组 $\{o_i, f_i, r_i, d_i\}_{i=1}^{N}$ 来优化目标组, 其中 f_i 和 r_i 来自于终端设备 agent_i 的 profile, 而 d_i 则表示设备上传一帧的时长. 帕累托改进调度器赋予不同智能体 agent_i, 不同的上传间隔补偿 o_i, 隔开不同终端设备的上传时机, 需要满足如下约束条件:

$$\begin{aligned} &\forall i, j \in N^*,\ k, l \in N,\ i < j \leqslant N, \\ &\text{s.t.}\ \left(\frac{k}{f_i} + o_i + d_i\right) \cap \left(\frac{l}{f_j} + o_j + d_j\right) = \emptyset. \end{aligned} \tag{1}$$

这种目标优化问题是一种典型的 NP-hard 问题 [44], 其中 r_i, f_i 都是有限的离散值, 还需讨论 o_i 的取值. 相关工作 [45, 46] 对于此类帕累托优化问题的近似解法讨论充分, 近似解集合表示为

$$\{\{\{o_i, f_i, r_i\}_{i=1}^{N}\}_j\}_{j=1}^{M}. \tag{2}$$

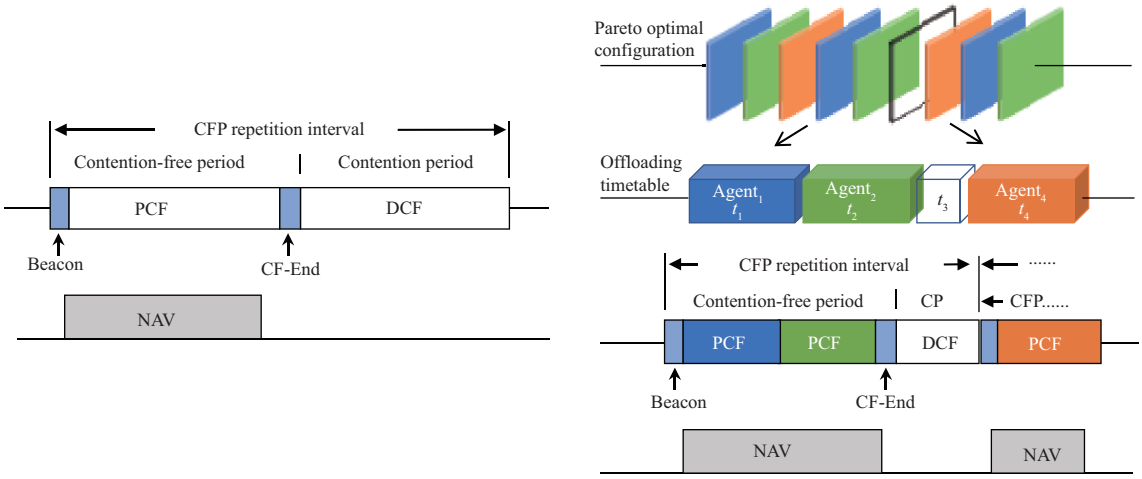

图 8 (网络版彩图) 虚拟流量整形器的工作流程

Figure 8 (Color online) Workflow of virtual traffic shaper

接下来, 为了提高系统一致性, 要在近似解集合中挑选满足以下条件的解:

$$\{o_i, f_i, r_i\}_{i=1}^{N} = \arg\min \sigma(\{a_i\}_{i=1}^{N}). \tag{3}$$

4.3 虚拟流量整形器

在多设备协同场景中, 终端设备与边缘服务器之间通过无线网络通信, 仍基于载波侦听多路访问/冲突避免机制, 因此无法保证特定设备在特定时间片上传视频帧. 虚拟流量整形器在无线网络中对多个时延敏感的视频数据流进行重塑, 确保不同设备根据帕累托改进调度器计算结果上传视频流数据.

虚拟流量整形器的设计利用了网络分类矢量 (network allocation vector, NAV), 图 8 展示了 NAV 在点协调功能 (point coordination function, PCF) 中的应用. PCF 是 IEEE 802.11 WLANs 中的基本模式之一, 以中心基站提供中心化的信道访问控制, 为时延敏感数据流提供严格的时延保证. 以某种频率不断重复的非竞争阶段 (contention-free period, CFP) 中, PCF 会实行轮询 (poll) 机制, 被轮询到的通信站会获得免竞争的传输信道, 而其他通信站则会将各自的 NAV 值设置成最大长度以保持静默.

图 8 右图描述了虚拟流量整形器的工作流程. 根据帕累托改进调度器得到的调度配置, 可以得到一张描述不同终端设备上传时机的时间表. 根据时间表, 边缘服务器会在对应时间触发 CFP 时段, 发送轮询指令 (CF-poll) 给对应的终端设备来获取最新的视频帧, 然后更新其他终端设备上的 NAV 值为上传该帧的时长. 当该帧发送完毕后, 时间表上如果紧跟着有上传其他视频帧的计划, 边缘服务器会维持 CFP 状态, 向对应的终端设备发送新的 CF-poll 并再次以同样方式更新其他终端设备的 NAV 值.

5 实验与设置

5.1 实验环境

我们在商用设备上测试了 MASSIVE 系统: 终端设备选用了基于 mt7628 芯片的开发板, 运行开

源的 Linux OpenWrt 系统, 边缘服务器选用了 Lenovo IdeaPad-Y700, i7-6700HQ CPU, 2.6 GHz main frequency, GTX 2080 GPU, 以及一个网络仿真器来获取精确的传输延迟和带宽消耗. 实验期间, 我们让多个终端设备实时向边缘服务器发送编码后的视频帧, 然后边缘服务器解码图片后运行指定的 DNN (deep neural network) 模型, 并返回分析结果. 在 OPNET Modeler 14.5 版本的仿真工具的帮助下, 我们设置网络带宽为 54 Mbps, 测量系统运行时的各项精确网络参数. 在大多数场景中, 视频流的比特率是 2400 kbps. 本文实现了 4 种代表性系统, 并为它们提供了所需要的硬件设备、网络环境和视频输入以完成公平的比较. 除非另行声明, 本文使用 YOLOv3[1)]作为边缘服务器端的目标检测模型, 使用 MobileNetV2dilated+C1_deepsup [47] 作为语义分割模型.

5.2 数据集

为了分析 MASSIVE 在不同视频类型下的表现, 本文收集了 4 种视频数据集, 每种代表了一类典型的多设备协同的任务场景. 其中交通数据集 (Traffic)、传送带数据集 (Conveyor) 和足球比赛数据集 (Soccer) 是我们从视频网站上搜索关键词下载, 并人工去除了相关性差的部分后制成的. 无人机数据集 (Drone) 取自于 VisDrone2019-MOT 公开数据集 [48, 49]. 我们刻意挑选了满足各种特性的视频, 比如物体运动速度不同、目标数量不同、目标相对大小不同等, 以增强实验结果的全面性和说服力. 这些实时视频流分析任务包括 (1) 在 Traffic 和 Drone 数据集中检测 (或分割) 车辆; (2) 在 Soccer 数据集中检测 (或分割) 足球; (3) 在 Conveyor 数据集中检测 (或分割) 箱体. 尽管部分视频中的物体已有人工标注, 我们将对原视频进行离线推断的结果作为真实值来计算实时准确率, 以消除不同神经网络模型本身的性能对实验结果的影响.

5.3 对比系统与性能评价指标

本文选取了 4 种近期的代表性系统作为对比系统, 代表两类最新的技术: 一种是专注于提升单个设备上的实时准确率的系统 (EAAR [34], DeepDecision [35], DDS [2]), 另一种则考虑了多设备的场景 (JCAB [32]). 实验中为所有对比系统提供了需要的模型和充足算力, 以保证对比实验的公平性. 系统综合性能评价指标采用在第 3.1 小节定义的实时准确率、系统容量和系统一致性. 目标检测任务使用 AP50 来计算准确率, 语义分割任务使用与目标类别相关像素的 IoU (intersection over union) 来计算准确率.

6 实验分析

6.1 端到端性能提升

整体性能. 图 9 从实时准确率、系统容量和系统一致性 3 方面对比了 MASSIVE 和其他系统的表现. 在两种视觉任务中, MASSIVE 的性能都全方位地显著超越了其他的系统, 比表现次好的 JCAB 在准确率方面提升了 22.7%, 在系统容量上拓展到了 1.8 倍并获得了更好的系统一致性. JCAB 的性能则整体优于其他系统 (DDS, EAAR, DeepDecision). 因为 JCAB 设计了一个全局的调度器来配置资源, 但是缺乏对多设备竞争的深度分析, 没有利用上传特性对系统调度进行优化. 其他 3 种系统的显著缺陷是其设计空间中缺乏对多设备协同场景中终端设备之间协同性的的考量. 我们从以下两个方面进一步解释.

1) ultralytics/yolov3: v9.0 — YOLOv5 Forward Compatibility Release. 2020. https://doi.org/10.5281/zenodo.4308573.

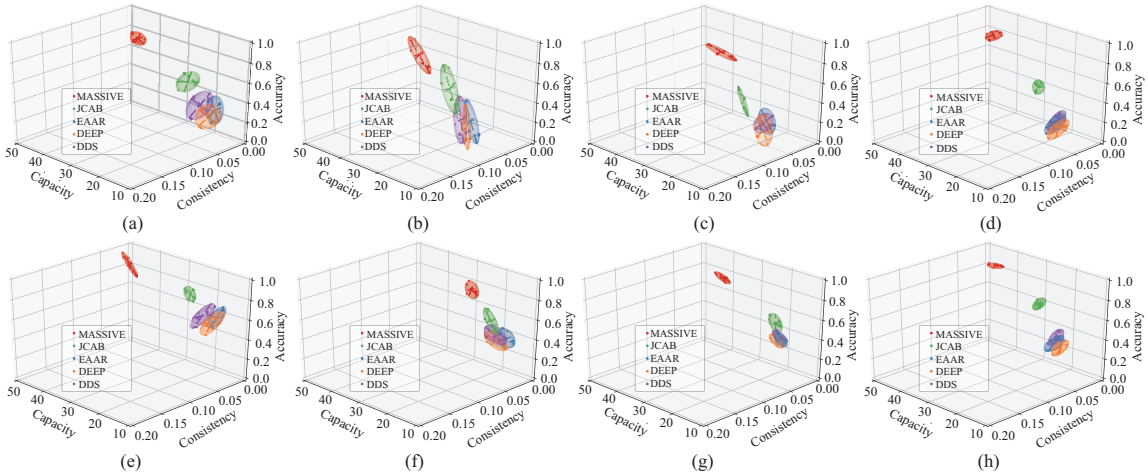

图 9 (网络版彩图) 各数据集上 MASSIVE 和对比系统的性能对比

Figure 9 (Color online) Comparison of streaming accuracy, capacity, and consistency between MASSIVE and all baseline methods on various datasets. (a) Object detection (Traffic); (b) object detection (Soccer); (c) object detection (Drone); (d) object detection (Conveyor); (e) semantic segmentation (Traffic); (f) semantic segmentation (Soccer); (g) semantic segmentation (Drone); (h) semantic segmentation (Conveyor)

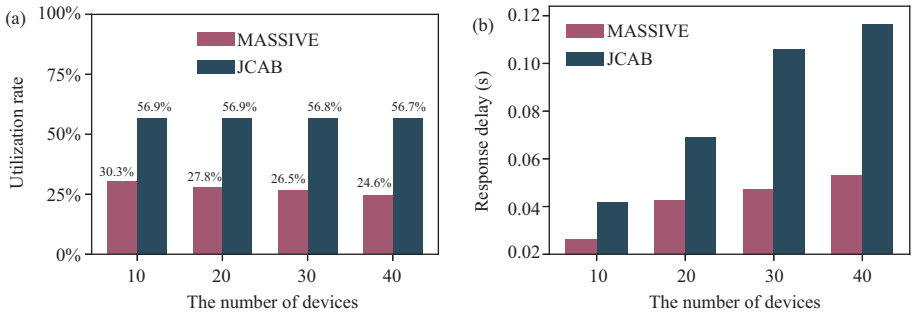

图 10 (网络版彩图) MASSIVE 和对比系统的带宽利用的对比

Figure 10 (Color online) Bandwidth utilization of MASSIVE and the best baseline method. (a) Bandwidth utilization; (b) response delay

带宽利用率. 图 10(a) 展示了 MASSIVE 和 JCAB 在不同设备数量的场景中运行时的带宽利用率. 在任何设备数量下, MASSIVE 的带宽利用率几乎是 JCAB 的两倍, 而 JCAB 的带宽利用率还会随着设备数量增加而下降. 这意味着如果终端设备以相同的参数上传视频帧, MASSIVE 将比 JCAB 容纳更多的终端设备而不对其实时准确率造成影响. 这就解释了 MASSIVE 的系统容量显著高于 JCAB 的原因. 为了研究每一个终端设备的表现, 我们计算了每个设备的实时准确率, 如图 11 所示, 其中每一维度变量表示一个终端设备的实时准确率. 如果 JCAB 想要容纳和 MASSIVE 一样数量的设备 (45 个), 根据其调度逻辑需要下调设备的上传参数, 导致实时准确率的显著下降. 同时, 降低不同设备上传视频帧的上传参数也会扩大性能差异, 导致系统一致性随之下降.

分析响应延迟. 图 10(b) 表明, 在各种数量终端设备的实验场景中, MASSIVE 的分析响应延迟始终低于对比系统, 是实时准确率超越其他系统的重要原因. 根据实验数据, MASSIVE 系统中的响应延迟只有 JCAB 的 1/2~1/3, 两者之间的性能表现随着设备数量的增加进一步扩大, MASSIVE 相比于对比系统竞争优势更加明显. 帕累托改进调度器的设计使得所有的设备能够错峰并轮流上传它们的视

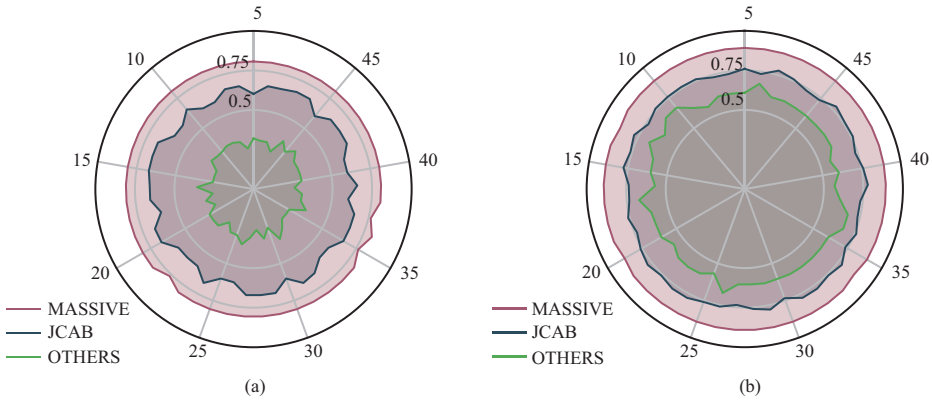

图 11 (网络版彩图) 在设备终端数量设置为 45 时 (即 MASSIVE 的系统容量值), MASSIVE 和 JCAB 在两种任务中的实时准确率的对比

Figure 11 (Color online) Streaming accuracy of each agent in two perception tasks when the number of agent is 45 which is the group capacity of MASSIVE. (a) Object detection; (b) semantic segmentation

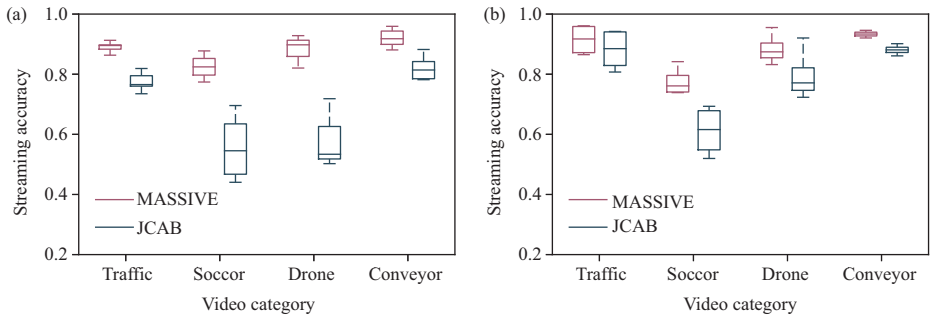

图 12 (网络版彩图) MASSIVE 和 JCAB 在 4 种不同种类视频上运行两种视频分析任务的实时准确率对比

Figure 12 (Color online) Streaming accuracy of two tasks in four different video categories. (a) Object detection; (b) semantic segmentation

频帧, 使得单个终端设备能在短时间内利用最大带宽而无需和其他设备竞争, 每个终端设备就能够尽可能快地得到返回的推断结果. 尽管对比系统也优化多设备系统的资源配置, 但是没有有效利用多设备协同的特性, 无法进一步压缩响应延迟. 实验结果表明, MASSIVE 相比于 JCAB 有显著的带宽利用率和分析响应延迟的提升, 是整体性能超越对比系统的重要原因, 证明了帕累托改进调度器和虚拟流量整形器这两项关键设计的有效作用.

6.2 不同应用场景的性能表现

视频类型的影响. 图 12 对比了 MASSIVE 和 JCAB 在 4 类数据集上运行两种视频分任务的性能表现. 在所有测试中, 智能体的数量设置成 JCAB 的系统容量而不是更高的 MASSIVE 的系统容量, 这样性能结果的对比会更利于 JCAB. 即便如此, MASSIVE 在两种任务中都取得了显著高于 JCAB 的实时准确率. 在目标检测任务中, MASSIVE 相比于 JCAB 在 Conveyor 数据集上的表现提升了 10% (最少), 而在 Drone 数据集上提升了 29.8% (最多). 在语义分割场景中, MASSIVE 在 Traffic 数据集上的表现提升了 3.4% (最少), 而在 Soccer 数据集上提升了 19.0% (最多). 基于这些表现, 我们不妨做出两点推测: 第一, MASSIVE 在目标检测任务获得的性能提升相比于在语义识别上更明显, 内在原因

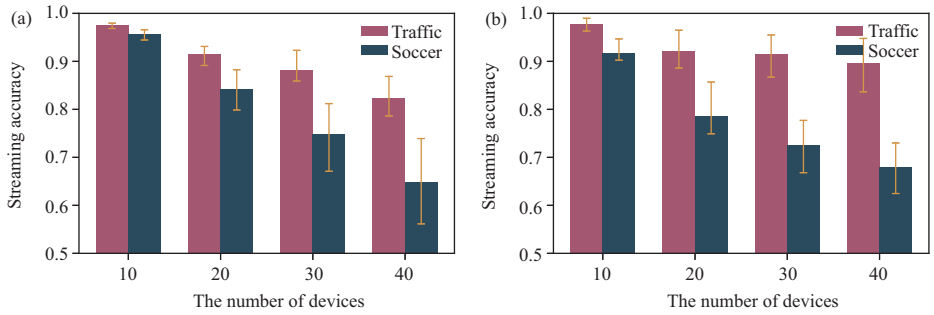

图 13 (网络版彩图) 在 Traffic 和 Soccer 数据集上运行 (a) 目标检测任务和 (b) 语义分割任务的准确率对比

Figure 13 (Color online) Streaming accuracy of two tasks in the Traffic and Soccer datasets. (a) Object detection; (b) semantic segmentation

可能是准确率的计算方式导致的. 具体来说, 目标检测的计算方式是预测的物体边框 (bounding box) 和真实的物体边框之间的 IoU, 而语义分割的计算方式是关于目标物体的预测像素和真实像素之间的 IoU. 真实的物体边框是包含物体的长方形, 而真实的物体像素则严格的区分了物体的边缘. 因此当物体运动时, 真实的物体边框移动明显, 而真实的物体像素则变化相对较小. 第二, 不同数据集的性能表现也和任务类型相关. 比如在目标检测中, Drone 数据集比其他的表现明显都好. 但是在语义分割中, Drone 数据集则比 Soccer 数据集的提升要小.

目标物体类别的影响. 我们以在视频中频繁出现的物体类别为对象进行研究, 发现目标物体的特性会对系统表现产生不同影响. 图 13 显示了在 Traffic 和 Soccer 数据集中, 实时准确率随着设备数量增长的变化的差异. 可以看到在两种任务中, 随着设备数量的增加, Soccer 数据集中实时准确率的降低要明显快于 Traffic 数据集. 通过观察发现, Soccer 数据集中的检测目标是足球, 而 Traffic 数据集中的检测目标是车辆. 因为足球在图像中的体积相对更小, 检测难度更大, 所以随着终端数量的增长, 设备上传频率降低, 检测足球的准确率下降速度比检测车辆更快. 同时, 从图 9 可以看出, MASSIVE 在 Soccer 数据集上的系统容量要低于在 Traffic 中的表现. 这是因为系统容量的定义是以实时准确率为前提的, 为了保证各个设备的实时准确率满足要求, 在 Soccer 数据集中设备必须以比在 Traffic 数据集中更高的频率上传视频帧, 系统容量就相应降低.

目标运动速度的影响. 图 14(a) 和 (b) 显示了在 Traffic 和 Drone 数据集中, 实时准确率随着设备数量增长的变化. 虽然这两个数据集中的目标物体种类都是车辆, 但是 Drone 数据集上准确率随着设备数量增加的下降速度明显高于 Traffic 数据集. 进一步对比两个数据集的内容, 我们发现原因应该是 Drone 中车辆的运动速度远高于 Traffic 数据集. 在实际应用场景中 (比如自动驾驶), 终端设备需要特别关注移动速度更高的物体, 因为它们需要采取更及时的措施来预防可能存在的风险. 目标物体移动速度更快的数据集对设备数量的增加更加敏感, 而 MASSIVE 对这些场景有着更明显的提升. 同时, 从图 9 可以看出, MASSIVE 在 Drone 数据集上的系统容量也要低于在 Traffic 中的表现. 这同样是因为系统容量的定义是以实时准确率为前提的, 为了保证各个设备的实时准确率满足要求, 在 Drone 数据集中设备必须以比在 Traffic 数据集中更高的频率上传视频帧, 系统容量就相应降低.

6.3 不同系统参数的性能表现

帕累托改进调度器中参数的影响. 在实验中我们固定系统容量为 20, 通过调整 f_i 和 r_i 的值, 观察对终端设备实时准确率的影响, 最终结果如图 15 所示. 首先, 当固定视频比特率 r_i 时, 随着上传间隔 $1/f_i$ 的增加, 实时准确率先增大后减小. 当上传间隔较小时, 上传数据量较大导致传输时延高, 从

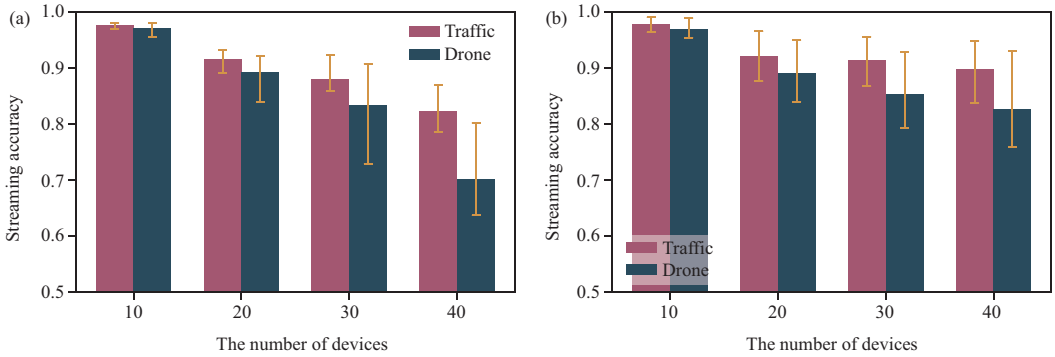

图 14 (网络版彩图) 在 **Traffic** 和 **Drone** 数据集上运行 (a) 目标检测任务和 (b) 语义分割任务的实时准确率对比

Figure 14 (Color online) Streaming accuracy of two tasks in the Traffic and Drone datasets. (a) Object detection; (b) semantic segmentation

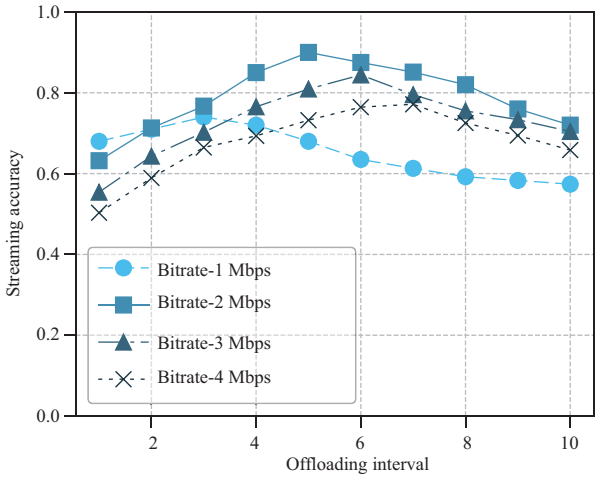

图 15 (网络版彩图) 上传间隔和视频比特率对实时准确率的影响

Figure 15 (Color online) Impact to streaming accuracy of offloading interval and video bitrate on the streaming accuracy

边缘端返回的分析滞后, 因此实时准确率下降. 随着上传间隔增大, 传输时延减少, 当上传间隔与传输时延相当时, 实时准确率达到峰值. 此时进一步增大上传间隔, 传输时延保持稳定, 则实时准确率进一步下降. 此外, 我们发现, 最优实时准确率和视频比特率 r_i 之间并不是正相关关系. 当视频比特率为 2 Mbps 时, 最优实时准确率最大. 而当视频比特率为 1 Mbps 时, 虽然最优的上传间隔很小, 边缘端的检测更加频繁, 但是视频比特率过低, 视频帧清晰度不足, 使得最优实时准确率反而更低. 而当视频比特率高于 3 Mbps 时, 增加视频帧的清晰度对于边缘服务器端检测结果的提升不再明显, 而过高的视频比特率会快速增大传输时延, 导致设备最优实时准确率的下降.

7 相关工作

边缘辅助的实时视频流分析. 在不断增长的网络边缘设备上, 分析驱动的视频应用在不断发展, 其中视频分析对设备的算力等资源要求较高, 因此将视频流上传到边缘服务器, 借助服务器算力进行

视频实时分析的技术不断发展. 如何在带宽有限的网络中高效地上传视频流, 并设计高效架构来使用服务器资源成为了这一领域主要的难题 [7]. 最近的优化工作主要分成两类: 一类是源驱动 (source-driven), 即在上传视频流到服务器之前对视频帧或者像素进行压缩来保证视频分析任务的实时性, 包括 AR 实时感知环境 [8,50]、智能摄像头实时监控 [9,39] 和机器人实时定位导航 [11,12]; 另一种是服务器驱动 (server-driven), 即根据服务器端运行的深度学习模型的结果来指导选择上传的内容 [2,34,51]. 本文提出的 MASSIVE 系统不局限于提高特定任务的表现, 而是致力于在多设备协同场景的系统设计上实现对资源的最大化利用, 并在设计空间中引入协同性而大幅提高了系统的整体性能. 因此, MASSIVE 可以和当前的系统相互补充, 进一步提高在各种应用中的实际表现.

多设备协同系统. 典型的多设备协同系统的例子包括足球机器人系统 [52~54]、无人车队系统 [55,56] 和无人机群 [57,58] 等. 同时, 很多工作致力于提升系统的智能, 通过多设备的合作来完成协同性的任务, 比如机器人的群体智能 [59,60]、多智能体增强学习 [61,62]、编队控制和协同等 [63,64]. MASSIVE 提升协同性任务中的多设备的视频流分析性能, 将对上层的各类具体协同性任务产生普遍的助力.

视频流分析的资源调度. MCDNN [65] 和 DeepDecision [35] 将资源调度作为优化问题来优化延迟和准确率等指标, 而 LAVEA [66] 和 JCAB [32] 在多个边缘节点调度计算任务来进行优化. VideoStorm [67] 和 VideoEdge [68] 在服务器集群层面讨论系统的资源分配, 而 Reducto [69] 则根据不断变化的外部条件来动态调整视频帧的过滤机制. 这些系统优化了系统资源调度, 没有结合实时视频分析系统的上传特性, 而 MASSIVE 则通过多设备的协同性调度, 达到了帕累托最优.

8 总结

在单体智能向群体智能发展的探索过程中, 实时视频流分析能力作为终端设备的重要基础能力, 某种意义上决定了群体智能在视觉感知层面的天花板. 本文讨论了多设备协同场景下的实时视频流分析系统的综合性能, 定义了评价体系和度量公式, 并对当前代表性系统进行了性能分析. 基于此, 本文提出了一个适用于多设备协同场景的实时视频流分析系统, 能够全面提升系统的综合性能, 并适用于所有基于单帧的视频流分析任务. 实验表明, MASSIVE 的性能全面地超越了代表性系统, 达到了帕累托最优, 其中准确率提升了 22.7%, 系统容量提升为 1.8 倍, 同时获得了更好的系统一致性.

参考文献

1 Li M, Wang Y X, Ramanan D. Towards streaming perception. In: Proceedings of European Conference on Computer Vision, 2020. 473–488

2 Du K, Pervaiz A, Yuan X, et al. Server-driven video streaming for deep learning inference. In: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, 2020. 557–570

3 Slowik A, Kwasnicka H. Nature inspired methods and their industry applications — swarm intelligence algorithms. IEEE Trans Ind Inf, 2018, 14: 1004–1015

4 Antão L, Pinto R, Reis J, et al. Cooperative human-machine interaction in industrial environments. In: Proceedings of the 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO), 2018. 430–435

5 Wang X, Yadav V, Balakrishnan S N. Cooperative UAV formation flying with obstacle/collision avoidance. IEEE Trans Contr Syst Technol, 2007, 15: 672–679

6 Simoni M D, Kockelman K M, Gurumurthy K M, et al. Congestion pricing in a world of self-driving vehicles: an analysis of different strategies in alternative future scenarios. Transpation Res Part C-Emerg Technol, 2019, 98:

167–185
7 Chen T Y H, Ravindranath L, Deng S, et al. Glimpse: continuous, real-time object recognition on mobile devices. In: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, 2015. 155–168
8 Chin T W, Ding R, Marculescu D. Adascale: towards real-time video object detection using adaptive scaling. 2019. ArXiv:1902.02910
9 Zhang B, Jin X, Ratnasamy S, et al. Awstream: adaptive wide-area streaming analytics. In: Proceedings of Conference of the ACM Special Interest Group on Data Communication, 2018. 236–252
10 Wang Y D, Wang W Y, Zhang J X, et al. Bridging the edge-cloud barrier for real-time advanced vision analytics. In: Proceedings of the 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19), 2019
11 Ali A J B, Hashemifar Z S, Dantu K. Edge-SLAM: edge-assisted visual simultaneous localization and mapping. In: Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services, 2020. 325–337
12 Xu J G, Cao H, Li D Y, et al. Edge assisted mobile semantic visual SLAM. In: Proceedings of IEEE Conference on Computer Communications, 2020. 1828–1837
13 Karnouskos S, Leitao P. Key contributing factors to the acceptance of agents in industrial environments. IEEE Trans Ind Inf, 2017, 13: 696–703
14 Leitao P, Marik V, Vrba P. Past, present, and future of industrial agent applications. IEEE Trans Ind Inf, 2013, 9: 2360–2372
15 Müller J P, Fischer K. Application impact of multi-agent systems and technologies: a survey. In: Proceedings of Agent-oriented Software Engineering, 2014. 27–53
16 Ismail Z H, Sariff N. A survey and analysis of cooperative multi-agent robot systems: challenges and directions. In: Proceedings of Applications of Mobile Robots, 2018
17 Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving. 2016. ArXiv:1610.03295
18 Sun Z Y. Cooperative coordination and formation control for multi-agent systems. Dissertation for Ph.D. Degree. Cnberra: The Australian National University, 2018
19 Nedic A, Ozdaglar A, Parrilo P A. Constrained consensus and optimization in multi-agent networks. IEEE Trans Autom Control, 2010, 55: 922–938
20 Cheng Y, Wang D, Zhou P, et al. Model compression and acceleration for deep neural networks: the principles, progress, and challenges. IEEE Signal Process Mag, 2018, 35: 126–136
21 Choudhary T, Mishra V, Goswami A, et al. A comprehensive survey on model compression and acceleration. Artif Intell Rev, 2020, 53: 5113–5155
22 Choi Y, El-Khamy M, Lee J. Towards the limit of network quantization. 2016. ArXiv:1612.01543
23 Yang J W, Shen X, Xing J, et al. Quantization networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2019. 7308–7316
24 Zhang Y, Chuangsuwanich E, Glass J. Extracting deep neural network bottleneck features using low-rank matrix factorization. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014. 185–189
25 Swaminathan S, Garg D, Kannan R, et al. Sparse low rank factorization for deep neural network compression. Neurocomputing, 2020, 398: 185–196
26 Lawhern V J, Solon A J, Waytowich N R, et al. EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces. J Neural Eng, 2018, 15: 056013
27 Li Z X, Song Y, Mcloughlin I, et al. Compact convolutional neural network transfer learning for small-scale image classification. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016. 2737–2741
28 Kim Y, Rush A M. Sequence-level knowledge distillation. 2016. ArXiv:1606.07947
29 Mirzadeh S I, Farajtabar M, Li A, et al. Improved knowledge distillation via teacher assistant. In: Proceedings of AAAI Conference on Artificial Intelligence, 2020. 5191–5198

30 Cheng Y, Wang D, Zhou P, et al. A survey of model compression and acceleration for deep neural networks. 2017. ArXiv:1710.09282
31 Jiang J, Ananthanarayanan G, Bodik P, et al. Chameleon: scalable adaptation of video analytics. In: Proceedings of Conference of the ACM Special Interest Group on Data Communication, 2018. 253–266
32 Wang C, Zhang S, Chen Y, et al. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics. In: Proceedings of IEEE Conference on Computer Communications, 2020. 257–266
33 Wang C, Zhang S, Qian Z, et al. Joint server assignment and resource management for edge-based MAR system. IEEE ACM Trans Netw, 2020, 28: 2378–2391
34 Liu L Y, Li H Y, Gruteser M. Edge assisted real-time object detection for mobile augmented reality. In: Proceedings of the 25th Annual International Conference on Mobile Computing and Networking, 2019
35 Ran X K, Chen H L, Zhu X D, et al. DeepDecision: a mobile deep learning framework for edge video analytics. In: Proceedings of IEEE Conference on Computer Communications, 2018. 1421–1429
36 Schmidhuber J. Deep learning in neural networks: an overview. Neural Netw, 2015, 61: 85–117
37 Choo J, Liu S. Visual analytics for explainable deep learning. IEEE Comput Grap Appl, 2018, 38: 84–92
38 Hu C, Bao W, Wang D, et al. Dynamic adaptive DNN surgery for inference acceleration on the edge. In: Proceedings of IEEE Conference on Computer Communications, 2019. 1423–1431
39 Mohammed T, Joe-Wong C, Babbar R, et al. Distributed inference acceleration with adaptive DNN partitioning and offloading. In: Proceedings of IEEE Conference on Computer Communications, 2020. 854–863
40 Zhou Z, Chen X, Li E, et al. Edge intelligence: paving the last mile of artificial intelligence with edge computing. Proc IEEE, 2019, 107: 1738–1762
41 Redmon J, Farhadi A. YOLOv3: an incremental improvement. 2018. ArXiv:1804.02767
42 Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 583–596
43 Maskin E. Nash equilibrium and welfare optimality. Rev Econ Stud, 1999, 66: 23–38
44 Caramia M, Dell'Olmo P. Multi-objective optimization. In: Proceedings of Multi-objective Management in Freight Logistics, 2020. 21–51
45 Roy R, Dehuri S, Cho S B. A novel particle swarm optimization algorithm for multi-objective combinatorial optimization problem. Int J Appl Metaheur Comput, 2011, 2: 41–57
46 Grandoni F, Ravi R, Singh M, et al. New approaches to multi-objective optimization. Math Program, 2014, 146: 525–554
47 Zhou B L, Zhao H, Puig X, et al. Semantic understanding of scenes through the ADE20K dataset. Int J Comput Vis, 2019, 127: 302–321
48 Zhu P F, Wen L Y, Bian X, et al. Vision meets drones: a challenge. 2018. ArXiv:1804.07437
49 Zhu P F, Wen L Y, Du D W, et al. Vision meets drones: past, present and future. 2020. ArXiv:2001.06303
50 Emmons J, Fouladi S, Ananthanarayanan G, et al. Cracking open the DNN black-box: video analytics with DNNS across the camera-cloud boundary. In: Proceedings of Workshop on Hot Topics in Video Analytics and Intelligent Edges, 2019. 27–32
51 Chinchali S P, Cidon E, Pergament E, et al. Neural networks meet physical networks: distributed inference between edge devices and the cloud. In: Proceedings of the 17th ACM Workshop on Hot Topics in Networks, 2018. 50–56
52 Farinelli A, Boscolo N, Zanotto E, et al. Advanced approaches for multi-robot coordination in logistic scenarios. Robot Auton Syst, 2017, 90: 34–44
53 Kim J H, Vadakkepat P. Multi-agent systems: a survey from the robot-soccer perspective. Intell Autom Soft Comput, 2000, 6: 3–17
54 Candea C, Hu H, Iocchi L, et al. Coordination in multi-agent RoboCup teams. Robot Auton Syst, 2001, 36: 67–86
55 Obdržálek Z. Mobile agents in multi-agent UAV/UGV system. In: Proceedings of International Conference on Military Technologies (ICMT), 2017. 753–759
56 Kim J H, Kwon J W, Seo J. Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to

assist path planning of UGV. Electron Lett, 2014, 50: 1431–1432
57 Rosa L, Cognetti M, Nicastro A, et al. Multi-task cooperative control in a heterogeneous ground-air robot team. IFAC-PapersOnLine, 2015, 48: 53–58
58 Thakoor O, Garg J, Nagi R. Multiagent UAV routing: a game theory analysis with tight price of anarchy bounds. IEEE Trans Autom Sci Eng, 2020, 17: 100–116
59 Brambilla M, Ferrante E, Birattari M, et al. Swarm robotics: a review from the swarm engineering perspective. Swarm Intell, 2013, 7: 1–41
60 Bayındır L. A review of swarm robotics tasks. Neurocomputing, 2016, 172: 292–321
61 Shoham Y, Powers R, Grenager T. Multi-agent reinforcement learning: a critical survey. 2003. https://jmvidal.cse.sc.edu/library/shoham03a.pdf
62 Busoniu L, Babuska R, de Schutter B. A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern C, 2008, 38: 156–172
63 Zhang Y, Mehrjerdi H. A survey on multiple unmanned vehicles formation control and coordination: normal and fault situations. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), 2013. 1087–1096
64 Oh K K, Park M C, Ahn H S. A survey of multi-agent formation control. Automatica, 2015, 53: 424–440
65 Han S, Shen H C, Philipose M, et al. MCDNN: an approximation-based execution framework for deep stream processing under resource constraints. In: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, 2016. 123–136
66 Yi S H, Hao Z J, Zhang Q Y, et al. LAVEA: latency-aware video analytics on edge computing platform. In: Proceedings of the 2nd ACM/IEEE Symposium on Edge Computing, 2017. 1–13
67 Zhang H, Ananthanarayanan G, Bodik P, et al. Live video analytics at scale with approximation and delay-tolerance. In: Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), 2017. 377–392
68 Hung C C, Ananthanarayanan G, Bodik P, et al. VideoEdge: processing camera streams using hierarchical clusters. In: Proceedings of IEEE/ACM Symposium on Edge Computing (SEC), 2018. 115–131
69 Li Y Q, Padmanabhan A, Zhao P Z, et al. Reducto: on-camera filtering for resource-efficient real-time video analytics. In: Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication, 2020. 359–376

Toward cooperative multi-agent video streaming perception

Zheng YANG*, Liang DONG & Xinjun CAI

School of Software, Tsinghua University, Beijing 100084*, China*
* Corresponding author. E-mail: hmilyyz@gmail.com

Abstract Video streaming perception ability is critical for AI applications on resource-constrained devices (agents), which prefers to offload video streams from devices to edge servers for real-time inference by deep neural networks (DNNs). Meanwhile, the multi-agent system (MAS) community is attempting to run DNNs on multiple cooperative agents to enable improved swarm intelligence-based tasks (e.g., drone swarm intelligence, self-driving fleet collaboration, and multi-agent robot cooperation). However, transferring video streaming perception capability from single-agent systems to MASs is extremely difficult due to spontaneous competition-induced trade-offs between the desired goals of accuracy, consistency, and capacity, which are three critical but conflicting measuring indexes. In this paper, we present the design and implementation of MASSIVE, an edge-assisted cooperative multi-agent video streaming perception system that simultaneously achieves all three desired goals. In our design, we consider the performance characteristics of video streaming perception and the insight of its periodic offloading pattern. On this basis, we develop a Pareto improvement scheduler to eliminate spontaneous competition among agents, allowing multi-objective optimization to achieve an ideal Pareto optimal state. Finally, we propose a virtual traffic shaper based on the mainstream 802.11 MAC protocol to ensure deterministic periodic video stream offloading in an uncertain wireless network. Our experiments demonstrate that MASSIVE achieves 122.7% accuracy and 1.8x capacity compared to the closest baseline on multiple actual cooperative vision tasks with even better consistency, and achieves an ideal Pareto optimal state in a wireless environment.

Keywords real-time video analysis, edge computing, multi-agent cooperation, multi-objective optimization, Pareto optimal state