

基于边缘计算的多摄像头视频协同分析方法

期治博^{1,2}, 杜磊³, 霍如^{3,4}, 杨帆^{1,4}, 黄韬^{1,4}

(1. 北京邮电大学网络与交换国家重点实验室, 北京 100876; 2. 中国信息通信研究院工业互联网与物联网研究所, 北京 100083;
3. 北京工业大学信息学部, 北京 100124; 4. 网络通信与安全紫金山实验室, 江苏 南京 211111)

摘要: 为了减少智慧城市市场景下多摄像头实时视频数据的处理量, 提出了基于机器学习算法的边缘端视频协同分析方法。首先, 针对各摄像头检测到的重要目标物体, 设计了不同的关键窗口来筛选视频的感兴趣区域, 缩减视频数据量并提取其特征。然后, 根据提取的数据特征, 对不同摄像头视频中的相同目标物体进行标注, 并设计了摄像头之间关联程度值的计算策略, 用于进一步缩减视频数据量。最后, 提出了基于图卷积网络和重识别技术的 GC-ReID 算法, 旨在实现多摄像头视频协同分析。实验结果表明, 与现有的视频分析方法相比, 所提方法能够有效降低系统时延和提高视频压缩率, 并保证较高的准确率。

关键词: 边缘计算; 机器学习; 视频协同分析; 感兴趣区域标注; 多摄像头关联性

中图分类号: TN919.85

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2023150

Multi-camera video collaborative analysis method based on edge computing

QI Zhibo^{1,2}, DU Lei³, HUO Ru^{3,4}, YANG Fan^{1,4}, HUANG Tao^{1,4}

1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
2. Department of Industrial Internet Institute, China Academy of Information and Communication, Beijing 100083, China
3. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
4. Purple Mountain Laboratories, Nanjing 211111, China

Abstract: In order to reduce the processing volume of multi-camera real-time video data in smart city scenarios, a video collaborative analysis method based on machine learning algorithms at the edge was proposed. Firstly, for the important objects detected by each camera, different key windows were designed to filter the region of interest (RoI) in the video, reduce the video data volume and extract its features. Then, based on the extracted data features, the same objects in the videos from different cameras were annotated, and a strategy for calculating the association degree value between cameras was designed for further reducing the video data volume. Finally, the GC-ReID algorithm based on graph convolutional network (GCN) and re-identification (ReID) was proposed, aiming at achieving the collaborative analysis of multi-camera videos. The experimental results show that proposed method can effectively reduce the system latency and improve the video compression rate while ensuring the high accuracy, compared with the existing video analysis methods.

Keywords: edge computing, machine learning, video collaborative analysis, region of interest annotation, association between cameras

收稿日期: 2023-03-17; 修回日期: 2023-07-10

通信作者: 霍如, huoru@bjut.edu.cn

基金项目: 国家重点研发计划基金资助项目 (No.2018YFB1800602); 2020 年工业互联网创新发展工程基金资助项目 (工业互联网标识资源搜索系统)

Foundation Items: The National Key Research and Development Program of China (No.2018YFB1800602), The MIIT of China 2020 (Identification Re-source Search System for Industrial Internet of Things)

0 引言

随着新一代信息技术的发展, 城市的智能化程度不断提升, 智慧城市迅速崛起并成为城市发展的新理念和方向。智慧城市利用物联网、云计算、边缘计算和人工智能等新型计算技术, 将城市规划、设计、建设、管理和运营等领域智能化, 实现更加高效的运行。智慧城市的发展使城市监控摄像头的数量急剧增加, 这对城市建设和安防带来了显著改善。然而, 现有的城市视频监控设备是各自独立的, 不利于城市建设中大数据分析的整合^[1]。因此, 研究多摄像头视频协同处理, 从庞大的视频数据中挖掘价值, 成为未来城市互联监控的发展方向 and 重点。多摄像头视频数据的处理也是许多视觉智能应用的基础, 如物体检测和跟踪、物体视觉定位、物体速度估计等。快速发展的网络技术对这些智能应用提供了支持并使之成为热点^[2]。因此, 多摄像头视频分析引起了众多学者的关注。

现有的多摄像头视频分析方法大多基于独立视频流进行处理^[3]。具体而言, 这些方法适用于同一场景下多个摄像头同时采集不同角度的视频, 然而在对这些采集的视频进行分析时, 主要针对每个摄像头的单个视频流进行处理。为了在成本和精度之间进行权衡, 通常采用帧采样或级联滤波器去除较不重要的视频帧以减少视频量, 从而降低成本。然而, 所有的调整和压缩操作都是在单个视频流内进行的, 不同摄像头采集的视频处理是相互独立的, 这导致了视频分析中计算和存储资源的线性增加。因此, 基于这些视频分析方法, 多摄像头视频分析需要较大的计算、网络和存储资源。

为了提高实时多摄像头视频分析的效率并降低处理时延, 可以在边缘端对多摄像头视频进行协同分析。它通过构建多个摄像头之间的关联性来减少待分析视频的数据量, 缩减后的视频可以在边缘端或传输到云端进行进一步的处理, 从而降低整体视频数据的冗余度, 减轻边缘端和云端视频处理的工作量, 进而减少处理时延。同时相较于上述独立视频流分析, 这种方法对计算能力和存储资源等方面的要求较低。

因此, 实现多摄像头视频的协同分析主要涉及以下2个问题。1) 实时的视频处理方法。实时视频处理需要丰富的计算、网络和存储资源, 但摄像头

本身的视频处理能力有限, 难以支持大规模且实时的视频数据检测和分析。因此, 考虑利用配备边缘服务器来扩展摄像头的视频处理能力是必要的。将视频传输到边缘服务器进行处理, 利用边缘计算将云端计算能力下沉到边缘端, 可以减轻摄像头的负担, 并降低计算和传输时延。2) 多个摄像头之间的关联性。在相同时段和场景下, 不同角度摄像头所拍摄的视频片段之间存在一定的关联性。这可能包括从不同角度拍摄了同一个目标的不同面, 或者同一个目标在不同摄像头的视频中先后出现。由于不同摄像头的空间布局和视角关系通常很复杂, 因此需要计算和处理不同视频片段之间的关联性, 并作为后续视频压缩处理的依据。

为了应对上述问题, 本文提出了一种边缘端多摄像头视频协同分析方法。该方法基于机器学习算法, 设计关键窗口模型和多摄像头关联程度计算策略, 为后续的视频数据量缩减和进一步的视频处理做准备。本文主要的研究工作总结如下。

1) 提出了一种新颖的多摄像头视频协同分析方法。该方法首先对每个摄像头采集的视频进行目标检测; 然后根据目标检测的结果生成关键窗口, 对视频进行特征提取和初步筛选; 随后结合提取的特征利用重识别技术对检测到的目标进行标注, 并根据标注结果计算多摄像头之间的关联程度值; 最后基于关联程度值对视频进行压缩处理。

2) 设计了用于表示不同摄像头视频帧中感兴趣区域 (RoI, region of interest) 的关键窗口。首先对视频帧进行分块处理, 然后基于包括重要目标的区域块构建关键窗口。同时本文提出多摄像头关联性计算策略, 基于摄像头关键窗口中目标的标注结果, 计算不同摄像头之间的关联程度。

3) 提出了一种 GC-ReID 算法来实现多摄像头视频协同分析方法。GC-ReID 结合图卷积网络 (GCN, graph convolutional network) 和重识别 (ReID, re-identification) 技术, 利用 GCN 对各摄像头中的重要目标进行快速检测, 并结合关键窗口迅速提取视频数据特征。同时, 利用 ReID 对关键窗口的数据特征进行快速标注, 结合计算策略得到多摄像头之间的关联程度值。

1 相关工作

对于引言中所讨论的问题, 近期已有学者进行

了相关的研究。其中,一些学者结合不同的视频处理策略和边缘计算的能力提出了在边缘服务器上进行视频分析的方法;另一些学者在结合边缘计算技术的基础上,研究视频分析中多个摄像头之间的资源竞争;此外,还有学者关注多个摄像头检测到的目标之间的匹配关联性。本节对这些相关研究进行简要回顾和分析。

现有的视频分析研究主要利用边缘或云的计算能力进行视频分析和处理,以从视频中提取关键特征和有效信息。Yang 等^[3]提出了基于检测的跟踪方法,适用于边缘端高分辨率视频的分析。赵羽等^[4]提出了一种部署在边缘端的视频处理方法,利用轻量级的神经网络和联邦学习分场景训练模型。Gao 等^[5]提出了一种针对云边缘视频流的智能视频处理架构,在云端部署并执行基于深度神经网络(DNN, deep neural network)的视频增强方法。然而,这些研究仅考虑对单个摄像头采集的视频进行分析,而未涉及跨摄像头和多摄像头的视频分析和处理。

关于视频分析中多个摄像头的资源竞争研究主要关注摄像头视频的任务卸载和计算资源分配。Chen 等^[6]研究了多服务器和多用户的异质视频分析任务卸载问题,基于博弈论进行卸载算法设计,最终使多服务器多用户的视频分析任务卸载达到纳什均衡。Yang 等^[7]提出了终端、边缘和云协同的实时视频分析方法,通过对终端摄像头进行资源分配,使其能够根据实时视频生成高质量的视频帧。Wang 等^[8]研究了多个视频流在同一边缘节点上共享上传链路时的配置自适应和带宽分配问题,提出了联合优化配置适应和带宽分配的在线算法。Ran 等^[9]提出了在边缘端部署算力更强的服务器,利用深度学习模型来确定终端视频分析任务的最佳卸载策略,并以时延和准确率等作为目标来优化资源调度。Hung 等^[10]通过引入主导需求来确定多种资源和准确性之间的平衡,并利用层次聚类缩小搜索空间,同时平衡了视频查询的资源效益和准确性损失。上述多摄像头视频分析方法主要研究边缘系统下多摄像头视频分析任务的分配和卸载,而对于摄像头本身视频数据进行协同处理的方法较少,缺乏对不同摄像头视频数据之间关系的细致分析,并且对视频画面之间的关联程度考虑有限。

一些学者研究了多个摄像头检测到的目标之

间的匹配关联性,并提出了进行协同视频处理的方法。Guo 等^[11]提出了多人物多摄像头实时视频跟踪框架,设计了基于时间的运动模型研究视频采样帧之间的精确时间间隔,并利用改进的多人匹配级联方案提高摄像头间人员再识别的准确性。Chung 等^[12]提出了一种多摄像头多目标车辆跟踪系统,利用车辆行为、合成数据和多种增强技术执行视频数据的上下文约束,从而减少跨摄像头的分析时间。Li 等^[13]提出了一种车辆跟踪系统,包括基于特征提取的车辆检测和再识别模型、单摄像机跟踪、增强轨迹预测和多级关联方法。Liu 等^[14]提出了一种基于遮挡感知的多目标多摄像头跟踪框架,设计了目标间信息模块提高匹配精度,并利用目标间关联发掘轨迹相似性,最后通过聚类对不同摄像头之间的目标轨迹进行匹配。Li 等^[15]设计了视频分析系统 Reducto,根据不同的外部条件自定义滤波器对多摄像头的视频帧进行过滤,并通过丢弃视频片段中的帧来权衡传输成本和准确性。尽管上述多摄像头协同分析方法考虑到了摄像头检测到目标的匹配和关联度,但缺少对摄像头本身之间关联性的考虑,导致关键视频帧的筛选力度不足,甚至可能会错失关键目标物体。因此,挖掘多个摄像头之间的实时关联并实现更高效的协同视频分析具有更重要的意义。

2 多摄像头视频数据协同分析模型构建

本节提出了一种多摄像头视频协同分析(MVCA, multi-camera video collaborative analysis)方法。如图 1 所示,该方法适用于同一环境中配备有多个摄像头的场景。这些摄像头从不同的角度和方向采集场景视频,并将其传输到边缘服务器。边缘服务器对接收到的视频进行分析和处理,其中包括使用 GCN 算法建立关键窗口,利用 ReID 算法的结果计算多摄像头之间的关联程度。关联程度值可以作为对视频进行压缩的参考依据。最后,将压缩后的视频在边缘端或者传输到云服务器进行深入分析和处理。这种方法大大减少了视频处理数据量,降低了系统时延,并减轻了边缘服务器和云服务器的负担,有助于最大化边缘系统的整体效用。GCN 和 ReID 的构建和训练在云服务器离线完成,然后分别部署到直连边缘服务器和主边缘服务器上。

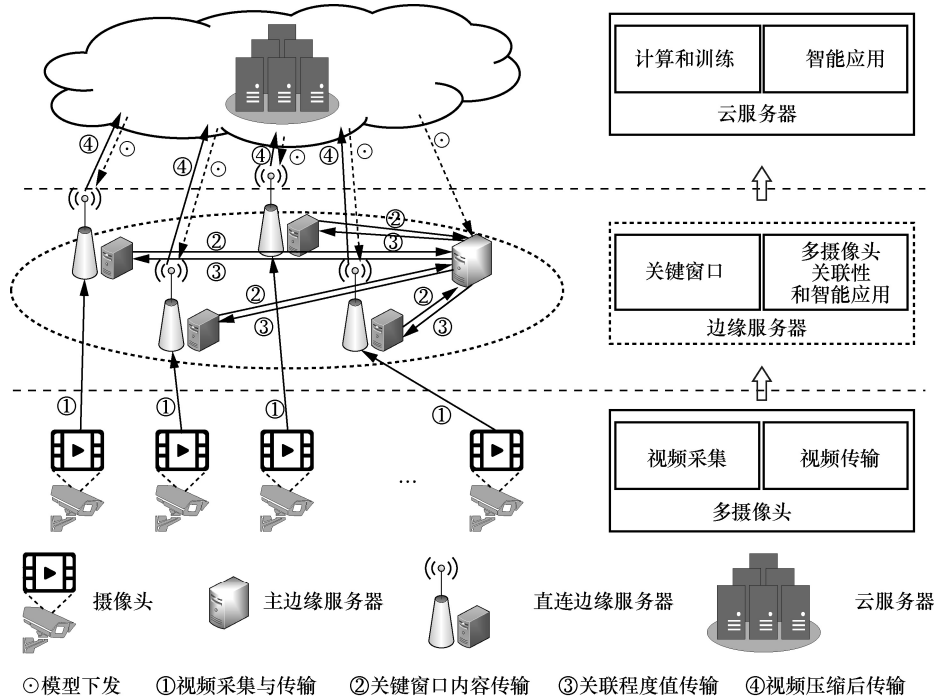


图 1 MVCA 方法框架

2.1 边缘系统模型

本文在边缘侧部署了多个摄像头，以捕获多个角度的实时视频数据。摄像头集合表示为 $N = \{n_1, n_2, n_3, \dots, n_x\}$ ，其中， x 为摄像头的数量。边缘服务器由一个主边缘服务器 Ms 及与摄像头直接连接的直连边缘服务器 $M = \{m_1, m_2, m_3, \dots, m_x\}$ 组成。每个摄像头都与一个直连边缘服务器相连。主边缘服务器用于计算多个摄像头之间的关联程度，而直连边缘服务器用于生成关键窗口。压缩后的视频可以在主边缘服务器或者由直连边缘服务器传输到云服务器进行进一步分析，如图 1 所示。通过权衡视频分析开销和分析粒度，本文对若干时间段的视频数据进行了分析，每个时间段 T 的最小分析单元为一个时隙 t ， $t \in T = \{t_0, t_1, t_2, \dots, t_y\}$ ， y 为时间段 T 中最后一个时隙的编号。在每个时间段 T 的开始，不同摄像头拍摄的视频以一个时隙为单位进行分割并依次上传至直连边缘服务器。边缘系统模型的参数和含义如表 1 所示。

1) 通信模型

如上文所述，多个摄像头 N 拍摄的视频将同时传输到各自的直连边缘服务器进行进一步处理。因此，本文使用频分多址 (FDMA, frequency division multiple access) 技术来处理多摄像头的视频传输任务。根据香农定理，可以得到摄像头 n_z ($0 \leq z \leq x$,

x 表示摄像头的数量) 的传输速率为

$$r_{n_z} = W_t^{n_z} \text{lb} \left(1 + \frac{P_{n_z} G_{n_z}}{\sigma^2} \right) \quad (1)$$

其中， $W_t^{n_z}$ 表示摄像头 n_z 在时隙 t 的上行带宽， P_{n_z} 表示摄像头 n_z 的传输功率， G_{n_z} 表示从摄像头 n_z 到直连边缘服务器的信道增益， σ^2 表示背景噪声方差。

表 1 边缘系统模型的参数和含义

参数	含义
N	摄像头集合
Ms	主边缘服务器
M	直连边缘服务器集合
T	时间段，组成单元为时隙
r_{n_z}	摄像头 n_z 的传输速率
r_{m_z}	直连边缘服务器 m_z 的传输速率
W_t	摄像头 n_z 在 t 时隙的上行带宽
P_{n_z}	摄像头 n_z 的传输功率
G_{n_z}	摄像头 n_z 到直连边缘服务器的信道增益
σ^2	背景噪声方差
$D_{n_z}^{\text{tran}}$	摄像头 n_z 到直连边缘服务器的传输时延
$D_{n_z}^{\text{tran}}$	直连边缘服务器 n_z 到主边缘服务器或云服务器的传输时延
$D_{n_z}^{\text{com}}$	计算时延

与一些现有的研究^[16-17]类似, 本文所提方法中忽略了主边缘服务器在得到多摄像头关联程度值后返回结果到各直连边缘服务器的时间成本。这是因为在视频分析和处理过程中, 传输和分析结果所需要的时间代价比传输和处理视频数据所需要的时间代价要小很多。因此, 当摄像头 n_z 将拍摄的视频单元传输到相应的边缘服务器时, 其传输时延 $D_{n_z}^{\text{tran}}$ 为

$$D_{n_z}^{\text{tran}} = \frac{u_{n_z}^t v_u}{r_{n_z}} \quad (2)$$

其中, $u_{n_z}^t$ 表示摄像头 n_z 在时隙 t 传输到直连边缘服务器的视频单元数量, v_u 表示每个视频单元中的数据大小 (单位为字节)。

虽然在本文中忽略了主边缘服务器返回结果到各直连边缘服务器的时间成本, 但是直连边缘服务器会将生成的关键窗口区域和检测到的特征传输到主边缘服务器和云服务器, 因此本文也采用 FDMA 来处理多个直连边缘服务器的数据传输任务。同样根据香农定理, 可以得到直连边缘服务器 m_z ($0 \leq z \leq x$, x 表示直连边缘服务器的数量, 也是摄像头的数量) 的传输速率为

$$r_{m_z} = W_t^{m_z} \text{lb} \left(1 + \frac{P_{m_z} (aG_{m_z}^{\text{main}} + bG_{m_z}^{\text{cloud}})}{\sigma^2} \right) \quad (3)$$

其中, $W_t^{m_z}$ 表示直连边缘服务器 m_z 在时隙 t 的上行带宽; P_{m_z} 表示直连边缘服务器 m_z 的传输功率; $G_{m_z}^{\text{main}}$ 表示从直连边缘服务器 m_z 到主边缘服务器的通道增益; $G_{m_z}^{\text{cloud}}$ 表示从直连边缘服务器 m_z 到云服务器的通道增益; 当直连边缘服务器传输数据到主边缘服务器时, $a=1$ 且 $b=0$, 当直连边缘服务器传输数据到云服务器时, $a=0$ 且 $b=1$; σ^2 表示背景噪声方差。

同样, 当直连边缘服务器 m_z 将关键窗口和提取的特征数据传输到主边缘服务器或云服务器时, 其传输时延 $D_{m_z}^{\text{tran}}$ 为

$$D_{m_z}^{\text{tran}} = \frac{u_{m_z}^t v_u}{r_{m_z}} \quad (4)$$

其中, $u_{m_z}^t$ 表示直连边缘服务器 m_z 在时隙 t 传输到主边缘服务器或者云服务器的视频单元数量, v_u 表示每个视频单元中的数据大小 (单位为字节)。

2) 计算模型

与现有研究^[8]类似, 本文使用虚拟并行处理技术^[18]来支持同时执行多个摄像头视频分析, 从而实

现对多个视频进行并行处理。在计算模型中, 摄像头 n_z 所采集视频数据中每个字节的计算需求为 C_r (CPU 周期)。因此, 直连边缘服务器和主边缘服务器在处理视频时的计算时延 $D_{n_z}^{\text{com}}$ 为

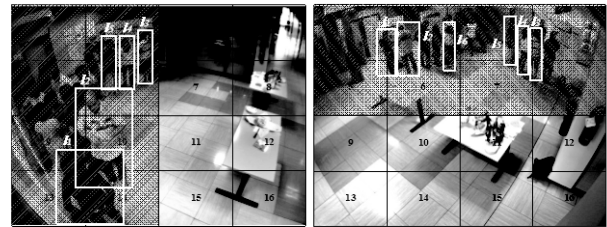
$$D_{n_z}^{\text{com}} = \frac{u_{n_z}^t v_u C_r}{F_e} \quad (5)$$

其中, F_e 表示执行视频处理的边缘服务器的 CPU 频率。本文设定所提方法中边缘系统的总时延 D_{total} 由传输时延和计算时延组成, 即

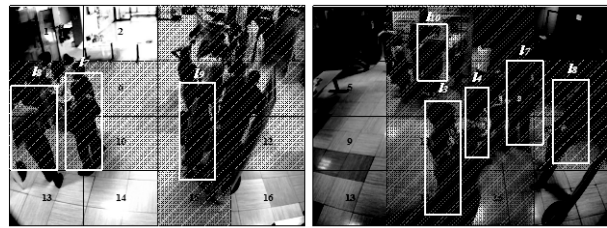
$$D_{\text{total}} = D_{n_z}^{\text{tran}} + D_{m_z}^{\text{tran}} + D_{n_z}^{\text{com}} \quad (6)$$

2.2 感兴趣区域与关键窗口建立

为了便于发现不同摄像头之间的视频数据关联性, 本文设计了一种对摄像头采集的视频帧进行分块的方法。具体而言, 摄像头 n_z 拍摄的视频帧被划分为大小相同的块状区域。这些块状区域是相对较小的矩形, 它们共同构成并覆盖了视频帧的整个画面。如图 2 所示, 每个摄像头采集的视频帧画面被划分为 16 个矩形区域块, 按照从左到右和从上到下的顺序, 索引分别为 1~16。摄像头 n_z 的区域块集合用 R_{n_z} 表示, 其中第 i 个区域块被表示为 $R_{n_z,i}$ 。如图 2(a) 所示, 摄像头 1 采集视频帧的左上角, 其中第一行的区域块可以表示为 $R_{1,1}$ 、 $R_{1,2}$ 、 $R_{1,3}$ 和 $R_{1,4}$, 其余的区域块依次类推。需要注意的是, 视频帧中的每个区域块并不特定于某一帧或某一时隙, 进行区域分块是为了能够更精确地划分摄像头所采集的视频帧画面。



(a) 摄像头 1 采集的视频帧画面 (b) 摄像头 2 采集的视频帧画面



(c) 摄像头 3 采集的视频帧画面 (d) 摄像头 4 采集的视频帧画面

图 2 摄像头 1~4 在时隙 t 内采集的视频帧画面

在本文提出的 MVCA 方法中, 首先需要使用机器学习算法对不同摄像头采集的视频帧进行快速检测, 检测结果中可能包含目标物体的区域块即感兴趣区域, 例如, 移动的人或物体都属于重要目标物体。这些感兴趣区域构成了关键窗口。在视频分析过程中, 本文优先处理关键窗口内的区域块, 因为这些区域块包含目标物体, 而关键窗口以外的区域块动态性较差且缺乏目标物体, 所以处理优先级较低, 甚至被忽略而不进行分析和处理。在本文中, 构成关键窗口的最小元素是以上所设计的区域块。摄像头 n_z 的关键窗口用 K_{n_z} 表示, 即 K_{n_z} 是包含目标物体的所有区域块的集合。例如, 图 2(b)展示了摄像头 2 的关键窗口, 即斜纹阴影标注的区域块, 包括区域块 1~8。这些区域块是通过目标检测选择得到的, 共同构成了摄像头 2 的关键窗口。因此, 图 2 中摄像头 1~4 的关键窗口分别表示为

$$K_1 = \{R_{1,1}, R_{1,2}, R_{1,5}, R_{1,6}, R_{1,9}, R_{1,10}, R_{1,13}, R_{1,14}\} \quad (7)$$

$$K_2 = \{R_{2,1}, R_{2,2}, R_{2,3}, R_{2,4}, R_{2,5}, R_{2,6}, R_{2,7}, R_{2,8}\} \quad (8)$$

$$K_3 = \{R_{3,3}, R_{3,4}, R_{3,5}, R_{3,6}, R_{3,7}, R_{3,8}, R_{3,9}, R_{3,10}, R_{3,11}, R_{3,12}, R_{3,15}\} \quad (9)$$

$$K_4 = \{R_{4,2}, R_{4,3}, R_{4,6}, R_{4,7}, R_{4,8}, R_{4,10}, R_{4,11}, R_{4,12}, R_{4,14}, R_{4,15}\} \quad (10)$$

在进行视频数据处理时, 为每个摄像头设计一个包含最少区域块的关键窗口非常重要, 特别是当人们希望关键窗口适用于较长的分析时间时。这样在对视频进行处理时, 只需要分析关键窗口中的区域就能够提取到视频数据中的关键信息, 从而有效减少了视频数据的处理量。本文提出的 GC-ReID 利用 GCN^[19]快速检测视频中的目标物体用以生成关键窗口, 更详细的介绍见第 3 节。

2.3 多摄像头关联性构建

为了进一步减少视频数据的分析量, 本文设计了多摄像头关联性的构建方法。其主要原理是采用机器学习算法对于关键窗口中的目标物体进行检测并提取特征, 然后对提取的特征进行分类。根据分类结果为关键窗口中每个检测到的物体分配一个 ID 号。然而, 由于这些摄像头处于同一场景下, 不同摄像头采集的视频帧往往会捕捉到相同的物体。因此, 通过分类算法, 同一物体将被赋予相同的 ID 号。例如, 图 2(a)和图 2(b)分

别表示在同一环境中 2 个不同的摄像头拍摄的画面。利用 GC-ReID 中的 ReID 算法, 可以识别 2 个画面中相同的人物, 为其分配相同的 ID 号。假设同一场景下同一时隙检测到的目标物体的集合为 $L = \{l_1, l_2, l_3, \dots, l_o\}$ 。虽然角度不同, 但是人物 l_1 、 l_2 、 l_3 、 l_4 和 l_5 同时出现在摄像头 1 和摄像头 2 的视频帧中, 即他们在 2 个摄像头中都被检测到。然而, 人物 l_6 只出现在图 2(b)中, 即只被摄像头 2 检测到。

根据 GC-ReID 算法对整个时间段 T 的处理结果得到每个摄像头的关键窗口, 然后, 根据这些关键窗口的内容继续发掘不同摄像头之间的关联性。目标物体 l_j ($1 \leq j \leq o$, o 表示目标物体数量) 在任意时隙 t 出现在摄像机 n_z 的关键窗口中的最小区域块的集合表示为 $rs_{n_z,t}^j$ ($rs_{n_z,t}^j \subseteq K_{n_z}$)。然而, 由于目标物体 l_j 可能出现在多个摄像头的关键窗口中, 因此将在时隙 t 多个摄像头画面中有目标物体 l_j 出现的区域块集合表示为 RS_t^j , 即

$$RS_t^j = \{rs_{n_z,t}^j \mid 1 \leq z \leq x, rs_{n_z,t}^j \neq \emptyset\} \quad (11)$$

在图 2 中, 共有 10 个目标物体在时隙 t 被同一场景下配备的摄像头检测到, 表示为 $L_t = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}\}$ 。 L_t 表示检测到的目标物体的集合, 其中 l_1 、 l_2 和 l_5 出现在摄像头 1 和摄像头 2 的两帧中。它们出现的区域块集合分别为

$$\begin{aligned} RS_t^1 &= \{\{R_{1,9}, R_{1,10}, R_{1,13}, R_{1,14}\}, \{R_{2,1}, R_{2,2}, R_{2,5}, R_{2,6}\}\} \\ RS_t^2 &= \{\{R_{1,5}, R_{1,6}, R_{1,9}, R_{1,10}, R_{1,14}\}, \{R_{2,2}, R_{2,6}\}\} \\ RS_t^5 &= \{\{R_{1,2}, R_{1,6}\}, \{R_{2,5}, R_{2,7}\}\} \end{aligned} \quad (12)$$

而目标物体 l_6 只能被摄像头 2 检测到。因此, RS_t^6 可以表示为

$$RS_t^6 = \{\{R_{2,2}, R_{2,6}\}\} \quad (13)$$

依次类推, 表 2 中列出了 10 个所检测到目标物体的所有区域块集合 RS_t^j 。

根据表 2 可以观察到, 不同摄像头之间检测到相同目标物体的数量存在差异。当 2 个摄像头检测到相同目标物体数量较多时, 可以认为它们之间的关联性较强。为了便于分析不同摄像头之间的关联性, 本文使用式(14)来计算关联程度值 AS。

表 2

图 2 中区域块的集合

时隙	检测到的目标物体	区域块集合
t	$L_t = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}\}$	$RS_t^1 = \{\{R_{1,9}, R_{1,10}, R_{1,13}, R_{1,14}\}, \{R_{2,1}, R_{2,2}, R_{2,5}, R_{2,6}\}\}$ $RS_t^2 = \{\{R_{1,5}, R_{1,6}, R_{1,9}, R_{1,10}, R_{1,14}\}, \{R_{2,2}, R_{2,6}\}\}$ $RS_t^3 = \{\{R_{1,2}, R_{1,6}\}, \{R_{2,3}, R_{2,4}, R_{2,7}, R_{2,8}\}, \{R_{4,6}, R_{4,10}, R_{4,14}\}\}$ $RS_t^4 = \{\{R_{1,2}, R_{1,6}\}, \{R_{2,5}, R_{2,7}\}, \{R_{4,7}, R_{4,11}\}\}$ $RS_t^5 = \{\{R_{1,2}, R_{1,6}\}, \{R_{2,5}, R_{2,7}\}\}$ $RS_t^6 = \{\{R_{2,2}, R_{2,6}\}\}$ $RS_t^7 = \{\{R_{3,5}, R_{3,6}, R_{3,9}, R_{3,10}\}, \{R_{4,7}, R_{4,8}, R_{4,11}, R_{4,12}\}\}$ $RS_t^8 = \{\{R_{3,5}, R_{3,9}\}, \{R_{4,8}, R_{4,12}\}\}$ $RS_t^9 = \{\{R_{3,7}, R_{3,11}, R_{3,15}\}\}$ $RS_t^{10} = \{\{R_{4,2}, R_{4,6}\}\}$

$$AS(n_1, n_2, n_3, \dots, n_x) = \sum_o s$$

$$\text{s.t.} \begin{cases} s = 1, \exists R_{n_1}, R_{n_2}, \dots, R_{n_x} \in RS_t^j \\ s = 0, \text{其他} \end{cases} \quad (14)$$

其中, $(n_1, n_2, n_3, \dots, n_x)$ 代表不同的摄像机, o 代表时隙 t 检测到的目标物体 L_t 的数量。例如, 在计算摄像头 1 和摄像头 2 的关联值的过程中, 对于所有检测到的目标物体, 其所处区域块 $R_{1,*}$ 和 $R_{2,*}$ 都同时属于 RS_t^1 、 RS_t^2 、 RS_t^3 、 RS_t^4 和 RS_t^5 , 即同时出现在摄像头 1 和摄像头 2 视频帧中的目标物体有 l_1 、 l_2 、 l_3 、 l_4 和 l_5 , 因此根据式(14)可以计算得到摄像头 1 和摄像头 2 的关联值为 5。

3 多摄像头视频数据协同分析方法实现

本文设计了 GC-ReID 算法用于实现多摄像头视频协同分析方法。如图 3 所示, 各摄像头直连的边缘服务器并行执行 GCN 算法, 从多个摄像头采集的视频数据中筛选关键特征, 并根据这些特征建立关键窗口模型。随后, 直连边缘服务器将关键窗口的内容传输到主边缘服务器。主边缘服务器使用 ReID 算法, 将 GCN 的结果和提取的关键窗口的内容作为输入, 对关键窗口中检测到的目标物体进行筛选并分配 ID 号。根据 ReID 算法的处理结果, 计算不同摄像头之间的关联程度值。随后将关联程度值返回到直连边缘服务器, 根据这些关联程度值对多个摄像头采集的视频进行压缩和优化。最后, 压缩后的视频可以选择在主边缘服务器或者传输到云服务器上进行进一步分析和处理, 例如目标检测和目标跟踪

等。GCN 和 ReID 算法的构建和训练均事先在云服务器完成并下发到边缘服务器。

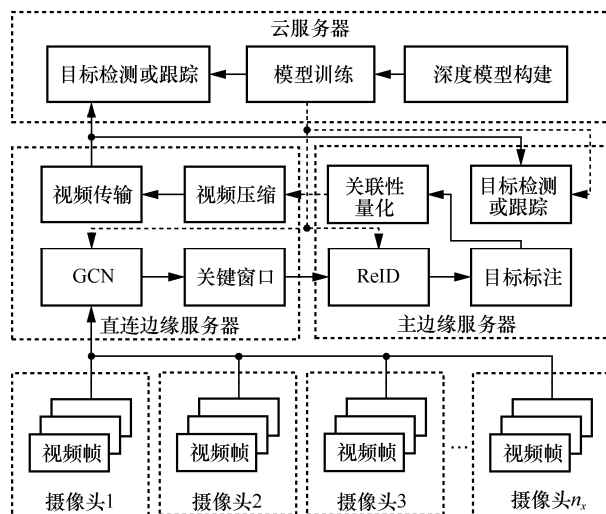


图 3 GC-ReID 的总体流程

3.1 关键窗口的建立

本文采用 GCN 提取摄像头采集的视频帧的关键特征, 并检测其中的目标物体。最后, 根据包含这些目标物体的区域块构建每个摄像头的关键窗口, 详细结构如图 4 所示。

图 4 中的输入是连续的视频帧片段, 由摄像头采集并上传到直连边缘服务器。输入 GCN 模块后先使用两层卷积层初步提取视频帧的特征。然后将处理后的连续视频帧建模为图 $G = \{V, E\}$ 。其中, 节点集合为 $V = \{v_1, v_2, \dots, v_z\}$, z 表示节点的数量; 边集合为 $E = E_{temp} \cup E_{sema}$, E_{temp} 表示时间边, E_{sema} 表示语义边。一个节点表示一个视频帧, 一条

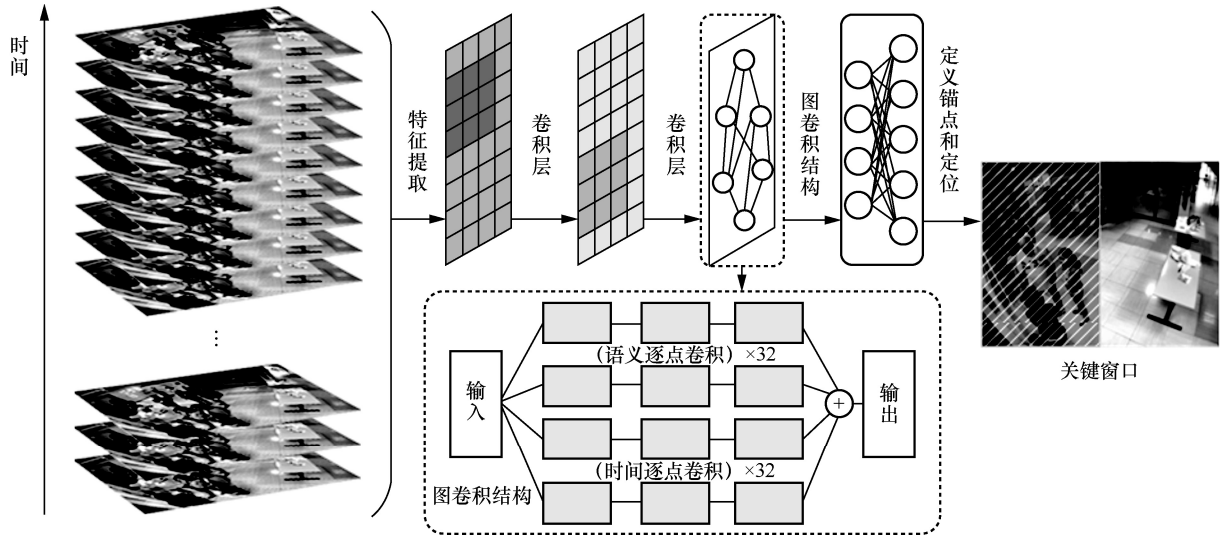


图 4 GCN 的结构

边表示一对节点之间的依赖关系。随后使用图卷积结构块提取特征。图卷积结构块包含 2 个图卷积流：时间流操作固定的时态相邻特征，语义流自适应地将语义上下文聚合到视频帧特征中。下面将描述 2 种卷积流边的设计以及图卷积过程。

1) 时间边

时间边 E_{temp} 的目标是对视频帧按照时间顺序进行编码。每个节点 $v_i \in V$ 都有一个到节点 v_{i+1} 的前向边和一个到节点 v_{i-1} 的后向边。因此， $E_{temp} = E_{temp}^{fw} \cup E_{temp}^{bw}$ 。其中， E_{temp}^{fw} 和 E_{temp}^{bw} 分别表示前向和后向的时间边集合，如式(15)和式(16)表示。

$$E_{temp}^{fw} = \{(v_i, v_{i+1}) | i \in \{1, 2, \dots, z-1\}\} \quad (15)$$

$$E_{temp}^{bw} = \{(v_i, v_{i-1}) | i \in \{2, \dots, z-1, z\}\} \quad (16)$$

2) 语义边

语义边 E_{sema} 是由动态边卷积的概念定义的^[20]，它是根据图节点之间的特征距离动态构建的。语义边的作用是从语义相关的视频帧中收集信息。图 G 中每个节点 v_i 的语义边 E_{sema} 的定义为

$$E_{sema} = \{(v_i, v_{nr_i(k)}) | i \in \{1, 2, \dots, z\}; k \in \{1, 2, \dots, K\}\} \quad (17)$$

其中， $nr_i(k)$ 表示节点 v_i 的第 k 个近邻节点索引， K 表示近邻节点索引数量。 $nr_i(k)$ 是在图卷积块上动态确定的，通过它能够更新整个图卷积块中本质上携带语义上下文信息的节点。因此， E_{sema} 可以自适应地改变以表示新的语义上下文级别。

3) 图卷积和流聚合

图 G 中所有节点的特征表示为 $Y = [y_1, y_2, \dots, y_z] \in \mathbb{R}^{c \times z}$ ，其中， c 表示每个视频帧的特征维度。本文使用单层边卷积^[20]来进行图卷积操作 GC。

$$GC(Y, A, W) = ([Y^T, AY^T - Y^T]W)^T \quad (18)$$

其中， $W \in \mathbb{R}^{2c \times c'}$ 表示可训练的权重矩阵， $A \in \mathbb{R}^{z \times z}$ 表示不带自环（节点和它本身之间的边）的邻接矩阵， $[*, *]$ 表示列的矩阵串联。本文将 A 中的第 (i, j) 个元素表示为 $A(i, j) = 1_{\{(v_i, v_j) \in E\}}$ ，其中， $1_{\{*\}}$ 是指示函数。图卷积结构模块中的 2 个卷积流都分别设计了 32 条路径来增加转换的多样性。

图卷积结构模块的输出是语义流和时间流以及输入的聚合，可以用式(19)表示。

$$H(Y, A, W) = \text{ReLU}(GC(Y, A_{temp}^{fw}, W_{temp}^{fw})) + GC(Y, A_{temp}^{bw}, W_{temp}^{bw}) + GC(Y, A_{sema}, W_{sema}) + Y \quad (19)$$

其中， A_{temp}^{fw} 、 A_{temp}^{bw} 和 A_{sema} 是邻接矩阵； $W = \{W_{temp}^{fw}, W_{temp}^{bw}, W_{sema}\}$ 是可训练权重，分别与 E_{temp}^{fw} 、 E_{temp}^{bw} 和 E_{sema} 相对应；ReLU 是激活函数的线性整流单元。

2 种图卷积流都遵循分裂-转换-合并的策略，首先，使用多个卷积流来对图进行更新和聚合，输出作为进一步提取的特征。然后，将提取的特征输入定位模块，其中锚点定义的子图由固定大小的特征表示。最后，定位模块对视频帧中的目标物体进行定位和检测，根据定位结果结合关键窗口的建立

方法输出关键窗口。

3.2 多摄像头关联性的构建

本文采用 DiDi-MTMC^[21]算法为关键窗口中检测到的目标物体分配 ID 号。该算法将从视频帧中提取的特征进行融合,并结合融合后的特征进行分类。该算法可为多个摄像头拍摄视频帧中的每个目标物体(本文场景中为人)分配一个唯一的 ID 号。通过对这些 ReID 的结果进行筛选,并根据计算策略计算多个摄像头之间的关联程度。

GC-ReID 算法如算法 1 所示。

算法 1 GC-ReID 算法

输入 GCN 和 ReID 的模型参数,同一场景下摄像头的数量

输出 摄像头之间的关联性

- 1) for $t=1: T$
- 2) 各摄像头对采集到的视频进行初步处理
- 3) 将初步处理后的视频片段传输到直连边缘服务器
- 4) 直连边缘服务器利用训练好的 GCN 模型进行特征提取和目标物体检测
- 5) 根据提取的特征得到关键窗口
- 6) 直连边缘服务器将关键窗口包含的视频片段传输到主边缘服务器
- 7) 主边缘服务器利用 ReID 模型对关键窗口中的目标物体进行进一步的检测并分配 ID 号
- 8) $t=t+1$, 进入下个时隙
- 9) end for
- 10) 输出摄像头之间的关联程度 AS 并返回直连边缘服务器
- 11) 直连边缘服务器结合关联程度值进行视频压缩后将处理后的视频传入云服务器

3.3 视频压缩

摄像头采集的视频经过边缘服务器的处理后传输到云服务器进行进一步的处理。在边缘服务器进行处理可以得到每个摄像头的关键窗口和摄像头之间的关联程度。因此,本文可以结合关键窗口和摄像头之间的关联程度对多个摄像头采集的视频进行视频帧筛选和压缩。压缩后的视频能够减少视频数据量,并提升云端视频处理的效率。

在视频压缩过程中,首先根据当前时段各摄像头生成的视频帧的关键窗口对视频画面进行分割,

去除关键窗口以外的视频帧区域,只保留关键窗口区域进行后续分析。然后,如果边缘端或者云端工作人员想要查看特定摄像头采集的视频或者某个摄像头中重要物体出现的画面,主边缘服务器会对选定摄像头与其余摄像头的关联程度值进行排序。根据排序结果,优先传输关联程度较高的摄像头视频帧的关键窗口区域到主边缘服务器或云服务器进行进一步的操作。这样,只需处理关联程度较高的几个摄像头的关键区域,可以大大缩减视频分析的数据量,同时提升分析效率。

4 实验结果及分析

本文通过相关实验,从不同评价指标的角度来对所提出的 MVCA 方法的性能进行分析。此外,本文还将 MVCA 方法与其他方法进行了比较,并对结果进行了讨论。

4.1 实验环境和数据集

仿真实验是在 SALSA 数据集^[22]的录制视频上进行的。该数据集展示了某学术会议海报会场(如图 2 所示)中不同人员的活动和交流情况,该会场配备了 4 个不同方向的摄像头,分别是摄像头 1、2、3 和 4。每个摄像头可以利用直连边缘服务器生成各自的关键窗口,并实时检测和分配人员的 ID 号。所采集的视频编码标准和格式分别为 H.264 视频编码标准和在主要网络广播应用中支持流媒体的 MP4 格式。仿真环境边缘服务器的配置包括一个英特尔酷睿 i5-10400F CPU 和一个拥有 8 GB 内存的 Nvidia GeForce GTX 1080 GPU。实验中视频片段的持续时间设定为 2 s,帧率为 30 Hz。

4.2 实验结果分析

进行多摄像头视频协同分析时,本文需要在保持分析效果较好的前提下,尽量降低处理的代价。为了评价视频分析方法的性能,需要考虑准确性和代价。本文使用准确率、系统时延和视频压缩率作为性能评价指标。为了验证所提出的 MVCA 方法的性能,本节设计了消融实验和不同方法的性能比较。通过对实验结果进行详细的数据和理论分析,进一步验证了所提方法的有效性。

1) 性能评价指标

准确率。本文采用准确率 Accuracy 作为指标之一来对方法的性能进行评价,如式(20)所示。其中, Q_{true} 表示场景中的实际人数, Q_{dec} 表示检测到的人数。然而,由于本文使用的数据集未提供场景中的

实际人数，因此在实验仿真中使用 YOLO V3 方法^[23]的检测结果作为人员检测的正确基线。

$$\text{Accuracy} = \frac{Q_{\text{dec}}}{Q_{\text{true}}} \quad (20)$$

系统时延。系统时延可以反映视频分析方法的实时性，低时延会有更好的用户体验。在本文中，视频以及处理后的数据在摄像头和直连边缘服务器之间、直连边缘服务器和主边缘服务器之间、直连边缘服务器和云服务器之间进行传输会产生传输时延。此外，关键窗口的生成和多摄像头关联程度的计算会产生计算时延。本文的系统时延主要包括传输时延和计算时延，如式(6)所示。

视频压缩率。为了在保持准确性的前提下降低系统时延并提升系统效用，视频分析方法需要减少视频分析量。因此，本文将视频压缩率 R_c 作为本文方法的性能评价指标之一，如式(21)所示。其中， B_{comp} 表示压缩后的视频数据量， B_{init} 表示原始视频数据量，视频数据量的单位为 MB。本文实验重点观察摄像头 1 及其检测到的目标物体，并根据关联程度来决定是否选择处理和分析其他摄像头采集或处理后的视频帧。

$$R_c = \frac{B_{\text{comp}}}{B_{\text{init}}} \quad (21)$$

2) MVCA 方法性能分析

为了验证关键窗口和多摄像头关联性的作用，首先进行消融实验，将本文提出的 MVCA 方法与只使用 ReID 方法（没有关键窗口，用 No-key window 表示）、使用文献[17]中 STS 方法的帧采样模型（替代 MVCA 中的关键窗口模型）以及只构建关键窗口模型而不进行多视频协同处理的方法（没有关联性构建，用 No-association 表示）进行比较。图 5 分别展示了 MVCA、No-key window、STS 帧采样和 No-association 在准确率、系统时延和视频压缩率 3 个方面的对比。

在准确率方面，MVCA 和 No-association 的准确率较高，这是因为它们利用关键窗口模型有效减少了视频中非重要目标物体的干扰。STS 帧采样的准确率次高，该方法虽然减少了非关键帧的数量，但是对关键目标物体所在的区域划分不够准确，容易受到非重要目标物体的影响，且可能会移除包含关键目标的画面。No-key window 的准

准确率相对较低，这是由于该方法没有对视频帧进行筛选或切割，其中的一些非重要目标会对检测准确率造成影响。

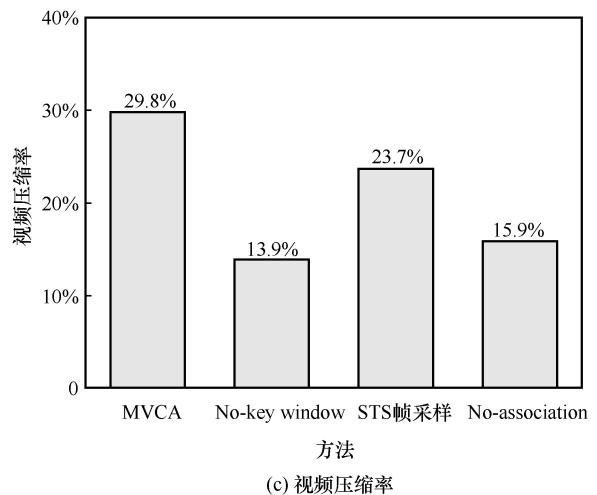
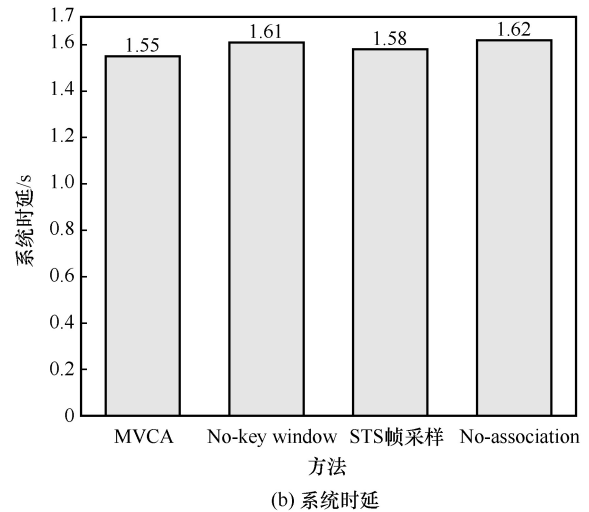
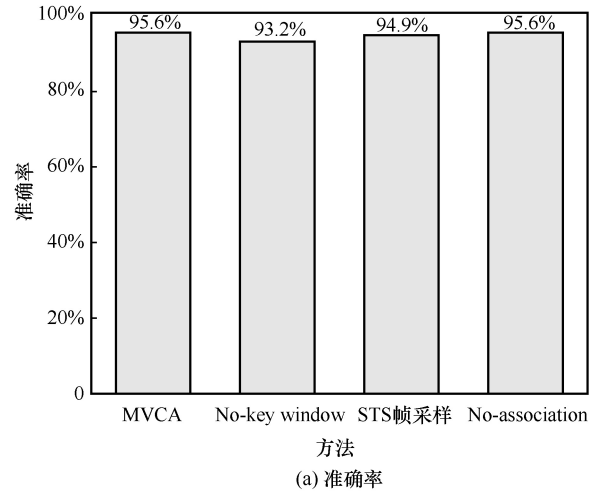


图 5 MVCA、No-key window、STS 帧采样和 No-association 的性能比较

在系统时延和视频压缩率方面, MVCA 方法有着最低的系统时延和最高的视频压缩率, 相比于 No-key window, MVCA 利用关键窗口模型筛选感兴趣区域缩减了视频数据量; 相比于 No-association 方法, MVCA 重点分析关联程度值较高的摄像头视频, 因此进一步减少了需要处理的视频量, 降低了传输和计算时延; STS 帧采样方法虽然通过删减非关键帧的方式减少了所需处理的视频量, 但其处理后的视频量比关键窗口处理后多, 因此可知关键窗口模型比帧采样更有效。

综上所述, 关键窗口对视频中重要目标物体检测准确率的提升和系统时延的降低具有显著作用。多摄像头关联性则可作为进一步减少视频处理量的有效依据。

3) 不同多摄像头视频分析方法的性能比较

本节比较了本文提出的 MVCA 方法与第 1 节相关工作中介绍的一些先进的多摄像头视频分析方法。这些方法包括 REDUCTO、文献[11]方法和文献[14]方法。图 6 分别展示了 MVCA、REDUCTO、文献[11]和文献[14]方法在准确率、系统时延和视频压缩率 3 个方面的对比。

在准确率方面, 文献[11]方法准确率最高, 这是因为该方法针对每个摄像头的视频帧建立了运动模型, 并对所有帧都基于时间进行了详细的分析。MVCA 的准确率次高, 原因是其在利用关键窗口进行视频压缩时, 忽略了部分非关键窗口区域中的重要目标物体, 从而影响了准确率。REDUCTO 和文献[14]方法的准确率都不理想, 这是由于 REDUCTO 采用帧过滤的方法, 有较高的可能性删除包含目标物体的关键帧; 而文献[14]方法在构建目标间的信息模块时候着重考虑目标本身的特征, 未考虑目标在视频帧中位置的变化。

在系统时延和视频压缩率方面, 文献[14]方法具有最低的系统时延和最高的视频压缩率, 这是由于该方法在构建目标间的信息模块时仅关注目标本身特征, 未考虑帧间目标特征, 因此时间复杂度较低; 同时, 在多目标分析时采用的聚类方法仅结合提取的目标特征数据, 未对多个摄像头采集的视频帧本身进行联合处理, 因此数据量较少。MVCA 的系统时延次低而视频压缩率次高, 这是由于利用关键窗口对视频帧进行分割实现视频压缩, 并结合多摄像头关联性模型对传输到云服务器的视频实现进一步压缩。REDUCTO 在系统时延和视频压缩率方面的

表现不如前两者, 原因是其虽然采用了帧过滤方法压缩视频, 但后续对多摄像头处理的压缩力度不足。文献[11]方法具有最高的系统时延和最低的视频压缩率, 这是由于该方法对所有视频帧基于时间建立模型进行详细分析, 未进行帧删减等操作, 同时建立了多人匹配级联模型, 因此耗时较多; 该方法的视频压缩仅通过多人匹配级联模型实现, 因此压缩幅度较小。

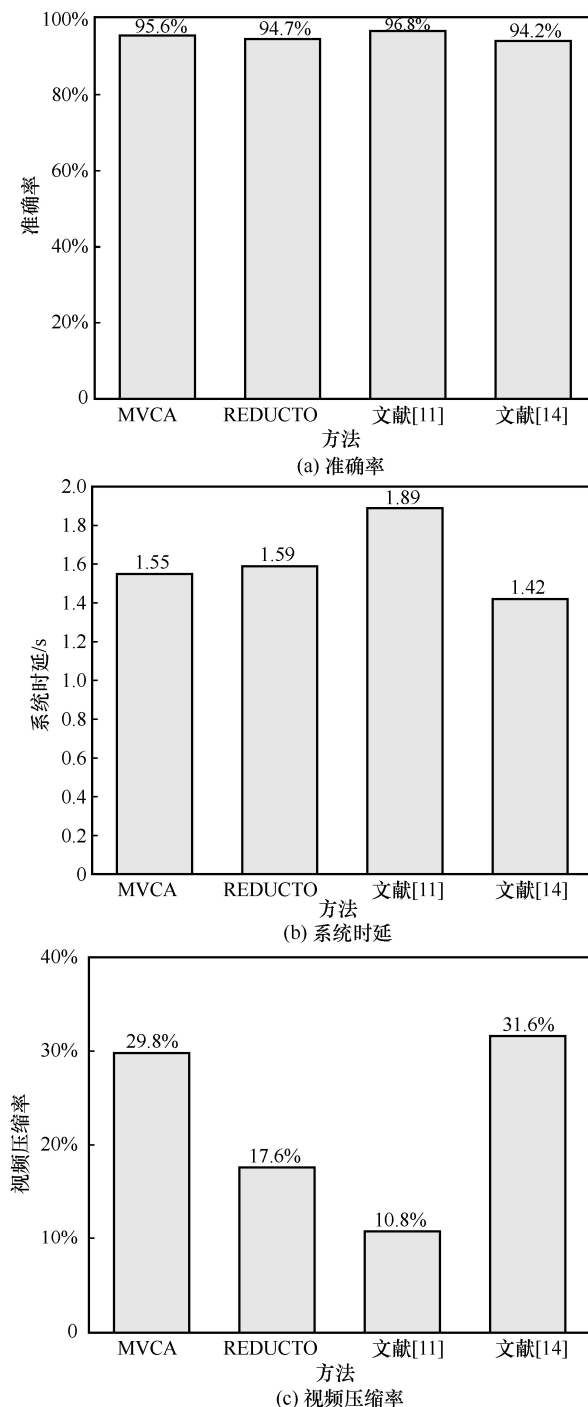


图 6 MVCA、REDUCTO、文献[11]和文献[14]方法的性能比较

综合考虑多个评价指标, 本文提出的 MVCA 方法在准确率、系统时延和视频压缩率 3 个方面整体优于其他方法。

5 结束语

本文研究了智慧城市场景下多摄像头视频分析的问题, 提出了一种高效的解决方法 MVCA。该方法能够实时地对多摄像头采集的视频进行协同分析。具体而言, 通过快速建立视频流的关键窗口来定义感兴趣区域, 并在此基础上快速构建摄像头之间的关联性。通过获取关联程度值, 有效地减少了视频数据的分析量, 从而提高了多摄像头视频分析的效率, 减轻了边缘服务器的负担, 并提升了边缘系统的性能。在 SALSA 数据集上进行的实验表明, 本文提出的 MVCA 方法降低了系统时延, 提高了视频压缩率, 并且保持了 95.6% 的准确率。

在未来的工作中, 笔者将考虑任务卸载和时间空间属性在多摄像头视频分析中的应用, 以进一步平衡边缘服务器的负载, 提升边缘系统的效用。

参考文献:

- [1] JIANG J, ANATHANARAYANAN G, BODIK P, et al. Chameleon: scalable adaptation of video analytics[C]//Proceedings of 2018 ACM SIGCOMM. New York: ACM Press, 2018: 253-266.
- [2] 黄韬, 刘江, 汪硕, 等. 未来网络技术与发展趋势综述[J]. 通信学报, 2021, 42(1): 130-150.
HUANG T, LIU J, WANG S, et al. Survey of the future network technology and trend[J]. Journal on Communications, 2021, 42(1): 130-150.
- [3] YANG K, YI J, LEE K, et al. FlexPatch: fast and accurate object detection for on-device high-resolution live video analytics[C]//Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2022: 1898-1907.
- [4] 赵羽, 杨洁, 刘淼, 等. 面向视频监控基于联邦学习的智能边缘计算技术[J]. 通信学报, 2020, 41(10): 109-115.
ZHAO Y, YANG J, LIU M, et al. Federated learning based intelligent edge computing technique for video surveillance[J]. Journal on Communications, 2020, 41(10): 109-115.
- [5] GAO C S, WANG Y, HAN Y H, et al. IVP: an intelligent video processing architecture for video streaming[J]. IEEE Transactions on Computers, 2023, 72(1): 264-277.
- [6] CHEN Y, ZHANG S, JIN Y B, et al. Multi-server multi-user game at edges for heterogeneous video analytics[C]//Proceedings of 2022 IEEE International Conference on Communications. Piscataway: IEEE Press, 2022: 841-846.
- [7] YANG P, LYU F, WU W, et al. Edge coordinated query configuration for low-latency and accurate video analytics[J]. IEEE Transactions on Industrial Informatics, 2020, 16(7): 4855-4864.
- [8] WANG C, ZHANG S, CHEN Y, et al. Joint configuration adaptation and bandwidth allocation for edge-based real-time video analytics[C]//Proceedings of 2020 IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 257-266.
- [9] RAN X K, CHEN H, ZHU X D, et al. DeepDecision: a mobile deep learning framework for edge video analytics[C]//Proceedings of IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2018: 1421-1429.
- [10] HUNG C C, ANATHANARAYANAN G, BODIK P, et al. VideoEdge: processing camera streams using hierarchical clusters[C]//Proceedings of 2018 IEEE/ACM Symposium on Edge Computing (SEC). Piscataway: IEEE Press, 2018: 115-131.
- [11] GUO Y D, LIU Z Y, LUO H. Multi-person multi-camera tracking for live stream videos based on improved motion model and matching cascade[J]. Neurocomputing, 2022, 492: 561-571.
- [12] CHUNG N M, LE H D A, NGUYEN V A, et al. Multi-camera multi-vehicle tracking with domain generalization and contextual constraints[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2022: 3326-3336.
- [13] LI F, WANG Z, NIE D, et al. Multi-camera vehicle tracking system for AI city challenge 2022[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2022: 3264-3272.
- [14] LIU Y M, ZHANG X C, ZHANG B Z, et al. Multi-camera vehicle tracking based on occlusion-aware and inter-vehicle information[C]//Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2022: 3256-3263.
- [15] LI Y Q, PADMANABHAN A, ZHAO P Z, et al. Reducto: on-camera filtering for resource-efficient real-time video analytics[C]//Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM Press, 2020: 359-376.
- [16] RAY D, KOSAIAI J, RASHMI K V, et al. Vantage: optimizing video upload for time-shifted viewing of social live streams[C]//Proceedings of the ACM Special Interest Group on Data Communication. New York: ACM Press, 2019: 380-393.
- [17] DU L, HUO R. Real-time video transmission optimization based on edge computing in IIoT[C]//Proceedings of 2021 IEEE 29th Interna-

tional Conference on Network Protocols (ICNP). Piscataway: IEEE Press, 2021: 1-6.

- [18] RANADHEERA S, MAGHSUDI S, HOSSAIN E. Minority games with applications to distributed decision making and control in wireless networks[J]. IEEE Wireless Communications, 2017, 24(5): 184-192.
- [19] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2009, 20(1): 61-80.
- [20] WANG Y, SUN Y B, LIU Z W, et al. Dynamic graph CNN for learning on point clouds[J]. ACM Transactions on Graphics, 2018, 38(5): 1-12.
- [21] LI P, ZHANG J B, ZHU Z, et al. State-aware re-identification feature for multi-target multi-camera tracking[C]//Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway: IEEE Press, 2020: 1506-1516.
- [22] ALAMEDA-PINEDA X, STAIANO J, SUBRAMANIAN R, et al. SALSAS: a novel dataset for multimodal group behavior analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(8): 1707-1720.
- [23] REDMON J, FARHADI A. YOLOv3: an incremental improvement[J]. arXiv Preprint, arXiv: 1804.02767, 2018.

[作者简介]



期治博（1987- ），男，河南郑州人，北京邮电大学博士生，中国信息通信研究院工业互联网与物联网研究所工程师，主要研究方向为工业互联网、物联网和区块链领域的技术及产业开发等。

杜磊（1995- ），男，山西晋中人，北京工业大学博士生，主要研究方向为工业互联网、边缘计算、物联网和深度学习算法等。

霍如（1988- ），女，黑龙江哈尔滨人，博士，北京工业大学讲师，主要研究方向为未来网络、工业互联网、边缘计算、网络资源管理、区块链等。

杨帆（1981- ），男，黑龙江哈尔滨人，博士，北京邮电大学工程师，主要研究方向为软件定义网络、高性能路由交换技术等。

黄韬（1980- ），男，重庆人，博士，北京邮电大学教授，主要研究方向为未来网络体系架构、软件定义网络、网络虚拟化等。