arXiv:2305.03222v1 [cs.MM] 5 May 2023

MOSAIC：多个并发视频传感流上的空间复用边缘人工智能优化

Ila Gokarn 新加坡管理大学 新加坡
ingokarn.2019@phdcs.smu.edu.sg

Hemanth Sabella 新加坡管理大学 新加坡
hemanthrs@smu.edu.sg

Yigong Hu 伊利诺伊大学厄巴纳-香槟分校 美国
yi
gongh2@illinois.edu

Tarek Abdelzaher 伊利诺伊大学厄巴纳-香槟分校 美国
zaher@illinois.edu

Archan Misra 新加坡管理大学 新加坡
archanm@smu.edu.sg

摘要

在边缘设备上，对视觉传感器流执行感知任务，同时保持高保真度和高吞吐量，仍然是一项艰巨挑战；尤其是图像尺寸持续增大（例如由 4K 摄像头生成的图像），且深度神经网络模型日益复杂。一个颇具前景的方法是关键性感知处理，即有选择地将计算资源分配给各个图像帧中的“关键”部分。我们提出 *MOSAIC*，一种用于对多个视觉传感流进行此类关键性感知并发处理的新型系统。它能大幅提升可实现的吞吐量，同时几乎不损失感知保真度。*MOSAIC* 从多个视觉传感器接收的图像中确定关键区域，并采用一种新颖的多尺度跨尺度拼接 (MoS) 平铺策略，将这些区域在空间上装箱到单个“画布帧”中；画布帧的尺寸经过设定，使边缘设备能够保持足够高的处理吞吐量。我们使用两个任务的基准数据集——自动车牌识别和基于无人机的行人检测——开展了实验研究。结果表明，在 Jetson TX2 边缘设备上运行的*MOSAIC*能够显著改善吞吐量与保真度之间的权衡。例如，对于基于无人机的行人检测，在批量大小为 4 时，*MOSAIC* 能够将来自 6 个摄像头的输入帧打包，实现 (a) 4.75×（475%）更高的吞吐量（每个摄像头 23 帧每秒，累计 138 帧每秒），而准确率损失为≤ 1%；相比之下，采用的是先到先服务 (FCFS) 处理范式。

CCS 概念

• **计算机系统组织 → 嵌入式与信息物理系统**。

允许免费为个人或课堂用途制作本作品的全部或部分内容的数字或纸质副本，前提是不得为盈利或商业利益而制作或分发副本，且副本须包含本声明及首页的完整引文。必须尊重本作品中由作者以外的其他人拥有版权的部分。注明出处后可以摘录。若要以其他方式复制、重新出版、发布到服务器或分发至列表，则须事先获得具体许可和/或支付费用。请向 permissions@acm.org 申请许可。*MMSys '23，2023* 年*6*月*7*日至*10*日，加拿大不列颠哥伦比亚省温哥华 © 2023 年版权归所有者/作者所有。出版权已授权给 ACM。ACM ISBN 979-8-4007-0148-1/23/06...$15.00
https://doi.org/10.1145/3587819.3590986

关键词

边缘人工智能、机器感知、基于画布的处理

ACM 参考格式：
Ila Gokarn、Hemanth Sabella、Yigong Hu、Tarek Abdelzaher 和 Archan Misra。2023 年。*MOSAIC*：在多个并发视频感知流上进行空间复用边缘人工智能优化。载于第 *14* 届 *ACM* 多媒体系统会议论文集（*MMSys '23*），*2023*年*6*月*7*日至*10*日，加拿大不列颠哥伦比亚省温哥华。ACM，纽约州纽约市，美国，11 页。https://doi.org/10.1145/ 3587819.3590986

1 引言

越来越多的机器感知实时应用（例如，用于自动驾驶的三维场景分析，或通过安装在街道上的摄像头进行车辆跟踪）需要在资源受限的边缘设备上，对多个并发的高分辨率多媒体传感器数据流执行基于深度神经网络的推理。鉴于深度神经网络模型复杂度和数据大小/分辨率（例如 4K 图像帧）均快速增加，此类感知任务的边缘实时执行仍然面临挑战。例如，英伟达 Jetson TX2 [26] 设备在 FP16 精度下、针对 1280×1280 图像帧执行 YOLOv5L6（191 层、4700 万参数）目标检测器时，每秒最多只能处理~2 帧（帧每秒，FPS）。应对这一吞吐量/延迟挑战的传统方法包括：（a）使用更小、准确率较低的深度神经网络模型，并在较低分辨率的数据上运行（例如，复杂度较低的 300×300 SSD 模型[23] 在 TensorRT 优化下可在 TX2 上达到 10–15 FPS 的处理吞吐量）；或（b）使用更昂贵、资源更丰富的边缘设备（例如 Jetson Xavier[26]）。此类方法会带来不利的成本/复杂度与准确率之间的权衡。

为降低这种计算开销，近期研究提出了以下方法：(a) 关键性感知处理方法，仅对单个图像帧中具有较高价值的选定部分进行更高注意力或保真度的处理 [11, 34, 36]，或将其卸载以进行 DNN 任务推理 [37]；或 (b) 选择性计算方法，对某些 DNN 层进行简化 [6] 或跳过 [32]。然而，当前方法未考虑以下场景：多个传感器流具有动态变化的场景特征（例如物体大小和速度），共享边缘设备上的相同计算资源，并且必须并发处理。

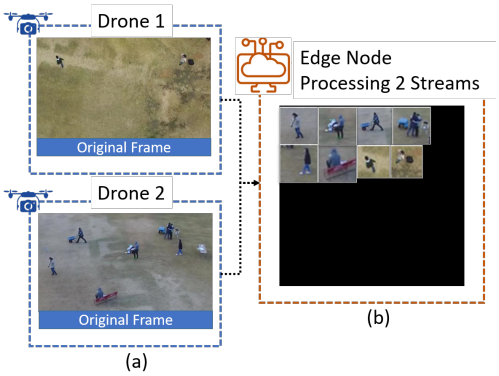

图 1：*MOSAIC* 整体功能：(a) 摄像头拍摄的输入帧；(b) 将来自多张图像的图块打包到单个画布帧上（图像未按比例绘制）。

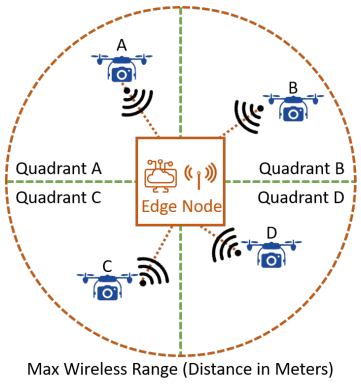

图 2：应用动机：多个独立的自主无人机摄像头流通过无线方式传输到单个边缘节点，并由其处理。

MOSAIC Task on Jetson TX2	Number of Cameras	Batch Size	FPS	CFPS
Pedestrian Detection	6	1	18	108
		4	**23**	**138**
License Plate Recognition	3	1	18	54
		4	23	69

图 3：*MOSAIC* 在 Jetson TX2 上的系统性能；CFPS = 跨 M 个摄像头的累计 FPS

为弥补这一差距，我们提出*MOSAIC*，这是一种由关键性驱动的边缘处理系统，可优化基于边缘的深度神经网络推理任务执行，处理多个图像流。通过这种优化，*MOSAIC* 能够**大幅提升**每路流的吞吐量，使边缘设备在**不牺牲任务准确率的情况下**维持更高吞吐量。*MOSAIC* 可应用于多种场景，包括但不限于：(i) 城市交通监控，其中多个摄像头流监控交叉路口，并由单个边缘节点进行处理；(ii) 城市事件监控，其中多个无人机的摄像头流由单个手持式边缘控制单元处理，如图 2 所示。

MOSAIC 的核心概念是画布，直观地说，它被定义为深度神经网络在配备 GPU 的边缘设备上执行时能够处理的最大输入帧尺寸（例如 C），同时确保处理吞吐量高于最低 FPS 阈值。我们将这些 GPU 能够及时处理的降分辨率帧称为画布帧。因此，并发处理多个摄像头流（例如 M）这一挑战，可以表述为在空间上打包或适配来自 M 个独立图像帧的高优先级区域，使其放入一个尺寸为 C的画布帧中。从概念上讲，*MOSAIC* 以顺序执行每个独立帧的 DNN 执行这一基线模式为基础，将其替换为空间复用范式，在这种范式下，M 帧（每个摄像头传感器各一帧）会被并发处理。

MOSAIC 的设计解决了这一范式面临的两个关键挑战：(a) 以极低的开销从帧中识别并提取关键区域，从而维持较高的吞吐量；(b) 根据目标特征的动态变化，在各关键区域之间公平分配共享画布空间。为了合理填充画布帧，*MOSAIC* 将输入帧高效地分解为多个图块（子区域），这些图块定义在不同的尺度或“缩放级别”上，共同表示不同尺寸的目标/感兴趣区域（RoI）。随后，将所选的、能够忠实表征目标或感兴趣区域（RoI）的图块通过二维逆向装箱 [10] 放置到画布帧上，从而如图 1(b) 所示，$M - fold$ 提升处理吞吐量。*MO- SAIC* 的装箱机制经过精心设计，以确保 (a) 图块能够根据其关键程度按比例调整大小，同时确保底层对象的尺寸符合应用程序定义的空间边界；并且（b）间歇性调用的平铺过程具有极低的开销。

通过两个不同应用的基准数据集——*Okutama-Action* [3] 用于基于无人机的行人检测，*UFPR-ALPR* [18]用于车牌识别（车牌识别），我们展示了*MOSAIC*能够在各种摄像头设置下，显著改善多种机器感知任务的吞吐量与准确率之间的权衡。**主要贡献：**我们的主要贡献如下：

- 关键性保留画布处理：我们开发了一种名为跨尺度拼接（MoS）的三阶段创新流水线，可将来自多个摄像头输入图像、数量可变的关键区域动态适配到单个画布帧中：(i) 一种多尺度平铺机制，利用最小成本最小集合覆盖算法 [2]，针对任意给定的摄像头输入图像，从所有可能的图块中选择一个合适的最小子集，既能捕获所有可能出现的感兴趣对象，又能确保这些对象在最终画布帧中占据尽可能大的区域；(ii) 一种极小极大优化技术，根据各图块计算出的关键性值，对选中的图块分别进行缩放；(iii) 一种差分进化算法 [24]，结合几何约束，将所有选中的图块（来自 M 张不同图像）通过二维逆装箱 [10] 放入画布帧中。对于行人检测，与吞吐量较低的图像顺序先到先服务（FCFS）处理相比，MoS 的准确率损失可以忽略不计（≤ 1%）；与将所有图像 $M = 6$ 均匀下采样并打包到单个画布帧中的策略相比，其准确率还提升了 8%。对于车牌识别，*MOSAIC* 能够将来自 $M = 3$ 张图像中包含车辆的图块差异化打包，使光学字符识别（OCR）的字符错误率（CER）达到 <33 %，远优于基准均匀缩放方法；后者即使逐帧处理（$M = 1$），即以与*MOSAIC*相同的深度神经网络推理图像尺寸（或画布帧尺寸）对每个输入帧进行先到先服务处理，也会导致 CER 达到 100%（完全无法辨读）。

• 实时MOSAIC 实现与性能提升：我们使用 NVIDIA Jetson TX2 作为边缘设备实现了*MOSAIC* 。开发的*MOSAIC* 系统结合间歇式全帧目标定位与连续运动跟踪，以低开销近似提取关键区域。表 3 列出了用于行人检测的 M =6 个摄像头图像，以及用于车牌识别的 M = 3 张图像的打包吞吐量。对于行人检测和车牌识别应用，与依次处理每个输入帧的 FCFS 基准方法相比，*MOSAIC* 的处理吞吐量分别提升了 5 倍（批量大小为=1，每个摄像头 18 FPS）和 4.75 倍（批量大小为=4，每个摄像头 23 FPS ）。

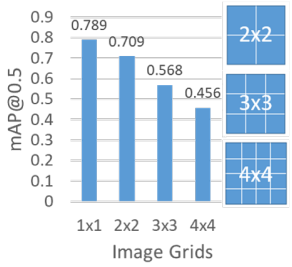

图4：准确率与网格大小的关系（均匀排列）

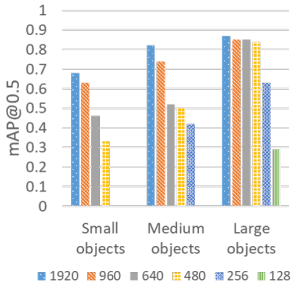

图5：目标检测准确率与{目标大小、分辨率}的关系

2 激发*MOSAIC*的动机

MOSAIC 的主要目标是将 M 个离散摄像头流中的高优先级区域或关键区域排列到单个画布帧上，从而提高每个流的帧处理吞吐量，同时不牺牲推理任务准确率。为此，我们首先探讨目标应用以及支撑*MOSAIC* 的关键设计决策的基本原理。

2.1 目标应用

我们设想有各种各样的应用能够受益于基于画布的处理所带来的吞吐量提升——例如监控、计数或检测场景，其中摄像头传感器和边缘节点都可能静止或移动。考虑一种城市交通监控应用：部署在路侧单元上的边缘节点处理来自 M 个不同交叉路口的交通信号灯摄像头视频流，每个路口在一天中的不同时段观测到不同的交通流量。或者，考虑部署在移动机器人或手持控制单元上的边缘节点，处理自主无人机的摄像头视频流，用于搜救、野生动物偷猎监测或城市人群管控等应用中的空中监控。对于所有这些场景，我们都可以通过将计算资源非均匀地分配，且有选择地仅分配给捕获图像帧中的相关部分，在不牺牲感知保真度的前提下降低基础设施成本并缩短响应时间。我们假设每个摄像头传感器监控一个独立且不重叠的物理区域，不过*MOSAIC* 很可能还可以进一步优化，以利用空间重叠（第 6 节）。

2.2 第一性原理

2.2.1 通过空间装箱多张图像提升吞吐量。我们首先研究将多个输入图像帧均匀地进行空间装箱（不考虑任何重要性），形成图像网格，以此提升处理吞吐量的影响。图 4 绘出了目标检测准确率（以*mAP* 即平均精度均值计算）随装箱图像帧数量变化的情况；这些图像帧来自 Okutama-Action[3]，原始分辨率为 3840×2160。在本文的所有研究中，画布大小设为 640×640，该设置采用了针对本文所选 YOLOv5 [30], 模型的已发表基准测试数据。如图 4 所示，网格中的元素（不同帧）越多，分配给每个输入帧的区域（像素）就越小，目标检测准确率也越低。直观而言，由于统一缩小尺寸，较小的目标会变得越来越小、越来越难以辨认，最终无法检测。因此，系统可以通过为原始帧中的高优先级区域分配更大的画布空间，来提高推理准确率。

2.2.2 影响目标检测置信度的因素。 为进一步探究为何将图像装入尺寸更小的网格（即减少其总像素数）会导致目标检测准确率降低，我们还分析了不同目标尺寸下目标检测准确率的变化。实验观察发现，目标检测准确率会因目标尺寸缩小（将图像帧缩小以适配更小网格所带来的自然结果）或目标分辨率损失（像素化加剧）而下降。如图 5 所示，随着图像尺寸增大，目标可检测性提高，平均目标检测准确率也随之上升。总体而言，图像缩小的影响是不均衡的：小型和中型目标从分辨率和尺寸提升中获益最大，而大型目标的可检测性则能较好地抵御分辨率损失。

2.3 *MOSAIC* 的设计选择

在边缘端，*MOSAIC* 摒弃了通过统一缩小尺寸，将多个图像帧装入单个画布帧的直接做法。相反，*MOSAIC* 采用差异化缩小策略，旨在 DNN 推理前，缩小画布帧中可能存在目标的像素区域之间的差异（这些目标尺寸不一，分布于多个输入图像中）。为了减小这种差异（从而提高小型目标的准确率，同时避免对大型目标造成不成比例的惩罚），*MOSAIC* 首先使用不同尺寸的瓦片，以最佳方式捕捉可能包含目标的像素区域，然后在可接受的范围内对这些瓦片进行空间缩放，使其适配目标画布帧。

3 *MOSAIC*

现在，我们描述如图6所示，*MOSAIC* 在边缘侧执行的关键性感知自适应处理。*MO-SAIC* 在两种运行模式之间交替切换：(1) 跨尺度拼接（**MoS**）和 (2) 周期性稳定（**PS**）。这两种模式均与*MOSAIC* 的 内存功能协同工作，以支持*MOSAIC*的目标：(i) 从输入帧中提取关键区域

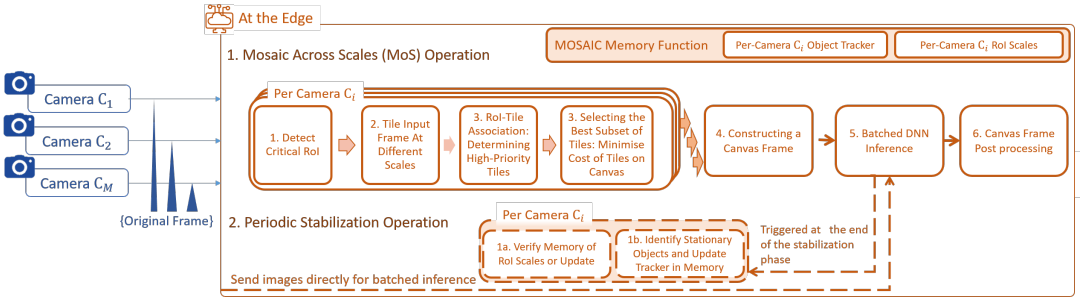

图6：*MOSAIC* 边缘运行的子组件框图

；以及 (ii) 将这些关键区域装箱到画布帧中，以供深度神经网络推理。周期性稳定（PS）操作通过对所有摄像头流执行全帧批量深度神经网络推理，检测特定类别的关键目标，从而初始化并间歇性刷新*MOSAIC* 流水线。对于每台摄像头，PS 会定位新检测到的目标、识别静止物体，并更新*MOSAIC* 的 内存功能中维护的目标跟踪器，以修正上一轮 MoS 操作累积的所有跟踪器不确定性。PS 还会促使*MOSAIC* 检查每台摄像头近期观测到的感兴趣区域和目标尺寸分布，以计算一组摄像头专属的感兴趣区域尺度。这些感兴趣区域尺度会更新至内存功能中，并用于指导下一轮 MoS 操作，确定各摄像头流中预期感兴趣区域的尺寸。对于每路摄像头流，MoS 操作首先通过基于运动的背景减除估算关键感兴趣区域的位置，并更新摄像头专属的目标跟踪器。MoS 将关键性理念进一步延伸：在每个已识别的感兴趣区域尺度下对输入图像进行分块，找出包含跟踪器所跟踪全部感兴趣区域的图块，并尽可能将这些图块（进而也将图块内的感兴趣区域）放大，以提高目标检测准确率。这种方法有效地将边缘侧有限的计算资源“空间分配”给关键区域。当然，总体而言，这种缩放是一种零和博弈，因此应优先放大较小、较远的感兴趣区域；但实际情况使这一目标更为复杂：输入帧中的单个图块可能包含不同尺寸的感兴趣区域（例如，既有较近的目标，也有较远的目标）。

从较高层面来看，MoS 流程必须在两个相互冲突的目标之间取得平衡：(a) RoI 理想情况下应占瓦片足够大的比例，这样调整瓦片大小最终才不会导致目标大幅缩小——这一标准倾向于较小的瓦片；(b) 理想情况下，应尽可能减少需要适配到画布中的瓦片总数，从而让每个瓦片在画布中占有更大的空间——这一标准倾向于较大的瓦片尺寸。MoS 通过接下来介绍的一系列创新性顺序步骤来创建画布帧，并综合平衡这两个目标：

- 首先，在每个图像帧中提取一组最小瓦片（位于内存功能中维护的各个摄像头特定感兴趣区域尺度上），以覆盖跟踪器维护的可能感兴趣区域。
- 然后，通过二维装箱的“逆向”方式打包此类多尺度图块（即确保所有相关图块均被打包），构建由这些图块组成的合成图像画布帧；随后将该画布帧发送至深度神经网络模型进行推理。

3.1 周期性稳定（PS）

PS 操作会利用针对所有传入摄像头流执行的批量全帧深度神经网络推理结果，初始化并刷新整个 *MOSAIC* 流水线，其中包含特定类别的目标、目标位置以及目标尺寸分布；此过程覆盖整个 PS 持续时间。PS 操作旨在应对 MoS 操作面临的两项重大挑战。首先，由于摄像头的物理安装位置以及观察者与目标之间的距离不同，摄像头观察到的目标尺寸分布也各不相同。例如，安装在路灯杆上的摄像头可能会观察到由大型前景目标和较小背景感兴趣对象组成的混合分布，而安装在无人机上的摄像头则可能观察到尺寸均匀较小的目标分布。目标尺寸分布也可能随时间动态变化，例如，无人机摄像头在飞行高度升高或降低时，可能会观察到不同尺寸的目标。为尽可能捕获不同尺寸的关键感兴趣区域，MoS 会以不同尺度对输入帧进行分块，并评估哪些图块包含输入帧中的关键感兴趣区域。MoS 依靠 PS 操作期间评估的目标尺寸分布来确定分块步骤所需的尺度数量，以及每个尺度采用的分块尺寸。其次，MoS 流水线通过执行背景减除来估计关键感兴趣区域，从而捕获目标可能处于运动状态的感兴趣区域，但可能会漏掉静止、停止或被遮挡的感兴趣对象。类似地，MoS 会利用 PS 操作获得的检测目标位置，更新特定摄像头的目标跟踪器，并纳入那些可能被基于背景减除的输入帧关键感兴趣区域估计所遗漏的目标位置。PS 的周期性和持续时间是可配置参数，用于描述观察到的目标尺寸分布预期平均变化速率。然而，PS 周期性参数与 *MOSAIC* 可实现的整体吞吐量呈反比：较短的 PS 周期和较长的 MoS 周期可带来更大的吞吐量提升，因为 PS 周期实际上是在全分辨率下处理帧（不具备任何空间复用增益）。在所有摄像头流完成 PS 持续时间内的全帧检测后，会触发 *MOSAIC* 刷新每个摄像头的感兴趣区域尺度和目标跟踪器，具体如下文所述。

(1a) 计算每个摄像头的感兴趣区域尺度：为了确定摄像头特定的感兴趣区域尺度及其尺寸，该函数首先收集在 PS操作期间观察到的物体尺寸分布，并评估检测到的对象之间是否存在重叠或彼此距离较近的情况。在 MoS 操作期间，此类目标可能会被检测为单个感兴趣区域，因此该函数会将此类重叠/邻近目标的最小外接矩形添加到

观察到的物体分布中。随后，使用 KNN聚类模型对所得的物体尺寸分布进行聚类，并通过肘部检测方法确定适当的 k 值（不同聚类的数量）。该 k 值随后决定 MoS 将为特定摄像头采用的尺度数量，而每个 k聚类的质心则决定相应瓦片的尺寸。对于每个尺度，MoS 取每个质心的 x 和 y 坐标中的较大值，即 $max(x_{centroid}, y_{centroid})$，作为该尺度的瓦片划分尺寸，并将其四舍五入到最接近的 32 的倍数，以提高计算效率，从而更好地表示尺寸落在该聚类范围内的目标。

图7通过绘制 Okutama-Action 数据集中的摄像头1在所有帧中观测到的目标分布，展示了这一过程；聚类识别出3个簇，其质心分别为 (36，39)、(50，54) 和 (81，44)。因此，MoS 确定两种瓦片尺度，分别为 64×64 和 96×96。MoS 还引入一个兜底瓦片 $\sim 1.5x$，其尺寸大于所确定的最大瓦片（在我们的示例中，这会额外产生一个 128×128 瓦片），以适应后续观测到的目标/感兴趣区域可能大于此前PS操作期间见到的任何目标/感兴趣区域的情况。

(1b) 识别静止物体并在内存功能中更新逐摄像头跟踪器：此功能初始化并刷新逐摄像头卡尔曼滤波器基于质心的跟踪器，用于维护摄像头视场中所有目标最近观测到的位置和状态。此状态指目标是“活动”（即处于运动中）还是“静止”，通过观测每个质心在整个 PS 持续时间内移动的距离来计算。

3.2 跨尺度 Mosaic（MoS）

PS操作完成后，MoS 会立即开始处理接收的摄像头帧。如图6所示，MoS 操作中的步骤1至4会在每路摄像头流上并行执行，直到确定所有 摄像头流中输入帧所包含关键 RoI 的相关图块。随后，这些相关图块会被装箱到一个画布帧中，以进行深度神经网络推理和后处理。

(1) 确定输入图像的关键区域：该 MoS 子组件会汇总一组掩码边界框，估计输入帧中可能存在关键感兴趣区域。为此，将接收到的输入帧与前一帧进行比较，以执行背景减除，从而检测可能包含运动目标的关键感兴趣区域。MoS 会根据检测到的感兴趣区域位置更新掩码边界框和摄像头物体跟踪器，以便与已知的感兴趣区域轨迹进行匹配，并将更新后轨迹的状态设为“active”。这种方法能够稳健地处理进入摄像头视场（FoV）的新目标，以及视场内移动的现有目标。对于未根据当前输入帧中的感兴趣区域进行更新的轨迹，MoS 会评估每条轨迹的状态。对于标记为“stationary”、且未被基于背景减除的运动估计检测到的轨迹，MoS 会从跟踪器内存中提取其最后已知位置，并将其加入掩码边界框。对于其余未匹配轨迹，MoS 假定先前处于运动状态的目标可能已经停下、与另一目标交错而导致轨迹切换，或被另一目标遮挡并因此与另一感兴趣区域合并。在所有这些情况下，MoS 都会将该轨迹的状态设为“last-seen”，并将所有这些

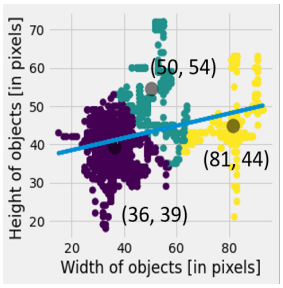

图7：Okutama-Action 无人机序列 1.1.8 中的目标尺寸分布与聚类

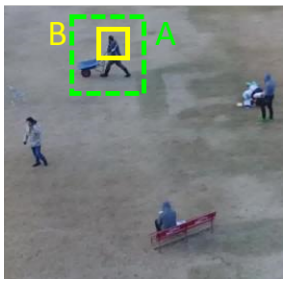

图8：不同尺度的图块，捕捉具有不同“优良度”的目标

未匹配轨迹的位置加入当前输入帧的掩码边界框列表中。此类“last-seen”目标会保留在内存中，直到重新激活或 MoS 操作周期结束。这样做是为了避免遗漏在 MoS 操作剩余期间停留在最后已知位置的目标。最后，如果检测到摄像头自身运动（例如，摄像头安装在移动无人机上），该子组件还会执行摄像头运动补偿：检测帧关键点，匹配当前输入帧与前一输入帧的描述符，并通过将摄像头运动建模为部分二维仿射变换，计算当前输入帧中所有已知轨迹的新位置。

（2）对输入图像进行切片：随后，MoS 会在*MOSAIC* 内存功能所维护的每个尺度下生成一组图块，并使用可配置的重叠参数来确定图块步长。

（3）确定高优先级图块：在 k 不同尺度下生成的一组图块中，可能不包含任何目标/感兴趣区域，也可能只包含目标的部分视图，或完整包含目标。接下来，MoS 会确定其中能够充分捕捉关键区域的图块子集，同时兼顾前文提到的两个相互冲突的目标。MoS 根据这组图块中的所有已生成图块构建空间四叉树，然后使用步骤1中汇总的掩码边界框执行相交边界框搜索。接着，对所有与各个掩码相交的图块，评估掩码在单个图块中的“拟合优度”。图8展示了这一选择过程：MoS 选择图块（A）而非图块（B），因为图块 B 仅捕捉到部分人体目标。

更正式地说，MoS 在瓦片满足两个不同的“良好性”标准时，会为其分配一个掩码。首先，瓦片必须在宽度和高度上都包含掩码的相当大一部分（≥ 95%）；这可确保目标清晰可见（且不会被过度裁切），从而能被下游目标检测器深度神经网络成功检测到。其次，掩码与瓦片的高度比必须处于 (0.5,0.9) 范围内——也就是说，目标所占的瓦片面积比例不能太小或太大。下限提高了目标在缩放并进行二维装箱放置到画布上后仍能保持较大尺寸的概率；尺寸小于瓦片尺寸一半的掩码实际上会被分配到较小尺度的瓦片。相反，上限（0.9）可确保优先将尺寸较大的目标分配到较大尺度的瓦片（这些瓦片很可能还能容纳更多目标），同时也能尽量减少

由于掩码不准确而导致过度裁剪的风险。通过这一过程，MoS 筛选出一个图块集合，其中每个图块都以适当的比例忠实呈现估计的目标，从而确保在构成画布帧时，这些目标仍能保持合理大小（成功进行深度神经网络检测的可能性更高）。

(4) 选择最佳图块子集：许多不同的图块组合都可能“覆盖”（从集合论的意义上说）所有需要纳入画布以进行推理的掩码/目标。然而，为了在画布帧上实现高效（冗余更少）的装箱，必须只选择那些不仅可能保留画布中目标尺寸、还能够最大限度减少“浪费像素”的图块（直观而言，即代表背景或其他无关目标的像素总数，以及在多个图块中重复捕获的目标）。这一双重优化过程可视为箱中箱装箱问题：MoS 不仅必须确保所选图块“覆盖”所有目标，还必须使所选图块产生尽可能少的“浪费像素”。由于该问题本质上是 NP难，我们通过 Min-Cost Min-Set Cover（*MCMSC*）算法的贪心近似方法进行选择，该算法总结于算法1中。

算法1：贪心最小成本最小集合覆盖算法

结果：所选图块的子集 S

$Universe = m_1,\ m_2,\ ...m_m$ 包含 M 个掩码的集合；

$Tiles = t_1,\ t_2,\ ...t_n$ 可能包含一个或多个掩码的 N 个图块集合

已分配的掩码；

$Costs = c_1,\ c_2,\ ...c_n$ N 个图块的成本集合；

$S \leftarrow \emptyset$;

当 $S! = Universe$ **时执行**

  如果 $|t_i - S| \geq 0$ & $(c_i/|t_i - S|) > 0$ **然后**

    $Subset \leftarrow \min(c_i/|t_i - S|)$;

    最小化包含相同掩码的图块数量

    并尽量降低与画布相关的额外成本。

    通过添加额外的瓦片

    $S \leftarrow S \cup Subset$;

  end

结束

直观而言，MCMSC 算法选择一组图块，使这些图块共同最大限度地降低图块中出现的无用背景/非目标像素的成本，同时确保每个感兴趣的目标都至少属于一个满足前述优良性标准的图块。为实现这一目标，MoS 对筛选后的图块集合中的掩码与图块的关系构建了两种不同的视图。在第一种视图中，MoS 针对每个掩码构建一组可接受的图块，以尽可能完整地捕获该掩码。为了计算将图块纳入画布的成本，第二种视图则同时汇总分配给每个独立图块的所有掩码。基于第一种视图的最小集合覆盖（Min Set Cover）确保所有掩码（可能与目标相关的像素）都被纳入画布，而基于第二种视图的最小成本（Min Cost）则选择画布必须容纳的最小图块子集。

这种独特的表述方式具有多项优势。首先，在实际可行的情况下能够映射到多个瓦片的目标，在最终画布中通常只会出现在一个（至多两个）瓦片中，从而减少在基于 DNN 后非极大值抑制（NMS）的推理步骤中出现假阳性的可能性。其次，空间邻近性较高的多个目标通常由一个较大尺度的瓦片表示（从而减少浪费的像素），而不是由多个较小尺度的独立瓦片表示。具体而言，优化中的最小集合覆盖步骤（第 7 行）会选择包含最多尚未出现在画布中的掩码的瓦片，而最小成本步骤（同样在第 7 行）会同时考虑单个瓦片添加到画布中的掩码数量（即基数或唯一掩码数），以及将该瓦片纳入画布的相关成本。该步骤结束时，MoS 会为每个摄像头传感器帧计算最终的最优瓦片子集。对于所选子集中的每个瓦片，MoS 还会计算空间尺寸界限和 弹性因子 （基于目标/瓦片的尺度与应用需求的组合），以确定画布帧构建期间可容忍的缩放范围和幅度。我们通过实证观察发现，该尺寸界限具有尺度依赖性：不同尺度的瓦片可容忍的缩放范围各不相同，超出这一范围后，目标会变得小到无法接受，或者在放大时出现过度像素化，从而严重影响 DNN 推理准确率。实际上，小目标对空间缩放幅度的剧烈变化尤其敏感。

(5) 构建画布帧：在上一步之后，MoS 已有效地筛选出一组瓦片、它们的空间尺寸界限和弹性因子，记为 ST_i，每个输入图像帧 (F_i) 的尺寸各不相同。现在，要将来自多个图像传感器的传感器数据打包到具有给定尺寸的画布帧上，MoS 需要确定所有这些M子集中所有瓦片的调整后尺寸——即，$\forall$ 瓦片 t：使得 $t \in \bigcup_{k=1}^{M} ST_k$ 能够打包到一个尺寸已定义的画布帧上。这可以概括为一个逆装箱问题 [10]：给定一组已定义的物品和箱子，算法必须找到物品尺寸向量的最小扰动，使所有物品都能装入规定数量的箱子。

MoS 通过采用一种计算上的方法来近似此类优化 - 快速 差分进化算法 [24]（一种元启发式优化方法）采用极大极小优化目标函数，在其定义的边界范围内尽可能缩小可获得的最大尺寸，同时确保所有包含的瓦片的总面积小于画布帧面积。优化器还将弹性因子作为图块权重，并接受用于限制每个单独瓦片尺寸的空间尺寸界限。差分进化算法还采用等式惩罚函数，用于监测已打包瓦片的数量是否少于 MoS 选定进行打包的数量。如果是，则要求下一代瓦片单调地缩小（或“挤压”）每个瓦片的尺寸（依据弹性因子），以便将所有瓦片打包到画布帧上，同时确保“挤压”幅度不违反瓦片定义的空间下界。本质上，求解器会在各瓦片可接受的边界内尽可能增大其尺寸，同时尝试将所有瓦片以二维装箱方式打包到画布帧上。如果所有瓦片都被“挤压”到各自允许的最小尺寸，MoS 会放宽下界，并告知用户可能会损失准确率，同时建议减少分配给边缘节点的摄像头流数量（这是一种“准入控制”）。求解器收敛后，MoS 不仅会获得每个瓦片调整后的尺寸，还会获得其最优打包所需的画布位置。

(6) 后处理：如前所述，深度神经网络随后会在生成的画布帧上执行推理任务，该画布帧由重新打包并重新定位后的瓦片组成。最后，MoS 还会维护画布中每个输入图像对应的瓦片→箱空间映射，并利用该映射对所有检测到的对象（及其位置）执行推理后坐标转换，使其回到原始输入帧中。转换完成后，MoS 还会针对每个原始输入帧中的转换后边界框执行通用的非极大值抑制（NMS），以消除可能在画布上出现多次（位于不同尺度的瓦片中）的目标所造成的重复计数。之后，再对后处理后的目标应用其他下游视觉处理流水线。

4 系统与评估设计

我们在边缘设备上部署 PS 和 MoS 流水线，如图6所示。在 PS 操作期间，来自 M 个摄像头的输入帧会被批量处理，并直接发送以进行深度神经网络推理。默认情况下，*MO-SAIC* 每隔 30 秒对 10 帧执行一次 PS 操作。此设置（更深入的分析请参见第5.2.2节）能够充分平衡两方面的需求：一方面，需要足够数量的帧来检测和分类静止目标与移动对象；另一方面，则希望尽可能减少对 *MOSAIC* 的整体可实现吞吐量的影响。在 MoS 操作期间，来自 M 个摄像头的输入帧会被接收，并由 MoS 流水线并行评估，以构建画布帧，如图9所示。MoS 接收 i^{th} 个输入帧、$f_m^i \forall m \in M$ 个摄像头，并从所有 M 帧中选定的瓦片子集中构建一个画布帧 C_i。深度神经网络推理任务的批量大小配置为 b，这样便可以在对上一批次进行深度神经网络推理的同时，将接下来的 b帧（$f_i,\ldots,f_{i+b}$）通过流水线式画布构建处理到画布帧（$C_I,\ldots,C_{i+b}$）中。因此，*MOSAIC* 可实现的吞吐量取决于 PS 和 MoS 两种运行模式。我们通过实验确定，在 Jetson TX2 上，经过 TensorRT 优化的 YOLOv5s 模型能够在尺寸为 640×640的画布帧上，以=4的批量大小实现合理的准确率和 ~ 170毫秒的推理延迟（从而使每个摄像头达到 ≈ 6帧每秒，累计达到 24 帧每秒）。*MOSAIC* 采用这一配置，并针对所选应用及 MoS 操作所使用的一组摄像头流，确定 M 的最大值，即单个边缘设备能够支持的最大摄像头数量。 M 的取值受到两个关键因素的限制：(i) 从 M 个摄像头输入帧构建 b 个画布帧所需的画布构建时间不得超过深度神经网络推理时间（$\sim\leq 170ms$），以确保流水线执行能够无缝进行；(ii) 将来自 M 个摄像头的所有选定瓦片打包到画布帧中时，不得违反其依赖于应用和尺度的空间尺寸界限。*MOSAIC*遵循这两项约束，为应用选择最合适的 M 值，以确保在 MoS 阶段，*MOSAIC* 始终能够为所有 M 个摄像头输入流实现 24 帧每秒的处理速度。然而，由于周期性执行的 PS 操作需要按顺序处理所有传入的摄像头帧，且不对其进行任何修改，因此*MOSAIC* 的整体可实现吞吐量会有所降低。在批量大小为 $b = 4$的情况下，在这些默认设置下，PS 操作会为跨越 M 个摄像头输入帧的 10 个稳定性检测帧处理增加 $10 \times M/24$ 秒的延迟。例如，对于 $M = 6$ 个摄像头流、批量大小为$b = 4$的情况，PS 操作会使整体处理延迟增加 ~2.5。

秒，使 PS 和 MoS 阶段的总体可实现 *MOSAIC* 吞吐量达到 23 帧每秒。

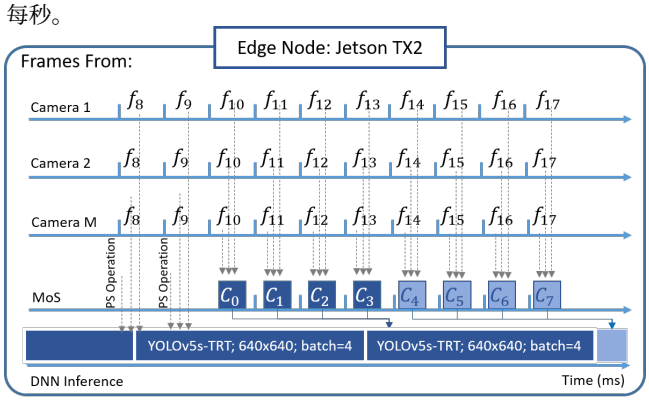

图9：Jetson TX2边缘节点上*MOSAIC*的 MoS流水线概念设计

评估平台：我们在英伟达 Jetson TX2 [25]，上对*MOSAIC* 进行了评估。Jetson TX2 [25]，是一款具有代表性的边缘设备，配备了一个包含 256 个 CUDA 核心的 PASCAL GPU，以及一套 ARMv8 多处理器架构，该架构同时支持双核 NVIDIA Denver 2 CPU 和四核 ARM Cortex A57 MPCore CPU。

基准数据集： 我们使用两个针对不同应用的基准数据集评估 *MOSAIC*： *Okutama-Action* [3]用于基于无人机的行人检测，*UFPR-ALPR* [18] 用于车牌识别。Okutama-Action 数据集包含 43 段无人机视频序列，分辨率为 4K（3840×2160），采用 30FPS 编码，分别产生 54664 张训练图像和 14210 张测试图像。UFPR-ALPR 数据集同样包含 90 段视频序列，分辨率为 1920×1080，采用 30FPS 编码，分别产生 3600 帧训练数据和 1800 帧测试数据。我们将每段视频序列视为一个独立的摄像头，并使用M 个摄像头的不同组合，同时避免摄像头输入流重复，以确保比较的公平性。*MOSAIC* 为 Okutama-Action 数据集构建包含“person”目标的画布帧，为 UFPR-ALPR 数据集构建包含 {car,motorcycle, bus} 类别车辆目标的画布帧；对于 UFPR-ALPR，下游 OCR 流水线随后会根据检测到的车牌边界框执行车牌识别。

评估模型：对于画布帧上的目标检测，我们采用经过 TensorRT 优化的 YOLOv5s 模型，这是一种兼容边缘设备的预训练模型，具有 720 万个参数和 16.5 GFLOPs。该模型使用 MS COCO数据集 [20] 进行预训练，并在所选数据集上进行微调，以提高对被遮挡、未见过以及低分辨率小尺寸目标的敏感性。

评估指标： 为了评估行人检测应用中感知准确率可能取得的提升，我们报告模型在 IoU 阈值为 0.5 时的平均精度均值，即 mAP@0.5，并报告推理延迟，即 FPS_C和累计帧率 CFPS $= M \times FPS_C \times b$；其中， FPS_C表示画布帧尺寸为 C、批量大小为 b且每个画布帧包含 M 个摄像头时达到的吞吐量。对于车牌检测应用，我们通过字符错误率（CER）指标评估下游光学字符识别（OCR）的质量。CER 使用莱文斯坦距离来计算将预测字符字符串转换为

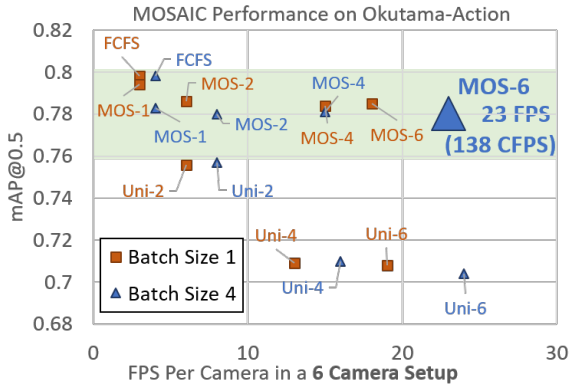

图10：*MOSAIC* 的性能（在 Okutama-Action 上进行行人检测）

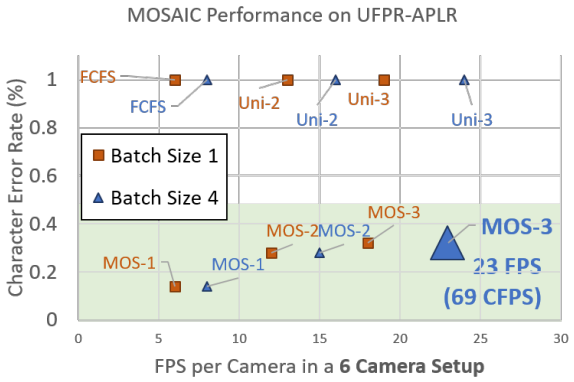

图11：*MOSAIC* 的性能（在 UFPR-ALPR 上进行车牌识别）

真实标注所需的最少单字符更改次数（即插入、删除或替换），并除以真实标注中的字符数取平均值。CER 越低，OCR 性能越好；CER 为 0 表示识别完全正确。

评估基线：我们将*MOSAIC* 的性能与两个基线进行比较：

1. FCFS：将接收到的帧调整为画布帧尺寸，并直接用于批量深度神经网络推理，不进行任何空间修改——实际上，这与*MOSAIC*在 PS 阶段的行为完全相同。**2. Uniform-M：**记作 Uni-M，其中 M 表示打包到单个画布帧中的图像数量。Uniform-M 将画布划分为相同数量的网格行和列，并将每个输入图像分配到网格中的一个单元格。Uniform-M 还会确定在网格、水平或垂直堆叠 M 输入图像的方法中，哪一种能形成最佳网格结构，使每个单元格对应的输入帧相较于其原始尺寸具有尽可能低的缩小比例。

5 评估

我们首先评估基本假设的有效性：采用 MoS 方法对多个输入帧进行关键性感知处理，有助于改善不同物体分布、摄像头设置和应用场景下吞吐量与准确率之间的权衡。我们还将 MoS 所采用的基于画布的处理方式，与近期研究采用的批量推理方法进行了比较。为了更深入地了解 MoS 的优势，我们还针对不同参数设置下*MOSAIC* 性能的变化进行了消融研究。

5.1 *MOSAIC* 系统性能

一般来说，我们预计吞吐量与准确率呈反比：增加 M（打包到单个画布帧中的摄像头帧数量）应使吞吐量提高 M倍，但会导致目标检测平均准确率降低。图 10 和图 11 绘出了不同*MOSAIC*配置和基线方法下的准确率与吞吐量关系，其中 Okutama-Action 的 M取值范围为 {1,. . .,6}，UPFR-ALPR 的取值范围为 {1,. ..,3}；两者使用的画布帧尺寸均为 C 640×640。图中按摄像头绘制吞吐量，计算方法是将整体处理吞吐量（即累计帧率）除以 M。如第 4 节所述，可打包到一个画布帧中的摄像头最大数量取决于画布构建时间、目标尺寸/比例，以及所得图块子集的空间尺寸界限。由于 UFPR 的目标是最终对检测到的车牌目标执行光学字符识别，因此它对每个瓦片设定了更严格的空间尺寸界限，限制了该数据集的摄像头数量为 $M = 3$。这与 Okutama-Action 的行人检测应用不同：后者允许较宽松的空间尺寸界限（尽管其中的人物类别目标更小（~ 64×64）），因此可以成功将 $M = 6$ 个摄像头打包到一个画布帧中。对于批量大小 $b = 1$，*MOSAIC* 从 $M = 3$ 和 $M = 6$ 个摄像头构建画布帧的平均耗时分别为 ~ $9ms$ 和 ~ $13ms$（批量大小为 $b = 4$时分别为~ $36ms$ 和 ~ $52ms$），远低于 170 毫秒的推理时限。因此，M 的两项约束均得到满足，我们据此针对各个应用比较 Uniform-M。

行人检测应用： 在图10中，对于批量大小$b = 1$，FCFS（每张图像均独立、按顺序处理，即 $M = 1$）提供了最高准确率 ~0.79，但在使用6个摄像头的设置中，每个摄像头的吞吐量仅为 ≤3帧每秒，十分有限。另一方面，Uniform-6（其中 $M = 6$张图像被均匀地压缩到画布帧中）为每个摄像头提供了最高吞吐量（~ 19 帧每秒），但准确率显著下降了8%，降至 ~ 0.71。相比之下，随着 M 变化，*MOSAIC* 提供了明显更有利的权衡。当 $M = 6$时：(a) 与 FCFS 相比，MOSAIC-6 的准确率仅损失可忽略的~< 1%，但吞吐量提高了500%以上（5倍）；(b) 与 Uniform-6 相比，MOSAIC-6 的准确率显著提高至 +7.8%，而吞吐量仅小幅下降0.04%（Uniform-6 = 19 帧每秒；MOSAIC-6 = 18 帧每秒）。这一轻微下降是由于 PS操作 每30秒处理10帧所致。对于批量大小 $b = 4$，也可以观察到类似的吞吐量与准确率权衡，但有一个例外：在6个摄像头上，Uniform-6和MOSAIC-6方法分别实现了每个摄像头24帧每秒和23帧每秒（累计吞吐量分别为144和138帧每秒），其中*MOSAIC*的 方法比Uniform-6的准确率高出 +8%。

车牌识别应用： 在图11中，对于批量大小 $b = 1$，Uniform-1 的字符错误率为=100%（光学字符识别完全失败），这表明，将单张图像统一缩小以适配 640×640画布，会导致车牌出现严重且灾难性的像素化

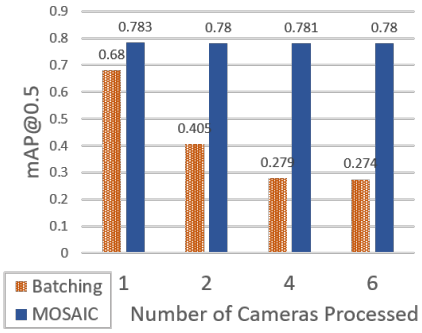

图 12：mAP 与摄像头数量的关系用于瓦片处理的区域边缘端的技术技巧

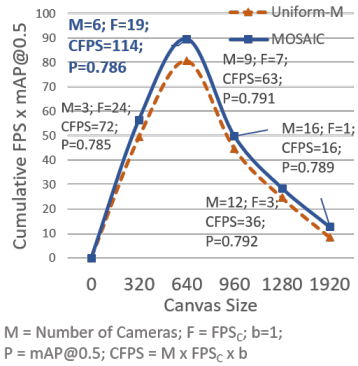

图13：画布大小与累计帧每秒×mAP@0.5

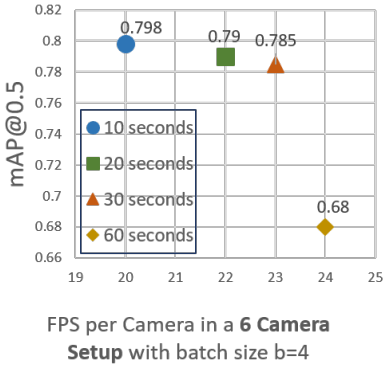

图 14：*MOSAIC* 的 性能增益与 PS 周期的关系

。另一方面，MOS-1 保留了高优先级车辆目标的像素分辨率，从而在下游完全恢复光学字符识别能力，同时字符错误率维持在合理的 15%。*MOSAIC* 进一步提升了下游光学字符识别质量：它还会将可能包含车辆感兴趣区域的图块从 $M = 2$ 和 $M = 3$ 图像中一并打包到画布帧上，从而同样在合理的字符错误率范围内（分别为 29% 和 33%）恢复下游光学字符识别能力；同时，对于由 3 台摄像机构成的系统，其系统吞吐量也大幅提升至每台摄像机 23 帧每秒（累计帧每秒数为=69）。**与批处理单个目标的对比研究：** 我们将*MOSAIC* 的 性能与近期具备优先级感知能力的处理方法 [13, 21, 22] 进行比较。这些方法从摄像机帧中提取高优先级感兴趣区域，并对选定区域进行批量推理。与逐帧推理相比，批量推理一次处理多个输入，以提升并行度并更充分地利用 GPU 的计算能力。然而，与*MOSAIC* 的 基于画布的执行方式不同，后者可以将不同大小的图块打包到同一画布帧中，而批量推理要求所有输入图块图像大小相同。因此，我们比较行人检测应用的平均准确率，以展示*MOSAIC* 的 差异化图块缩放策略在边缘 MoS 流水线中的优势。首先，我们通过对 Jetson TX2 进行系统性能分析，获取 640×640 画布帧以及不同批量大小/图像尺寸组合下批量图块图像的平均执行时间。随后，在线确定适用于基于批处理执行的最佳图块图像尺寸，即能够处理所有输入图块、且执行时间不超过处理一帧画布帧所需时间的最大图块图像尺寸。MoS 中筛选出的图块会通过填充调整为这一输入图像尺寸，以进行批量推理。这样，批量推理在批量大小为 $b = 1$时，便能达到与*MOSAIC* 的 基于画布的推理相同的吞吐量。

图 12 展示了使用 Okutama-Action 数据集，同时处理 B={1, 2, 4, 6} 个输入帧时的 mAP 分数。我们可以看到，当输入帧数量为 1 时，基于画布的方法和基于批处理的执行都能达到较高的准确率。随着 M 增加，批处理推理的准确率会下降，而*MOSAIC* 的 基于画布的方法则显著优于基于批处理的方法（并发处理 4 或 6 帧时，准确率高出 3 倍）。随着 B 增加，准确率基于批处理的执行的准确率下降得快得多，这是因为它要求所有图像尺寸必须相同，这会导致准确率大幅下降，尤其是对于小目标。另一方面，基于画布的方法支持尺寸不一致的图像，因此能够更好地保持小目标的准确率。

5.2 消融研究

5.2.1 MOSAIC 在不同画布尺寸下的性能增益。鉴于摄像头所观测到的物体分布及其空间尺寸界限限制了可打包到画布帧上的摄像头最大数量 M，因此，增大画布帧尺寸 C 理应能够提高 M 的值（从而提升吞吐量）；并且可以认为，如果所有图块均采用其尺寸界限内的最大尺寸，准确率也会更高。然而，更大的画布帧尺寸 $C \geq 640 \times 640$ 会带来更高的计算负载，并增加每个画布帧的推理延迟。画布帧推理延迟的增加会导致累计推理吞吐量降低，即 CFPS $= M \times FPS_C \times b$，其中 FPS_C 表示批量大小为 b、画布帧尺寸为 C 时所达到的吞吐量。图 13 通过绘制联合“吞吐量—准确率”指标（定义为累计帧率 × mAP@0.5）与画布帧尺寸的关系，展示了 Okutama-Action 数据集上的吞吐量—准确率权衡。我们得出结论：无论画布帧尺寸 C如何，*MOSAIC* 的性能始终优于统一缩放与打包（Uniform-M）基线。我们还发现，首选画布帧尺寸 640×640 带来的吞吐量—准确率增益最高（15.4%），优于其他所有画布帧尺寸。

5.2.2 MOSAIC 相较于 *PS* 周期的性能提升。我们已经证明，由于基于 FCFS 的深度神经网络推理需要对来自 M 个摄像头的所有摄像头输入帧进行处理，并由此产生处理延迟，PS 操作周期性会影响 MOSAIC 所能达到的吞吐量。较短的 PS 周期会使 FCFS 处理的帧数增多，从而获得更准确的类别特定目标检测，但会降低*MOSAIC* 整体可达到的吞吐量。相反，较长的 PS 周期可能会导致“最后出现”目标的跟踪器失效率升高，并使丢失或未匹配轨迹增多，进而产生额外的图块（其中可能包含目标，也可能不包含目标），

MoS 需要对这些图块进行空间打包，从而降低任务准确率。图 14 展示了 Okutama-Action 数据集上 PS 周期、*MOSAIC* 吞吐量与 mAP@0.5 之间的关系。可以看到，PS 周期为 =30 秒时，能够恰当地兼顾高吞吐量（23 帧每秒）和高目标检测准确率（ 78.5%）这两项要求；而 PS 周期为 =60 秒时，吞吐量仅增加 1 帧每秒，mAP 却大幅下降（ ~ 10 %）。

6 讨论

在网络带宽或能源受限的情况下运行：
在带宽或能源受限的实际系统中，具备计算能力的摄像头平台还可以在传输前降低图像分辨率，以减少冗余；尤其是在边缘端会使用 *MOSAIC* 在处理过程中缩小帧尺寸的情况下。先前的研究表明，针对输入帧采用感知重要性的差异化下采样技术（例如 MRIM [33]），相比统一下采样，能够更好地保留目标细节，从而提升基于边缘的视觉推理任务的性能。此类摄像头还可以实现轻量级多目标检测器和跟踪器，例如 Fast MoT [35], ，以便在光线不足和雾天等具有挑战性的环境条件下识别特定类别的感兴趣对象。*MOSAIC* 的 MoS 操作不仅可以与这种感知重要性的传输方式共存，事实上还可以从此类摄像头端处理获益，有效消除间歇性的*PS* 阶段。

基于画布的时空相关摄像头流处理：对于具有时空相关性的摄像头流，多个目标可能出现在不同的摄像头流中，且观察者与目标之间的距离、角度和视角各不相同。对于我们的 MoS流水线，可以根据以下因素进一步微调空间尺寸界限：（i）目标在多个摄像头内部帧和摄像头间帧中的出现情况；（ii）目标的运动情况以及对更短或更长检测周期的需求（即目标被纳入画布帧进行检查的频率）。对关键性和应用相关的空间尺寸界限进行此类调整，无疑能够扩展*MOSAIC*系统的功能，使其适用于更加多样化的应用场景，这将留待未来工作进一步探索。

7 相关工作

近期关于信息物理系统实时视觉流水线的研究，既源于（1）已证明边缘端需要对多样且高分辨率的传感器流进行并发计算 [1], ，也源于（ 2 ）现实情况：用于视觉任务的最先进深度神经网络通常过于复杂，无法在资源受限的边缘设备上高效运行 [13]。

由于计算能力是关键瓶颈，实时边缘 AI 日益受到学术界关注。从系统角度来看，早期研究着重于分析和理解智能边缘平台，包括 GPU [28],CUDA 调度 [27], 以及 CPU/GPU 协同调度 [5]。另一方面，从机器学习模型的角度来看，已有研究致力于优化深度神经网络执行的灵活性 [4, 17], ，这有助于进一步将其部署到实时应用中。这些研究本质上会修改深度神经网络执行，以支持多种形式的抢占，从而能够应用相应的实时调度算法。近期，有研究提出并采用了注意力调度 [16, 21, 22]，将其作为一种新颖的数据级优化与调度策略，以实现实时边缘 AI，且无需修改深度神经网络模型或底层操作系统。不过，这些研究大多通过任务批处理来探索 GPU 的并行处理能力。为降低边缘设备处理单个视频帧时的平均推理延迟，先前的感兴趣区域（ROI）方法尝试仅在输入图像帧中选定的部分执行计算量较大的目标检测深度神经网络；这些部分通过背景减除 [29, 36], 、使用复杂度较低的“预处理”深度神经网络 [11] 以及最新的图块提取 [34]等技术识别。尽管这些技术在概念上与我们筛选高优先级瓦片的方法相似，但它们主要只关注单个传感器输入，而我们的方法则是在单个图像画布中对多个传感器流进行空间装箱。

由于网络带宽是关键瓶颈，一些方法在输入过滤的背景下探索注意力和关键性，通过限制传输的帧数来实现这一目标，所采用的技术包括帧采样 [8] 、微分类器和级联过滤器摄像头 [12]。最近，研究人员利用帧之间的时间相关性，限制摄像头仅传输帧中的特定区域 [14] ，或调整帧内感兴趣对象的分辨率 [15, 33]。为应对 AI 工作流中的优先级反转问题，近期研究（例如 [16, 22]）根据每帧的空间和时间关键性，对其调度和分辨率调整进行探索。在更广泛的计算机视觉和 AI 社区中，大量工作通过提前退出和模型选择 [7], 、模型压缩 [9],、权重量化与权重/特征图稀疏化 [31], 以及动态剪枝 [19] 来解决深度神经网络（DNN）的计算开销问题。尽管这些策略降低了边缘侧的计算负载，我们面向画布的处理方法采用 输入预处理 来加速多个多媒体传感器流的并发处理，并且可以与此类 DNN 修改技术结合使用。

8 结论

我们介绍了*MOSAIC*，一种面向关键程度、基于“空间多路复用”的方法，用于在边缘设备上进行深度神经网络推理。该方法会提取单幅图像中的高优先级区域，再将其空间打包到尺寸更小的复合画布帧中，从而确保较高的处理吞吐量。*MOSAIC* 的关键创新是尺度镶嵌（MoS）概念，这是一种多尺度平铺方法，可确保不同尺寸的目标都能以适当的尺寸呈现，同时将画布上的冗余降至最低。使用具有代表性的 Jetson TX2 边缘设备开展的实验表明，*MOSAIC* 能够大幅提升吞吐量——例如，将来自 6 个不同摄像头图像的关键区域打包到单个画布帧中时，*MOSAIC*可实现 ~ 138 帧每秒的累计吞吐量，同时在 Okutama-Action 数据集上达到 79% 的行人检测准确率。相比之下，逐帧单独处理每幅图像虽然能将准确率略微提升（ ≤ 1%），达到 80%，但吞吐量（~ 3*FPS*）却大幅降低；而将 6 幅图像

均匀打包到画布帧中虽可实现相近的吞吐量（~144 帧每秒），但准确率明显较低（71%）。在另一个车牌识别应用中也观察到了类似的提升，进一步证明了*MOSAIC*的通用性。未来，我们计划在*MOSAIC* 的空间打包策略基础上加入额外的时间打包，即提取连续帧中的高优先级区域，并将其打包到同一画布上。

致谢

本研究得到新加坡国家研究基金会通过其 NRF 研究员资助项目（NRF-NRFI05-2019-0007）的支持，并部分得到波音公司、IBM（IIDAI）、美国陆军研究实验室 W911NF-17-2-0196 以及美国国家科学基金会 CNS 20-38817 的支持。本材料中表达的任何观点、研究结果、结论或建议均属于作者的意见，不代表新加坡国家研究基金会的观点。

参考文献

[1] Tarek Abdelzaher、Yifan Hao、Kasthuri Jayarajah、Archan Misra、Per Skarin、Shuochao Yao、Dulanga Weerakoon 和 Karl-Erik Årzén。2020。云端智能与控制面临的五项挑战。《*ACM*互联网技术汇刊》（*TOIT*）20，1（2020），1–19。[2] Noga Alon、Baruch Awerbuch、Yossi Azar、Niv Buchbinder 和 Joseph Naor。2009。在线集合覆盖问题。《*SIAM*计算杂志》39，2（2009），361–370。[3] Mohammadamin Barekatain、Miquel Martí、Hsueh-Fu Shih、Samuel Murray、Kotaro Nakayama、Yutaka Matsuo 和 Helmut Prendinger。2017。Okutama-动作：用于并发人体动作检测的航拍视频数据集。载于*IEEE*计算机视觉与模式识别会议研讨会论文集。28–35。[4] Soroush Bateni 和 Cong Liu。2018。Apnet：近似感知实时神经网络。载于*2018*年*IEEE*实时系统研讨会（*RTSS*）。IEEE，67–79。[5] Soroush Bateni、Zhendong Wang、Yuankun Zhu、Yang Hu 和 Cong Liu。2020。通过集成CPU/GPU内存管理协同优化性能与内存占用：在自动驾驶平台上的实现。载于*2020*年*IEEE*实时与嵌入式技术及应用研讨会（*RTAS*）。IEEE，310–323。[6] Sourav Bhattacharya 和 Nicholas D. Lane。2016。面向可穿戴设备受限资源推理的深度学习层稀疏化与分离。载于第*14*届*ACM*嵌入式网络传感器系统会议光盘论文集（*SenSys '16*）。计算机协会，176–189。[7] Tolga Bolukbasi、Joseph Wang、Ofer Dekel 和 Venkatesh Saligrama。2017。用于高效推理的自适应神经网络。载于国际机器学习会议。PMLR，527–536。[8] Tiffany Yu-Han Chen、Lenin Ravindranath、Shuo Deng、Paramvir Bahl 和 Hari Balakrishnan。2015。Glimpse：移动设备上的连续实时目标识别。载于第*13*届*ACM*嵌入式网络传感器系统会议论文集。155–168。[9] Yu Cheng、Duo Wang、Pan Zhou 和 Tao Zhang。2018。深度神经网络的模型压缩与加速：原理、进展与挑战。《*IEEE*信号处理杂志》35，1（2018），126–136。[10] Yerim Chung 和 Myoung-Ju Park。2015。逆装箱问题注记。《信息处理快报》115，1（2015），60–68。[11] M. Gao、R. Yu、A. Li、V. I. Morariu 和 L. S. Davis。2018。用于大幅图像快速目标检测的动态放大网络。载于*2018*年*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*）。IEEE计算机学会，洛斯阿拉米托斯，加利福尼亚州，美国，6926–6935。[12] Kevin Hsieh、Ganesh Ananthanarayanan、Peter Bodik、Shivaram Venkataraman、Paramvir Bahl、Matthai Philipose、Phillip B Gibbons 和 Onur Mutlu。2018。Focus：以低延迟和低成本查询大型视频数据集。载于第*13*届*USENIX*操作系统设计与实现研讨会（*OSDI 18*）。269–286。[13] Yigong Hu、Shengzhong Liu、Tarek Abdelzaher、Maggie Wigness 和 Philip David。2021。探索图像缩放以优化基于关键性的机器感知。载于*2021*年*IEEE*第*27*届嵌入式与实时计算系统及应用国际会议（*RTCSA*）。IEEE，169–178。[14] Junchen Jiang、Ganesh Ananthanarayanan、Peter Bodik、Siddhartha Sen 和 Ion Stoica。2018。Chameleon：可扩展的视频分析自适应。载于*2018*年*ACM*数据通信特别兴趣组会议论文集。253–266。[15] Shiqi Jiang、Zhiqi Lin、Yuanchun Li、Yuanchao Shu 和 Yunxin Liu。2021。边缘设备上具有可调延迟的灵活高分辨率目标检测。载于第27届移动计算与网络年度国际会议论文集。559–572。[16] Woosung Kang、Siwoo Chung、Jeremy Yuhyun Kim、Youngmoon Lee、Kilho Lee、Jinkyu Lee、Kang G Shin 和 Hoon Sung Chwa。2022。DNN-SAM：面向实时目标检测的DNN拆分与合并执行。（2022年5月）。[17] Jung-Eun Kim、Richard Bradford 和 Zhong Shao。2020。Anytimenet：控制深度神经网络架构中的时间-质量权衡。载于2020年欧洲设计、自动化与测试会议暨展览（DATE）。IEEE，945–950。[18] Rayson Laroca、Evair Severo、Luiz A Zanlorensi、Luiz S Oliveira、Gabriel Resende Gonçalves、William Robson Schwartz 和 David Menotti。2018。基于YOLO检测器的鲁棒实时自动车牌识别。载于2018年国际神经网络联合会议（IJCNN）。IEEE，1–10。[19] Ji Lin、Yongming Rao、Jiwen Lu 和 Jie Zhou。2017。运行时神经网络剪枝。《神经信息处理系统进展》30（2017）。[20] Tsung-Yi Lin、Michael Maire、Serge Belongie、James Hays、Pietro Perona、Deva Ramanan、Piotr Dollár 和 C Lawrence Zitnick。2014。Microsoft COCO：常见场景中的常见物体。载于欧洲计算机视觉会议。Springer，740–755。[21] Shengzhong Liu、Xinzhe Fu、Maggie Wigness、Philip David、Shuochao Yao、Liu Sha 和 Tarek Abdelzaher。2022。关键性驱动视觉机器感知中的自提示实时注意力调度。载于第28届IEEE实时与嵌入式技术及应用研讨会（RTAS）论文集。[22] Shengzhong Liu、Shuochao Yao、Xinzhe Fu、Rohan Tabish、Simon Yu、Heechul Yun、Lui Sha 和 Tarek Abdelzaher。2020。消除任务关键型机器推理流水线中的算法优先级反转。（2020年12月）。[23] Wei Liu、Dragomir Anguelov、Dumitru Erhan、Christian Szegedy、Scott Reed、Cheng-Yang Fu 和 Alexander C Berg。2016。SSD：单次多框检测器。载于欧洲计算机视觉会议。Springer，21–37。[24] M McKerns、P Hung 和 M Aivazis。2021。mystic：高度约束的非凸优化与不确定性量化。https://github.com/uqfoundation/mystic [25] 英伟达。2022。Jetson TX2开发套件。英伟达（2022）。https://developer. nvidia.com/embedded/jetson-tx2 [26] 英伟达。。Jetson Xavier NX开发套件。英伟达（2022）。https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit [27] Ignacio Sañudo Olmedo、Nicola Capodieci、Jorge Luis Martinez、Andrea Marongiu 和 Marko Bertogna。2020。剖析CUDA调度层级：性能与可预测性视角。载于2020年IEEE实时与嵌入式技术及应用研讨会（RTAS）。IEEE，213–225。[28] Nathan Otterness 和 James H Anderson。2020。AMD GPU作为支持实时工作负载的英伟达替代方案。载于第32届Euromicro实时系统会议（ECRTS 2020）。Schloss Dagstuhl-Leibniz-Zentrum für Informatik。[29] Theodore Stone、Nathaniel Stone、Puneet Jain、Yurong Jiang、Kyu-Han Kim 和 Srihari Nelakuditi。2019。迈向边缘可扩展的视频分析。载于2019年第16届IEEE传感、通信与网络年度国际会议（SECON）。IEEE，1–9。[30] Ultralytics。2022。Ultralytics/yolov5：PyTorch、ONNX、CoreML和TFLite中的Yolov5。GitHub（2022 5）。[31] 2020 https://github.com/ultralytics/yolov Thomas Verelst 和 Tinne Tuytelaars。。动态卷积：利用空间稀疏性实现更快推理。载于IEEE/CVF计算机视觉与模式识别会议论文集。2320–2329。[32] Xin Wang、Fisher Yu、Zi-Yi Dou、Trevor Darrell 和 Joseph E. Gonzalez。2018。SkipNet：在卷积网络中学习动态路由。载于《计算机视觉——ECCV 2018：第15届欧洲计算机视觉会议》。Springer-Verlag，420–436。[33] Ji-Yan Wu、Vithurson Subasharan、Tuan Tran 和 Archan Misra。2022。MRIM：为低功耗普适视觉任务实现混合分辨率成像。载于2022年IEEE普适计算与通信国际会议（PerCom）。IEEE，44–53。[34] K. Yang、J. Yi、K. Lee 和 Y. Lee。2022。FlexPatch：面向设备端高分辨率实时视频分析的快速准确目标检测。载于IEEE INFOCOM。IEEE。[35] Yukai Yang。2020。FastMOT：基于Deep SORT和KLT的高性能多目标跟踪。https://doi.org/10 5281 4294717 。/zenodo。Juheon Yi、Sunghyun Choi 和 Youngki Lee。。EagleEye：拥挤城市空间中的可穿戴相机人员识别。载于第26届移动计算与网络年度国际会议论文集。计算机协会，纽约州纽约市，美国，文章4，14页。[37] Wuyang Zhang、Zhezhi He、Luyang Liu、Zhenhua Jia、Yunxin Liu、Marco Gruteser、Dipankar Raychaudhuri 和 Yanyong Zhang。2021。Elf：通过内容感知并行卸载加速高分辨率移动端深度视觉。载于第27届移动计算与网络年度国际会议论文集。201–214。[36] 2020