纪红蕾,丁晗,赵朝阳,等. 面向移动端的多人高效行为识别方法 [J]. 江西师范大学学报(自然科学报),2023,47(3):317-324.
JI Honglei, DING Han, ZHAO Chaoyang, et al. The efficient multi-person action detection on mobile devices [J]. Journal of Jiangxi Normal University (Natural Science),2023,47(3):317-324.

文章编号:1000-5862(2023)03-0317-08

面向移动端的多人高效行为识别方法

纪红蕾[1],丁　晗[2,3],赵朝阳[2],唐　明[2],王金桥[2]
(1. 中车工业研究院有限公司,北京　100071;2. 中国科学院自动化研究所,北京　100190;
3. 中国科学院大学人工智能学院,北京　100049)

摘要:视频行为识别是有前景并且有挑战性的任务,但现有的大部分方法依赖大量的参数和运算. 该文提出了一种基于连续多帧缓存的高效行为识别方法:首先针对多人场景的问题,输入单帧图片,结合人体检测器给出所有人的动作分类和得分;然后通过使用时序位移模块缓存之前帧的特征,使网络具有时序信息处理的能力. 实验结果表明:该方法取得了较好的轻量化效果,搭配额外的目标检测网络,可以做到多人场景实时的行为识别,体现了一定的识别速度和准确率优势.

关键词:多人行为检测;轻量化;移动端
中图分类号:TP 391.41　　**文献标志码**:A　　**DOI**:10.16357/j.cnki.issn1000-5862.2023.03.12

0　引言

视频理解是计算机视觉的一个重要分支,视频行为识别是视频理解的核心领域. 近年来,对于视频行为识别,CNN(convolutional neural network)[1]和 Transformer[2]逐渐成为主流. 此外,大型视频数据集的发布加快了这一进程,并在一定程度上推动了该领域的各种算法研究. 尽管在识别性能方面取得了显著的成功,但大多数方法依赖于大数据和大模型,具有较高的计算复杂度,因此它们更倾向于部署在云服务器上. 而目前的行为识别研究正逐渐转向对视频流的实时计算和分析,如在线动作识别、异常行为检测和行为预测等. 大量视频在传输时可能会出现延迟和数据丢失的情况. 另外,对于一些应用领域(如智能家居和城市监控等),对数据保护的需求也较高. 这些数据应该尽量避免传输,否则容易出现隐私泄露.

总体来看,用于视频分析的深度神经网络的发展仍然滞后于图像分析,其主要原因是引入了时间维度所带来的计算成本. 视频的时间维度包含有价值的运动信息,这些信息在视频行为识别任务中起着关键作用. 早期的一些方法是直接从图像扩展到视频领域,即直接增加时间维度(如 C3D[3]、I3D[4]等),在多个基准数据集上取得了较优的结果,但同时它们都伴随着极高的计算负载. 虽然也有一些方法将时空 2 个维度进行分解(如 R(2+1)D[5]),但在实际应用中比较少见. Transformer 相对于 CNN 增加了更多全局特征,在很多主流方法中都被使用,但它却是朝着做大做强的方向发展. 一些轻量化的方法(如 MobileViT[6]、EdgeViT[7]、TrtViT[8]等)将自注意力模块插入 CNN 结构中,在保证精度的情况下减少了参数量. 在视频处理方面,TokenLearner[9]、TokenMerging[10]等考虑用减少 token 的方式来减少 4 次方的计算量. 现在虽然也有了一些轻量化 ViT 的部署方案,但在参数量相近时效果不如 MobileNetV2[11]好,其原因是在实际部署中对自注意力算子的优化支持不够好,可见轻量化之路任重而道远.

目前提出的大部分方法是基于全图进行分类,理由是常用的数据集 Kinetics[12]、UCF101[13]等都是

收稿日期:2022-11-15
基金项目:国家自然科学基金(61976210,62176254)资助项目.
作者简介:纪红蕾(1995—),女,河北衡水人,助理工程师,主要从事轨道交通智能产品应用技术研究. E-mail:jhl@crrc.tech. 前两位作者对本文有同等贡献.

全图划分的,这意味着一段视频只有一个类别. 这基本脱离了实际应用,现实的大部分场景是多人的. 对于密集场景,需要经过人体检测和行为分类 2 个部分来获得所有人的行为类别. 多人场景的行为识别也有端到端的方法,如 YOWO[14]、WOO[15]. 但是这些方法比较难部署(因为一些平台不支持 3D 算子),这意味着无法批处理,只能单帧进行操作.

基于以上原因,本文提出了在多人场景下基于多帧缓存的高效行为识别处理流程,其和常规方法的对比如图 1 所示. 借助时序位移模块,2D 的骨干网络(backbone)具备处理时序信息的能力. 具体来说,它能缓存当前帧的大部分有用信息并传递给后续帧,模拟了 3D 网络的处理模式. 在此基础上,使用了 YOLOv5 作为人体检测器,完成了多人行为分类的任务. 该方法在视频行为识别任务上兼具轻量化和高准确率的特点. 此外,本文尝试了多种骨干网络和设置,对其进行了深入分析,测试了其速度和性能,均能满足多人场景的实际需求.

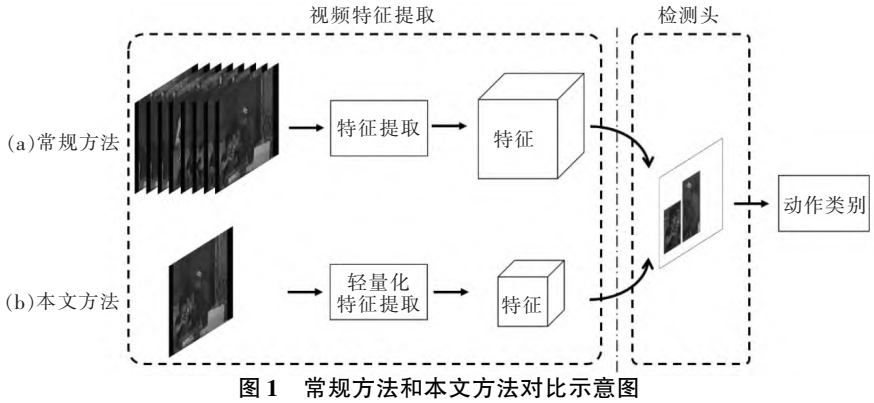

图 1 常规方法和本文方法对比示意图

1 相关工作

1.1 多人场景行为识别

大部分的行为识别数据集是全图分类的,目前多人场景的分类数据集只有 AVA[16]. 有一些方法使用了端到端的方式,如 YOWO 和 WOO,它们的结构有些不同. YOWO 将动作定位和行为分类拆成双分支,最后进行融合并使用了类似 YOLO[17] 的检测头. WOO 则更为巧妙地运用了统一的骨干网络,检测头参照了 Sparse RCNN[18],其他的工作则使用了共享的骨干网络,着重于探索更好的视频特征. 然后搭配额外的人体检测器,用 ROI Align 提取特征进行行为分类,如 VideoMAE[19]、MViT[20] 等. 本文的工作属于后者,并针对时序特征提取的网络进行轻量化设计,以完成多人实时分类的任务.

1.2 轻量 CNN

当输入从图像变为视频,一个最直接的想法就是将 2D 的卷积扩展时间维度到 3D 卷积. 最早的工作是 C3D,验证了不同深度的卷积核对结果的影响. 但由于 3D 卷积的复杂度太高,因此很多学者考虑将时间维度简化. R(2 +1)D 将时间和空间解耦,不仅增加了非线性而且更容易优化. TSM[21] 是一种零参数、零计算的方法,它将相邻帧之间的信息进行位移,以 2D 的计算量完成了 3D 的效果. 但它们只适用于短时视频的分类,没有考虑长视频多人场景的问题,而这是本文的重要目标.

1.3 轻量 Transformer

自从 SlowFast[22] 的出现,3D 卷积似乎已经做到了上限. 它可以看作一个通用的行为识别框架,可以自行替换骨干网络和输出头. 直到 Transformer 的出现,人们开始了新的探索. 目前的 Transformer 网络在朝着通用模型的方向发展,如自监督和多模态,但这其中还是有不少方法有轻量化的影子. Swin Transformer[23] 将空间划分为不同的窗口,在内部计算自注意力. X-CLIP[24] 虽然使用了 CLIP[25] 的预训练模型,但需额外提出相邻帧之间的跨帧交互模块,值得借鉴. 其他针对 ViT[26] 的改进也非常多,例如 MViT[27] 使用了 Multi Head Pooling Attention,在计算注意力时降低了计算量;MeMViT[28] 压缩了每一帧的缓存信息,在处理当前帧时需要和 cache 结合处理. TokenLearner 提出了 Tokenlearner 和 Tokenfuser,将多个 token 缩小到 8 个 token 再还原. 此外,MobileViT 和 EdgeViT 也是比较好的轻量化方法,它们使用了代理 token 的思想,交互少量 token 的信息再扩散到其他的 token 上. 虽然它们是针对图像领域,但是也可以很容易地扩展到视频上. 在实际应用中,考虑 Transformer 的结构需要额外优化,并不如 CNN 方便,不过本文仍考虑了这些网络,进行了基本的测试.

2　基于连续多帧缓存的多人高效行为识别方法

一般的行为检测模型参数量大,且视频帧的分支需要存够一定数量再进行处理,这就意味着一段时间只能给出 1 个结果. 该文多帧缓存的整体流程如图 2 所示. 为了满足多人场景分类任务的需要,行为识别流程分为人体检测器和行为分类器 2 个部分. 人体检测器可以根据需要选择任意的模型,输入关键帧图片,标注出图上的人体框位置. 另一部分行为分类器同样接收这一帧输入,虽然使用 2D backbone,但额外增加了多帧缓存模块. 该模块能够存储之前帧的有用信息替换当前帧的部分特征,并在当前帧提取完特征后进行更新. 最后在检测头中使用 ROI Align 将人体框的位置在时空特征图上等比例映射,根据对应特征进行分类. 本流程能做到 1 帧给出 1 个结果,同时具备 2D backbone 的速度和 3D backbone 的精度. 下面将对细节详细介绍.

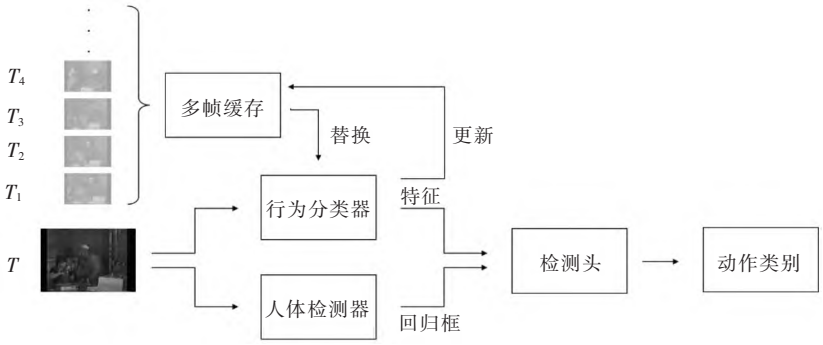

图 2　基于连续多帧缓存的多人高效行为识别总体流程图

2.1　高效行为分类器

从图像到视频领域多了时间维度,一个直接的想法就是从 2D 变为 3D 卷积. 如 3D 卷积神经网络可以同时学习时空特征,但是其训练和推理成本较高,模型的参数量和计算量差距非常明显,并且 3D 卷积和自注意力更难优化. 在推理时若在边缘设备上进行部署,则挑战性较大,并且很多平台并不支持 3D 算子;同时,对于实时视频识别,无法达到要求. 想要以单帧的特征完成时序行为识别任务,可以参照文献[21]中提出的 TSM(temporalshiftmodule)(见图 3).

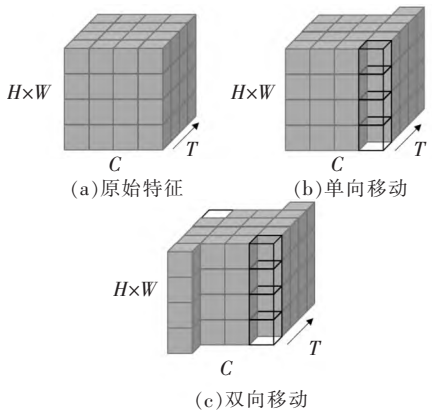

图 3　时序位移模块操作示意图

在 CNN 中,时序位移模块对时间建模提供了新的思路. 具体来讲,用 $\boldsymbol{V}\in\mathbf{R}^{N\times C\times T\times H\times W}$ 表示一个视频,其中 N 是批处理大小,C 是通道数,T 是时间维度,H 和 W 是空间分辨率. 图 3 可视化着重于通道和时间维度,省略了 batch 维度. 对于 2D 卷积神经网络而言,只能将时间拼到 batch 上处理,不会对时间进行建模(见图 3(a)). 时序移位模块沿着时间维度向相邻帧移动通道,其定义如下:

$$T_{sm}(\boldsymbol{X}_t)=\alpha\boldsymbol{X}_{t-1}+\beta\boldsymbol{X}_{t+1}+(1-\alpha-\beta)\boldsymbol{X}_t,$$

其中 $\boldsymbol{X}_t\in\mathbf{R}^{C\times H\times W}$ 是 t 时刻的单帧特征,α、$\beta\in(0,0.50)$ 为移动比例. 图 3(b)和图 3(c)分别是单向($\alpha=0.25,\beta=0$)和双向($\alpha=0.25,\beta=0.25$)移动. 经过通道的移动操作,相邻帧的信息和当前帧的信息得到了交互. 在文献[21]中,将该模块插入到 2 维卷积之前,实现零计算、零参数的时间建模为

$$\boldsymbol{X}_t=C_{\mathrm{onv2d}}(T_{SM}(\boldsymbol{X}_t)).$$

同样地,可以将以上结论扩展到 Transformer 上. 与 CNN 不同,Transformer 能获取到远距离特征进行全局建模,虽然计算复杂度和 patch 的平方成正比,但目前已有一些小模型的计算量低于常规 CNN. TokShift[29]是 TSM 的一个变体,以 ViT 模型为基准,在 Transformerblock 的 2 个 layernorm 之前,将 cls token 位置的 patch 进行移动. 这是因为:cls token 已经聚集了当前帧的有效信息,可以以较小的代价达到同样的目的. 但这不适用于没有 cls token 的 Transformer 模型,包括一些 CNN-Transformer 的复合模型. 同时由于 2D 的 backbone 缺少时序信息,所以需要交互更多的 patch 特征. 因此同样将时序移动用在 2D 的 Transformer Block 中所得模型为

$$\boldsymbol{X}_t=M_{HSA}(\ln(T_{SM}(\boldsymbol{X}_t))),\boldsymbol{X}_t=M_{LP}(\ln(\boldsymbol{X}_t)).$$

图4 给出了 CNN 和 Transformerblock 使用了 temporal shift 操作的示意图. 相对于其他工作而言,本文将 2 者统一,更具有泛用性. 通过这个模块,可以将 2D 网络转换为具有处理时序数据能力的网络,完成高效的行为识别任务. 这一操作带来的额外开销仅在于数据的移动,在后面的消融试验中会展示它们的对比信息,速度差异在可接受范围内.

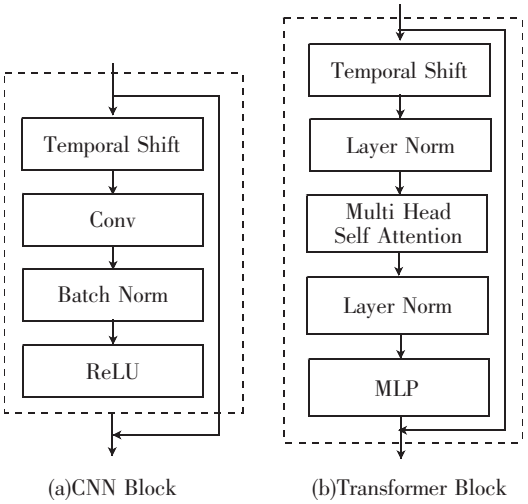

图 4 CNN 和 Transformerblock 使用时序位移的结构图

2.2 连续多帧缓存

通过以上模块能够大幅缩短时序特征提取的时间,行为识别的分支将不再是边缘部署的瓶颈. 在推理时,人体检测器和行为分类器同时接受单帧图片的输入. 人体检测器可以任意选择,给出图中所有人体框的坐标和置信度. 另一个分支使用 2.1 节所述的高效行为分类器,并在检测头中根据回归框预测所有人的类别. 连续多帧缓存推理流程图如图 5 所示. 在处理当前帧时,会将部分网络层的输出特征缓存下来,在处理后一帧时替换对应层的部分特征.

这里需要注意的是训练和推理的流程不尽相同. 在训练时可以批量地对一个 clip 内的视频帧进行移动,但是在推理时模型只接受单帧输入,因此需要将前一帧的特征缓存下来,替换当前时刻中间层输入的部分特征,并将此时的缓存更新. 随着时间的推移,特征会在相邻帧之间传递. 此外,单向和双向移动的选择也有不同. 对于实时的视频流而言,只能获取到之前时刻的信息,而在训练时却可以双向移动,这会损失一些精度,但是微乎其微,后续会在实验部分中验证这一点.

此外,缓存多少层的特征可以根据网络而定. 缓存的特征越多,时序能力越强,但同时特征移动所带来的时间开销会加大. 整体流程的代码用 PyTorch 可以形式化(见图 6).

总体来说,连续多帧缓存相比常规的行为识别流程更新颖. 常规的方法需要使用队列,在存满一定数量之后送入 3D 网络推理,同时抽取 1 个 clip 的关键帧(通常是中间 1 帧)预测人体框. 这一流程不仅耗时,而且 1 个 clip 只能输出 1 个结果,“刷新率”较低. 上述方法不仅在速度上有显著优势,而且 1 帧有 1 次输出,这对于多路视频的处理更加友好.

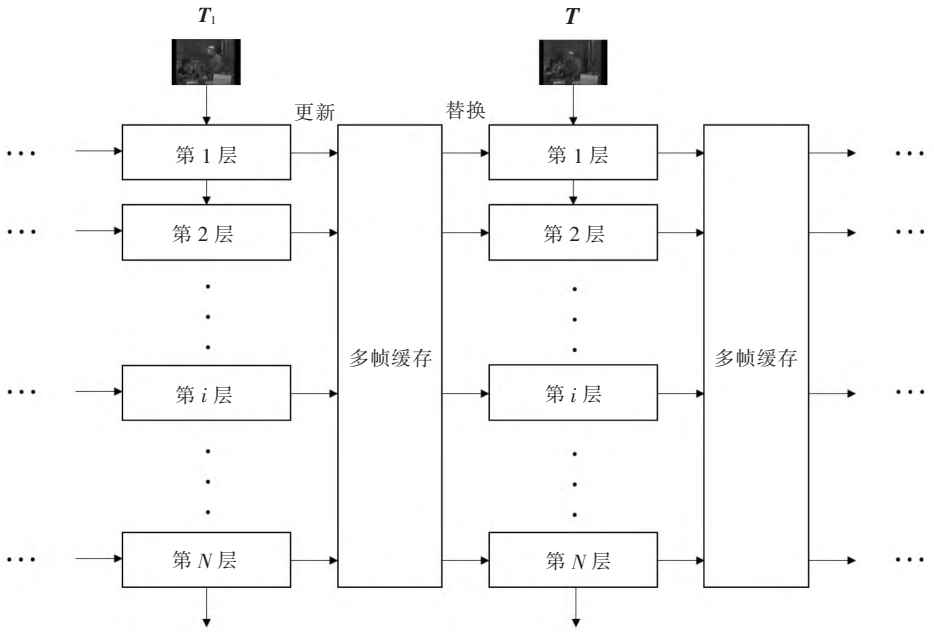

图 5 连续多帧缓存推理流程图

算法 1　连续多帧缓存的推理伪代码

```
Class CMC( ):
def __init__( ):
    self. frame_cache = [ ]
    self. layers = Transformer( depth = 12)
    self. head = DetectionHead( num_class = 80)
def forword( x):
output_cache = [ ]
    for i, layer in enumerate( self. layers):
    x, c = layer( x, self. frame_cache[ i])
output_cache. append( c)
    x = self. head( x)
    self. frame_cache = output_cache
    return x
```

图 6　用 PyTorch 形式化代码

2.3　分类头和检测头

若是全图分类的情况,则只需要分类头,经过全连接得到最后的结果. 若是多人场景的检测问题,则人体检测器是必不可少的. 这一部分可以使用已有的 AnchorBased 或者 AnchorFree 检测器,考虑速度的要求和部署的简便性,可以选择 YOLOv5 或者 YOLOv7[30]. 它们都提供了完整的部署方法,包括 TensorRT 和 NCNN 等,方便在边缘设备上运行. 人体检测器给出人体框的位置和得分,然后需要 ROI Align 将其映射到视频分支. 但 ROIAlign 的算子在某些平台上不支持,因此需要自己重新定义.

从图 1 可以看出:由于输入为单帧图片,所以视频分支提供给检测头的特征也更少. 假设 3D backbone 接受视频 clip 输入为 16 帧,那么该方法提取的特征仅为它的 1/16. 一般来说前者会在检测头中对时间维度做 Average Pooling,这相当于对特征做了平滑处理;而后者已经具备了有用的时序信息,因此少量特征仍然能有比较好的结果.

3　实验结果与分析

3.1　数据集与实验设置

实验主要使用 UCF101 和 AVA v2.2 作为数据集来评估插入时序位移模块的模型性能以及在最终部署后的速度. 仅使用 UCF101 测试了单向和双向模型的精度差异,后续实验都基于数据集 AVA. UCF101 是 2012 年发布的行为分类数据集,这些视频来自 YouTube,包含了 101 个行为类别,共 13 320 个视频,主要有 5 大类:人-物交互、肢体运动、人-人交互、弹奏乐器和运动. 数据集 AVA 由 YouTube 公开视频的 URL 组成,这些视频被 80 个原子动作标注,如走路、踢东西、握手等,所有动作都具有时空定位,产生 5.76 万个的视频片段、9.60 万个人类动作以及 21.00 万个的动作标签.

除非特别说明,本文都选择在 K400 上的预训练模型,优化器选择 SGD,学习策略为 StepLR. 学习率根据骨干网络不同会有所差异. 在 UCF101 上训练 25 个 epoch,在 AVA 上训练 20 个 epoch. 视频进行均匀采样,在 64 帧中采样 8 帧. 分类数据集选择全连接头,检测数据集额外增加了 ROI Align,后面都接 softmax 预测最终结果. 考虑 Transformer 的位置编码,模型训练和推理尺寸均为 224×224,但由于原版 MobileViT 需要输入尺寸为 64 的倍数,所以它比较特殊,选择 256×256.

3.2　参数量和计算量比较

表 1 将主流方法和本文提出的方法进行对比. 因为重点在于轻量化部署,所以更关注模型的参数量和计算量. 由表 1 可以看出:常规的方法即使是比较小的模型也难以在移动端跑到实时,而本文的方法却较有优势.

表 1　不同模型的计算量和参数量对比

模型	骨干网络	计算量	参数量/M
SlowFast	Resnet101	106.0	53.7
ViT	Base	134.7	85.9
TimeSformer	Large	2 380.0	121.4
Video Swin Transformer	Large	604.0	197.0
本文算法	Resnet50	33.0	24.3

3.3　消融实验

3.3.1　*双向和单向模型*　双向和单向模型的区别在于当前帧是否可以获取到前或后一帧的信息. 在训练时可以为双向,但在测试时只能为单向. 想要将其统一,必须确保 2 者的准确率不能差异太大. 这里选择 ResNet50[31]、MobileNetV2 和 Swin-Tiny 等 3 个模型作为骨干网络,表 2 为 UCF101 的结果.

表 2　单向和双向移动的 top1 和 top5 准确率　%

骨干网络	单向		双向	
	Top1	Top5	Top1	Top5
ResNet50	93.31	99.26	94.19	99.37
MobileNetV2	90.14	99.02	91.52	99.10
Swin-Tiny	83.70	96.85	84.09	96.78

从表 2 可以看出:单向和双向之间的差距在 1% 以内. 需要说明的是,对于 2D 版本,ResNet50 和 MobileNetV2 已经有了 Kinetics 的预训练,而由于

Swin-Tiny 仅使用 ImageNet 1k 进行初始化,并没有在 K400 上进行预训练,所以效果会差一些. 本文的重点在于整套流程的可行性,因此没有在预训练上花费更多时间. 若有更好的预训练模型,则这个结果会更好.

3.3.2 Transformer 特征移动位置 由于没有额外的预训练支持,所以对 Swin-Tiny 做了额外的消融实验,改变模块插入在多头自注意力中的位置和移动的比例,在 AVA 数据集上得到的结果如表 3 所示.

表 3 时序位移模块插入位置和移动比例对 mAP 的影响

插入位置	移动比例	mAP
MHSA 之前	1/8	15.4
MHSA 之后	1/8	13.9
MHSA 之前	1/4	15.2
MHSA 之后	1/4	15.2

从模型的结果来看模型是有效的,且在 MHSA 之前使用效果更好. 但相对于 ResNet 和 MobileNet 而言准确率确实还要低一些. 理论上 Swin-Tiny 和 ResNet50 的参数量相近,它们的结果应该也相似,由此猜测可能是缺少 K400 预训练这一关键因素.

3.3.3 在 AVA 数据集上的检测性能 本文更侧重于多人场景的分类任务,因此在 AVA v2.2 上进行检验. 模型训练基于 SlowFast,结果如表 4 所示.

表 4 不同骨干网络在 AVA 数据集上的检测结果

骨干网络	关键帧(左)	关键帧(中)	关键帧(右)
ResNet50 3D		17.5	
ResNet50 2D	17.9	18.5	16.6
MobileNetV2	16.0	16.6	14.7
Swin-Tiny		15.4	

从 ResNet50 2D 和 ResNet50 3D 的结果来看,2D 卷积确实能达到 3D 卷积的效果,甚至准确率更高. 表 4 第 1 行为 SlowOnly 的结果,使用的是官方的骨干网络,并在采样率 8×8 上微调. 这个值会比预测值稍微低一些. 因为 SlowFast 在推理时把短边缩放到 256,而考虑 Transformer 的位置编码可能需要额外插值,因此在训练和推理时都固定尺寸为 224×224.

此外,后续还对关键帧的选择进行了消融实验. 因为多帧缓存需要获取当前帧和之前帧进行交互,因此其中的关键帧并不是一个 clip 的中间,而是位于末尾的一帧. 考虑到这一点,本文对关键帧选择进行了消融实验. 表 4 最右边一列的结果会差一些,但在可接受范围内.

3.3.4 损失函数和优化器选择 下面将对损失函数和优化器进行了消融实验,模型选择效果较好的 ResNet50 2D. 从表 5 和表 6 可以看出,多人场景的分类任务可能更适合用 BCE 和 SGD.

表 5 不同损失函数对结果的影响

损失函数	mAP
BCELoss	18.5
SCELoss	3.8
BCEWithLogitsLoss	4.6
MSELoss	13.3

表 6 不同优化器对结果的影响

优化器	mAP
SGD	18.5
Adam	16.3

3.3.5 采样间隔 在实际部署时,本文的流程可以做到 1 帧给出 1 个结果,保证了实时性. 但同时采样间隔会有变化,在训练时应采取连续的 8 帧作为输入. 因此选择轻量的 MobileNetV2 对不同的采样间隔进行分析,结果如表 7 所示.

表 7 不同采样率对结果的影响

采样数×采样间隔	mAP
8×8	14.7
8×4	16.8
8×2	16.0
8×1	15.8

采样频率过低会导致一些问题,如走路这一动作,若采样帧数间隔较大,则可能会错认为跑步的动作. 8×1 意味着视频所有的帧都参与检测和分类,也是本文最终的选择.

3.4 部署速度

为了保证实际场景的可用性,整体流程需要在边缘设备上部署. 设备选择 JetsonNX 和 JetsonNano,将模型分别转换为 TensorRT 和 NCNN 进行测试. 本节先给出在 TensorRT 下视频多帧缓存的测试结果,表 8 为不同模型的 FPS(Frames Per Second).

表 8 在 TensorRT 下不同网络的 FPS 帧·s^{-1}

骨干网络	添加时序位移		不添加时序位移	
	fp32	fp16	fp32	fp16
ResNet50	44.3	80.0	45.0	88.6
MobileNetV2	74.6	109.2	111.6	126.5

对于 ResNet50 而言,由于计算量更大,所以特征移动对于它的影响不如 MobileNetV2 明显. 表 9 使用了 TensorRT 的工具对模型的运算时间进行更

进一步地量化.

表 9 对比的是给出一个结果所需要消耗的时间. 对于输入为 8 帧的 3D 时序模型,尽管可以批处理,但是本文方法还是具有优势. 此外,因为没有对在 Transformer 中的算子进行额外优化,所以在 TensorRT 上的支持不如 CNN,其耗时也会更高.

表 9　在 TensorRT 下不同骨干网络的耗时分析　ms

骨干网络	添加时序位移		不添加时序位移		常规 8 帧模型	
	fp32	fp16	fp32	fp16	fp32	fp16
MobileViT			4.03	3.13	19.74	10.43
MobileNetV2	4.43	2.63	4.26	2.13	16.60	7.36
ResNet50	15.52	7.46	14.33	4.37	85.00	21.06
Swin-Tiny			22.87	9.78	150.25	58.28

除此之外还需考虑使用 CPU 这一更为困难的任务. 这一部分会串接人体检测器,但是不同的检测器速度的差异很大,本文选择了 YOLOv5s 进行人体检测,不同模型的 FPS 如表 10 所示.

表 10　在 NCNN 下不同模型的 FPS　帧 · s^{-1}

行为分类器 + 人体检测器	FPS(fp16)
MobileNetV2 + YOLOv5s	8.84
ResNet50 + YOLOv5s	5.56

4　结论

本文提出了基于连续多帧缓存的多人高效行为识别方法. 在提取特征时使用了高效的行为分类器,具备 2D 网络的速度和 3D 网络的精度. 并且给出了在多人场景下的连续多帧缓存处理流程,能对每一帧输出一次分类结果. 在实验中通过不同的骨干网络和参数对比,验证了其可行性和有效性,最后在多人场景中使用 Jetson 进行部署. 针对在实际场景中计算资源有限和实时性的要求,本文提出的方法有其独特的优势.

5　参考文献

[1] KRIZHEVSKY A,SUTSKEVERI, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM,2017,60(6): 84-90.

[2] VASWANI A,SHAZEER N,PARMAR N, et al. Attention is all you need [EB/OL]. [2022-06-12]. https://arxiv.org/abs/1706.03762v3.

[3] TRAN D,BOURDEV L,FERGUS R,et al. Learning spatiotemporal features with 3D convolutional networks [EB/OL]. [2022-06-12]. https://ieeexplore. ieee. org/document/7410867.

[4] CARREIRA J,ZISSERMAN A. Quo vadis, action recognition? a new model and the Kinetics dataset [EB/OL]. [2022-06-12]. https://arxiv. org/pdf/1705.07750. pdf.

[5] TRAN D, WANG Heng, TORRESANI L, et al. A closer look at spatiotemporal convolutions for action recognition [EB/OL]. [2022-06-15]. https://ieeexplore. ieee. org/document/8578773.

[6] MEHTA S, RASTEGARI M. MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer [EB/OL]. [2022-06-15]. https://arxiv. org/abs/2110.02178.

[7] PAN Junting, BULAT A, TAN Fuwen, et al. EdgeViTs: competing light-weight CNNS on mobile devices with vision transformers [EB/OL]. [2022-06-17]. https://arxiv. org/abs/2205.03436.

[8] XIA Xin, LI Jiashi, WU Jie, et al. TRT-ViT: TensorRT-oriented vision transformer [EB/OL]. [2022-06-15]. https://arxiv. org/pdf/2205.09579. pdf.

[9] RYOO M S, PIERGIOVANNI A J, ARNAB A, et al. TokenLearner: what can 8 learned tokens do for images and videos? [EB/OL]. [2022-06-16]. https://arxiv. org/abs/2106.11297.

[10] BOLYA D, FU Chengyang, DAI Xiaoliang, et al. Token merging: your ViT but faster [EB/OL]. [2022-06-16]. https://arxiv. org/abs/1801.04381.

[11] SANDLER M, HOWARD A, ZHU Menglong, et al. MobileNetV2: inverted residuals and linear bottlenecks [EB/OL]. [2022-06-16]. https://ieeexplore. ieee. org/document/8578572.

[12] KAY W, CARREIRA J, SIMONYAN K, et al. The Kinetics human action video dataset [EB/OL]. [2022-06-19]. https://arxiv. org/pdf/1705.06950. pdf.

[13] SOOMRO K,ZAMIR A R,SHAHM. UCF101: a dataset of 101 human actions classes from videos in the wild [EB/OL]. [2022-06-16]. https://arxiv. org/abs/1212.0402.

[14] KÖPÜKLÜO, WEI Xiangyu, RIGOLL G. You only watch once: a unified CNN architecture for real-time spatiotemporal action localization [EB/OL]. [2022-06-19]. https://arxiv. org/abs/1911.06644v3.

[15] CHEN Shoufa, SUN Peize, XIE Enze, et al. Watch only

once: an end-to-end video action detection framework [EB/OL]. [2022-06-25]. https://ieeexplore.ieee.org/document/9710781.
[16] GU Chunhui, SUN Chen, ROSS D A, et al. AVA: a video dataset of spatio-temporally localized atomic visual actions [EB/OL]. [2022-06-25]. https://arxiv.org/pdf/1705.08421.pdf.
[17] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [EB/OL]. [2022-06-25]. https://arxiv.org/abs/1612.08242.
[18] SUN Peize, ZHANG Rufeng, JIANG Yi, et al. Sparse R-CNN: end-to-end object detection with learnable proposals [EB/OL]. [2022-06-25]. https://ieeexplore.ieee.org/document/9577670.
[19] TONG Zhan, SONG Yibing, WANG Jue, et al. VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training [EB/OL]. [2022-07-02]. https://arxiv.org/abs/2203.12602v3.
[20] FAN Haoqi, XIONG Bo, MANGALAM K, et al. Multiscale vision transformers [EB/OL]. [2022-07-02]. https://ieeexplore.ieee.org/document/9710800.
[21] LIN Ji, GAN Chuang, HAN Song. TSM: temporal shift module for efficient video understanding [EB/OL]. [2022-07-02]. https://ieeexplore.ieee.org/document/9008827.
[22] FEICHTENHOFER C, FAN Haoqi, MALIK J, et al. Slowfast networks for video recognition [EB/OL]. [2022-06-19]. https://ieeexplore.ieee.org/document/9008780/.
[23] LIU Ze, LIN Yutong, CAO Yue, et al. Swin Transformer: hierarchical vision transformer using shifted windows [EB/OL]. [2022-07-09]. https://ieeexplore.ieee.org/document/9710580.
[24] NI Bolin, PENG Houwen, CHEN Minghao, et al. Expanding language-image pretrained models for general video recognition [EB/OL]. [2022-07-09]. https://arxiv.org/abs/2208.02816.
[25] RADFORD A, KIM J W, HALLACY C, et al. Learning transferable visual models from natural language supervision [EB/OL]. [2022-07-09]. https://arxiv.org/abs/2103.00020.
[26] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16x16 words: transformers for image recognition at scale [EB/OL]. [2022-07-09]. https://arxiv.org/abs/2010.11929v1.
[27] FAN Haoqi, XIONG Bo, MANGALAM K, et al. Multiscale vision transformers [EB/OL]. [2022-07-19]. https://arxiv.org/abs/2104.11227v1.
[28] WU Chaoyuan, LI Yanghao, MANGALAM K, et al. MeMViT: memory-augmented multiscale vision transformer for efficient long-term video recognition [EB/OL]. [2022-07-19]. https://arxiv.org/abs/2201.08383v2.
[29] ZHANG Hao, HAO Yanbin, NGO C W. Token shift transformer for video classification [EB/OL]. [2022-07-19]. https://arxiv.org/abs/2108.02432v1.
[30] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [EB/OL]. [2022-07-17]. https://arxiv.org/abs/2207.02696.
[31] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, et al. Deep residual learning for image recognition [EB/OL]. [2022-07-17]. https://ieeexplore.ieee.org/document/7780459.

The Efficient Multi-Person Action Detection on Mobile Devices

JI Honglei[1], DING Han[2,3], ZHAO Chaoyang[2], TANG Ming[2], WANG Jinqiao[2]
(1. CRRC Academy, Beijing 100071, China; 2. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; 3. School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Video action detection is a promising yet challenging task. However, most existing methods are computationally expensive. An action detection method based on consecutive multi-frame cache is presented. For multi-person scenarios, action classification can still be handled efficiently in combination with person detector based on single frame. Temporal shift module is introduced to cache the features of previous frames so that the network is endowed with the ability to process temporal information. Experiments show that the framework achieves fantastic light-weight effects. It proves the possibility to perform real-time action detection on multi-person scenarios with the help of person detector and shows advantages in both speed and accuracy.

Key words: multi-personaction detection; light-weight; mobile device oriented

(责任编辑:冉小晓)