基于姿态与双流神经架构搜索的行人动作识别

龚申健¹, 张姗姗^{1*}, 郭煜¹, 杨健¹, 陶冶²

1. 南京理工大学计算机科学与工程学院, 高维信息智能感知与系统教育部重点实验室, 江苏省社会安全图像与视频理解重点实验室, 南京 210094

2. 地平线, 北京 100085

* 通信作者. E-mail: shanshan.zhang@njust.edu.cn

收稿日期: 2021-06-05; 修回日期: 2021-12-21; 接受日期: 2022-02-02; 网络出版日期: 2023-03-14

国家自然科学基金国际(地区)合作与交流项目(批准号: 61861136011)、国家自然科学基金(批准号: 62172225)和中央高校基本科研基金(批准号: 30920032201)资助

摘要 行人是城市交通场景下的弱势群体, 为了避免碰撞, 有必要准确地预测他们的动作行为. 为此, 本文首次提出城市交通场景中行人动作识别这一问题并提出了有针对性的解决方案. 首先, 我们创建了一个新的行人动作识别数据集 (PARD) 作为实验的数据基础, 并给出了一个有效的基准模型 MFVGG, 该模型能够以较低的计算成本达到与之前先进人体动作识别方法相当的性能. 为了更针对性地解决问题, 本文在两个方面对基准模型进行了改进. 首先, 利用姿态先验来丰富特征表示, 构造双流网络融合双分支编码特征. 其次, 本文引入双流神经架构搜索得到对于这项任务的最优层级网络架构. 实验表明, 提出的方法的性能超过了一般人体动作识别相关的先进算法. 数据集以及代码公布在 <https://github.com/Yankeegsj/PARD>.

关键词 深度学习, 计算机视觉, 动作识别, 网络架构搜索, 姿态估计

1 引言

行人是城市交通场景中最重要参与者之一, 例如在马路边的站台等车的人, 在斑马线上行走的人, 或是坐在马路边休息的人等. 近年来, 研究者们针对行人检测^[1~5]、姿态估计^[6] 这些行人分析的基础任务进行了大量的研究. 分析行人的目的在于避免行人与智能交通工具的碰撞, 因此我们需要预测每个行人的轨迹. 当两个处于同一位置的行人的动作模式不一样时, 会导致其未来轨迹辐射范围不一致, 智能车的路径规划需要避免与行人未来轨迹辐射区域出现交集. 从这个意义上说, 仅仅知道每个行人的当前位置是不够的, 未来的轨迹还取决于当前的行动. 因此, 行人动作识别对自动驾驶的决策具有重要意义. 它可以判断观察到的行人是在行走、站着还是骑行, 从而更准确地预测行人未来轨迹. 可是, 目前还没有针对城市交通场景中行人行为识别问题的研究.

引用格式: 龚申健, 张姗姗, 郭煜, 等. 基于姿态与双流神经架构搜索的行人动作识别. 中国科学: 信息科学, 2023, 53: 485–499, doi: 10.1360/SSI-2021-0198
Gong S J, Zhang S S, Guo Y, et al. Pose-guided pedestrian action recognition with two-stream neural architecture searching (in Chinese). Sci Sin Inform, 2023, 53: 485–499, doi: 10.1360/SSI-2021-0198

虽然在过去的十年中, 一般人体动作识别的问题已经被广泛研究, 但我们认为, 行人动作识别是一项更为困难的任务, 值得进行具体的研究. 与一般人体动作识别相比, 行人动作识别任务在以下几方面更具挑战性: (1) 背景是动态变化的. 路径规划的主体是被控制的智能车, 它需要根据车载摄像机的信息分析行人行为并作出相关决策. 车载摄像机随着智能车移动, 使图像背景出现巨大偏差. 此外, 城市交通场景由大量不同种类的物体组成, 如车辆、建筑物、植物等. (2) 不同运动模式的行人在画面表现中比较相似. 例如, 站立和缓慢行走的人从形态上来看非常相似, 尤其是从前面或后面观察时. (3) 行人运动类别相对有限, 但不同类别之间的数据极度不均衡. 行走着的行人占大多数, 而其他动作却不常见.

我们提出了行人动作识别这一问题, 创建了一个新的数据集并提出了一个有针对性的解决方案. 本文的主要贡献总结如下:

(1) 我们对比了行人动作识别和一般人体动作识别问题的具体区别. 据我们所知, 这是第 1 次提出针对城市交通场景中行人的动作识别. 同时, 为了支持这一领域的研究, 我们提出了第 1 个行人动作识别数据集 (pedestrian action recognition dataset, PARD) 作为行人动作识别的数据基础. PARD 由 4 个不同动作的约 4000 个视频片段组成.

(2) 我们提出了一种简单的基准模型 MFVGG (multi-frame VGG), 该方法以较低的计算成本获得了与目前常用的先进人体动作识别方法相当的识别精度.

(3) 本文从以下两个方面进一步改进基准模型: 首先, 本文研究了如何利用人体姿态估计作为先验知识, 在丰富特征表示的同时使得模型在动作识别上具有更强的分辨力; 其次, 提出了一种双流神经架构搜索方法来寻找适合行人动作识别任务的最优网络架构. 基于 MFVGG, 定义一个双流层级搜索空间, 以找到 RGB 分支与姿态热图分支的最佳特征编码器.

2 相关工作

由于行人动作识别问题与一般人体动作识别密切相关, 本文先回顾最近的一般人体动作识别相关的工作, 其中也包括对现有数据集的简要介绍. 此外, 还回顾了最近在神经网络架构搜索领域的工作.

2.1 一般人体动作识别

由于一般人体动作识别的广泛应用, 近年来该任务受到越来越多的关注. 同时, 由于卷积神经网络 (convolutional neural networks, CNNs) 的巨大成功, 动作识别的性能得到了显著的提升.

在一般人体动作识别任务中, 研究者认为时间信息是非常有用的, 因此利用视频帧代替单一图像成为一种典型的方法. 另外一种对于时间信息的表示是光流. 为了利用光流, Simonyan 等^[7] 设计了一个双流网络, 并证明了它对动作识别是有效的. STM^[8], TDN^[9] 不选择光流, 而是采用图像差分来建模时序和运动. 另外, 为了更全面的时序建模, TSM^[10], TEA^[11] 对时序进行单独建模然后将其应用到 2D CNNs 中. 然而, 相机的快速运动导致光流和图像差分不能反映画面中人的运动规则, 因此不适合用于车载动态视角下的行人动作识别. 另外, 还有将 2D CNNs 膨胀成 3D CNNs 的方法, 通过时序通道上的卷积以隐式的方式建模时序关联^[12,13]. 但是, 3D CNNs 的一个明显缺点是它比 2D CNNs 包含更多的参数, 它们更容易对相对小规模的数据集过拟合^[14]. Qiu 等^[15] 通过把 3D CNNs 解耦成 2D 空间卷积和 1D 时间卷积来减少 3D CNNs 的计算成本. 为了利用视频帧的优势, 同时使分类器模型在复杂的道路场景下具有鲁棒性, 我们沿通道维度按时序连接 RGB 帧构成输入.

除了时间信息,人体关节也是动作识别的有用线索,它们代表着人体的结构. Du 等^[16]使用 Microsoft Kinect 提供的人体关节,根据人体的物理结构将人体骨骼关节划分为 5 个部分,然后利用层次递归神经网络对时间信息进行编码. 可是, Microsoft Kinect 设备只能在室内场景使用. Huang 等^[17]建议利用李群表示人体骨骼,但这样做导致结果对人体关节的准确性非常敏感. Baradel 等^[18]提出基于姿态的时空注意力模型来指导模型专注于手部区域,但他们依赖的 RGB-D 数据在城市交通场景中很难获得. Chéron 等^[19]提出了基于姿态的 CNN 特征,并表明姿态先验有利于人体动作识别. 他们是通过在人体关节的相应位置周围裁剪 RGB 图像块实现的,但本文选择采用姿态热图作为姿态先验. 姿态热图通过在相应点上应用高斯响应来表示人体关节,对噪声关键点检测具有更强的鲁棒性.

2.2 人体动作识别数据集

为了促进人体动作识别的研究,研究者提出了许多动作识别数据集. 其中, Kinetics 数据集^[20]包含 400 个动作类别,并专注于人类的动作,例如绘画、喝水、大笑、拥抱、洗碗等. 还有一些规模较小的数据集,例如,包含 101 个动作类别的 UCF-101 数据集^[21],包含 51 个类别的 HMDB-51 数据集^[22],以及包含 200 个动作类别的 ActivityNet-200 数据集^[23].

与现有数据集相比,提出的 PARD 数据集在以下 3 个方面显著区别. (1) 一般的人体动作识别数据集中的动作几乎全部都是日常活动或体育锻炼. PARD 专注于城市交通场景中动态视角下的行人. (2) 一般的人体动作识别数据集中的场景与动作相关联,例如篮球场上打篮球或在游泳池里游泳,但 PARD 中的行人动作发生在城市的任意场景中. (3) 一般动作识别数据集通常由静止或缓慢移动的手持摄像机捕获. 在 PARD 中时,摄像机会随智能车一起快速移动,导致摄像机拍摄画面中的背景运动无法忽略.

2.3 神经架构搜索

神经架构搜索 (neural architecture search, NAS) 是自动机器学习 (auto machine learning, AutoML) 的一项衍生应用,旨在解放人力,自动设计性能良好的神经网络结构. 近两年来, NAS 已被应用于许多计算机视觉应用中,如图像分类^[24, 25]、物体检测^[26]和语义分割^[27]. 众多的实验已经证明神经架构搜索在设计合理的搜索空间中能找到优于专家设计的神经网络结构.

NAS 主要由两个部分组成: 优化方法和搜索空间. NAS 的优化方法可分为 3 类: 强化学习 (reinforcement learning, RL)^[25, 28~30]、进化算法 (evolutionary algorithm, EA)^[30~32]和可微控制器^[24, 33]. RL 将选择的网络结构作为行为,把选择的结构在验证集上得到的评估数值作为行为的反馈. EA 通过逐渐改变当前架构序列的某些组件来生成新架构,同时保留性能良好的架构. RL 和 EA 都是在离散的架构空间中引导搜索过程,效率非常低. 相反,可微控制器将搜索空间映射到一个连续的空间中,在连续空间中,可以直接求导,沿优化方向得到更好的结构. 本文选择基于可微分控制器的优化方法,这样可以通过候选架构的评估性能训练控制器,在连续空间中获得最优化的结构.

NAS 的搜索空间可以按照从局部到全局分成多个级别,包括层级^[34]、块级^[35]、单元级^[24, 27, 36]和路径级^[37]. 其中,层级搜索空间更易于定义和优化,并且层级结构的网络复杂度不高,可以节省运算成本. 此外,之前的双流模型很多只是简单地将光流与图像用同样的模型单独训练,最后在评估的时候将预测结果相加. 本文为双流编码器选择层级搜索空间,并将两个编码器融合搜索,寻找适合各自且适应融合网络的特征编码器.

表 1 PARD 数据集的相关统计
Table 1 Statistics of our proposed PARD dataset

Class	Total	Train	Val	Test
Walk	2157	1295	215	647
Stand	759	456	76	227
Ride	830	498	83	249
Sit	118	71	12	35
Sum	3864	2320	386	1158

3 数据集

据我们所知, 目前没有公开的针对行人动作识别的数据集, 因此我们创建了一个新的数据集, 称之为行人动作识别数据集 (PARD). 本文的所有实验都在 PARD 上进行.

PARD 数据集总共包含 3864 个视频序列. 摄像机安装在汽车上, 得到了汽车穿越亚洲大城市时沿途的图像数据. 同时, 为了增加多样性, 我们还从欧洲收集的 MOT 数据集^[38,39] 和美国的 Caltech 数据集^[40] 中选择了一些序列. 因为交通场景下背景非常复杂, 且数据来自不同的国家和城市, 拥有不同的光照条件, 也有的是在不同季节收集的, 因此 PARD 数据集是多种多样的. 数据集中每个序列由包含同一人的一段视频帧组成. 对于每段序列, 我们通过使用扩大的行人边界框从完整图像中裁剪出每一帧中的同一人, 该边界框使目标人可以定位在中心, 并且扩大的边界框使序列还能包含一些图像上下文信息.

3.1 标注与统计

通过观察发现, 行人常见的动作有 4 类: 步行、站立、骑乘和坐着. 每个类别均显示不同的运动模式, 这是预测未来的运动的重要线索. PARD 中每个序列都带有上述 4 个标签之一. 序列中的每一帧都满足这个标签, 对于某些难以判断的帧, 我们要求标注者查看更多相邻帧后确定行人运动模式再进行序列的标注, 这样的动作标注更可靠. 最后, 将完整的数据集按 6:1:3 的比例分为训练、验证和测试子集. 统计信息展示在表 1 中. 另外, 图 1 展示了每个类别的一个示例.

3.2 性能评估

本文在实验中使用两个评价指标.

第 1 个指标是所有序列的总体精度, 这也是本文在实验中的主要评估标准. 它的计算如下:

$$\text{Acc}(f; D) = \frac{1}{m} \sum_{i=1}^m \mathbb{I}(f(x_i) = y_i), \quad (1)$$

其中 D 为视频序列, m 为序列总数, y_i 为序列的标签; $\mathbb{I}$ 是指标函数, 如果条件满足则返回 1, 否则返回 0. 公式被广泛应用于分类任务, 并成为许多动作识别数据集 (如 HMDB-51^[22], UCF-101^[21], ActivityNet-200^[23] 和 Kinetics^[20] 数据集) 的标准评估指标. 在每类数据分布不平衡的情况下, 只有总体的准确性并不能反映出每个类的准确性. 与图像分类相似, 本文采用平均类精度 (mAcc) 作为附加评估指标, 计算如下:

$$\text{mAcc}(f; D) = \frac{1}{n} \sum_{j=1}^n \frac{1}{m_j} \sum_{i=1}^{m_j} \mathbb{I}(f(x_{i,j}) = y_{i,j}), \quad (2)$$

图 1 (网络版彩图) PARD 数据集的一些样本. 每列的 4 帧来自同一序列. 对应序列的标签从左到右分别是 (a) 坐着, (b) 行走, (c) 骑乘和 (d) 站立

Figure 1 (Color online) Some examples from PARD dataset. Each column contains 4 frames of a sequence. The ground-truth labels from left to right are (a) sit, (b) walk, (c) ride, and (d) stand, respectively

其中 n 代表数据集中的类别数量, m_j 为第 j 类的序列数, $x_{i,j}$ 是 j 类的第 i 个样本, $y_{i,j}$ 是该样本的标签. 实际上是对每个类别的准确率取平均值, 表示不同类别的平均精度.

4 方法

本节将具体介绍提出的行人动作识别方法. 在简要概述提出的方法之后, 一步步介绍单分支简单基准以及如何使用人体姿态先验知识, 并在此基础上使用双流神经架构搜索更好地解决问题.

4.1 方法概述

如图 2 所示, 提出的基于姿态的双流神经架构 ($\text{Net}_{\text{TS}} + \text{Net}_{\text{F}}$) 包括 3 部分: 双流编码器 (Net_{TS}), 融合网络 (Net_{F}) 和分类器. 双流编码器是通过双流神经架构搜索到的 RGB 分支和姿态热图分支的最佳层级架构. 融合网络和分类器是预定义的结构. 具体来说, RGB 分支和姿态热图分支分别提

图 2 (网络版彩图) 本文提出的 Net_{TS}+Net_F 模型流程图. 它是一个级联网络, 由 3 个阶段组成. 双流编码器产生丰富的特征, 通过融合网络聚合后经过分类器进行动作识别. 由于姿态热图分支和 RGB 分支特征图的空间尺度不一致, 在对两个分支特征进行通道上的级联时, 使用 crop 操作将姿态热图分支的特征图裁剪成 28×28, 与 RGB 分支保持一致

Figure 2 (Color online) Pipeline of our proposed Net_{TS}+Net_F method. It is a cascaded network consisting of three stages. The two-stream encoder produces enriched features which are then aggregated by a fusion net, followed by a classifier for action recognition. We concatenate the feature maps from two branches along the channel dimension. However, for the pose branch, the spatial scale of the feature maps is 32×32 and for the RGB branch is 28×28. Therefore, we crop the feature maps from the pose branch to 28×28

取输入的 RGB 视频帧和姿态热图的特征, 融合网络合并聚合两分支的编码特征后输入到分类器进行行人动作预测.

本文使用 softmax 损失来训练模型. 对于样本输入数据 x_i , 样本标签 y_i 的样本, 它的损失计算公式如下:

$$L_{\text{softmax}}(x_i, y_i) = - \sum_{j=1}^n (1 - p_j) \mathbb{I}(y_i = j) \log p_j, \quad (3)$$

其中 p_j 表示类 j 的预测概率, n 是类别的数量.

4.2 单分支分类基准

在动作识别领域, 目前先进的方法都是使用重量级模型来实现更高的识别精度^[12~14]. 然而, 自动驾驶时需要在相对较低的计算成本下获得合理的性能. 为此, 本文提出使用 VGG11^[41] 作为骨干网络的简单基准. VGG 网络已广泛应用于计算机视觉任务, 比如图像分类和目标检测. 为了让它适应视频分析任务, 我们将视频帧按时序在通道维度上级联, 得到对应维度的输入. 同时为了快速收敛, 我们还在每个卷积层之后应用批量归一化层^[42]. 因为该 VGG 基准适用于多帧输入, 所以将其称为多帧 VGG (MFVGG).

4.3 姿态先验

人体姿态由某些具体关节表示, 人体关节代表了身体骨架, 而骨架的姿态在不同的动作过程中是不同的. 因此, 我们认为人体姿态是动作识别的重要线索. 然而, 姿态估计的结果有时会产生噪

表 2 预定义的融合网络和分类器. 它们接收来自 RGB 分支和姿态热图分支的编码特征进行融合. 这部分结构在 MFVGG_{pose} 模型以及 Net_{TS} + Net_F 模型中使用

Table 2 Our pre-defined simple fusion net: Net_F and the classifier that follows it. The fusion net is used while generating MFVGG_{pose} and Net_{TS} + Net_F

Operation	Concat	Conv3x3 s1	Conv3x3 s1	Maxpool2x2 s2	Conv3x3 s1	Maxpool2x2 s2	Avgpool7x7 s1	Classifier
Out	1024	512	512	512	512	512	512	4

表 3 MFVGG_{pose} 和 Net_{TS} 分支对比. 左边两列是 MFVGG_{pose} 的两个分支, 右边两列是搜索到的 Net_{TS} 的两个分支. 其中所有的操作都已经在表 4 中声明. s1/s2 指的是该操作的步长为 1 或者 2. p0 是指对应操作时填补设置为 0. Out_C 是对应层的输出特征图的个数

Table 3 The two branches of MFVGG_{pose} (the left two columns) and the searched results of Net_{TS} (the right two columns). All operations are illustrated in Table 4. s1/s2 indicates that the stride of this operation is 1/2. p0 indicates that the padding of this operation is 0. Out_C is the number of output feature maps

B_{RGB}	Out_C	B_{Pose}	Out_C	B_{RGB}	Out_C	B_{Pose}	Out_C
Conv3x3 s1	64	Conv3x3 s1	128	Sepconv5x5 s1	64	Conv3x3 s1	64
Conv3x3 s1	128	Maxpool2x2 s2	128	Sepconv3x3 s2	64	Sepconv5x5 s1(Conv1x1)	128
Conv3x3 s1	128	Conv3x3 s1	256	Sepconv3x3 s1(Conv1x1)	128	Conv5x5 s1(Conv1x1)	256
Maxpool2x2 s2	128	Conv3x3 s1	512	Conv3x3 s1	256	Avgpool2x2 s2	256
Conv3x3 s1	256	Conv3x3 s1 p0	512	Sepconv5x5 s1(Conv1x1)	256	Conv3x3 s1(Conv1x1)	256
Conv3x3 s1	256	Conv3x3 s1 p0	512	Conv5x5 s1	512	Sepconv3x3 s1	512
Maxpool2x2 s2	256	—	—	Maxpool2x2 s2	512	Crop	512
Conv3x3 s1	512	—	—	—	—	—	—

表 4 双流神经架构搜索的搜索空间中的候选操作集合. 括号中的操作与括号外的操作平行, 双分支分别处理后将其结果相加

Table 4 A set of candidate operations of our two-stream encoder search space. Sepconv3x3 (Conv1x1) means that the operation Sepconv3x3 is paralleled with Conv1x1

Operation	Avgpool	Maxpool	Conv	Conv	Conv3x3	Conv5x5	Sepconv	Sepconv	Sepconv3x3	Sepconv5x5
	2x2	2x2	3x3	5x5	(Conv1x1)	(Conv1x1)	3x3	5x5	(Conv1x1)	(Conv1x1)
Stride	2	2	1/2	1	1/2	1	1/2	1	1/2	1

声, 特别是在复杂的室外场景中. 另一方面, 有些动作表现出相当相似的骨架, 如走路和站立. 这使得很难区分不同的动作和姿势. 因此, 本文提出使用姿态作为辅助信息, 配合 RGB 视频帧进行行人动作识别.

本文使用的姿态热图是使用在 MPII 数据集^[43]上训练的 Hourglass 模型^[44]预测的, 其空间分辨率统一为 64×64. 对于每一帧, 有 16 个姿态热图, 每个代表一个身体关节的响应.

具体来说, 本文使用两个结构与 VGG 一致的分支, 分别从 RGB 视频帧和姿态热图中提取特征, 然后应用融合网络将两个分支中的特征进行融合. 将使用两个预定义卷积网络作为两个分支的完整模型称为 MFVGG_{pose}. 模型的具体组成见表 2 和 3.

4.4 双流神经架构搜索

我们在 MFVGG 中使用 VGG11 作为骨干网络, 但是尚不清楚它是否是此任务的最佳网络结构. 另一方面, 当使用姿势作为辅助信息时, 选择一个合理的结构同时兼顾这两个分支就更加困难了. 为

了解决这个问题, 本文采用了神经架构搜索技术来搜索适合于行人动作识别任务的最佳网络架构。

在 4.3 小节, 我们已经手动定义了融合网络。同时, 根据下文的消融实验可知, MFVGG 和 MFVGG_{pose} 能够达到较高的预测水平, 说明了层次分支的有效性。因此我们选择将其作为基本体系结构并执行双流层级搜索, 以获得双流编码器的最佳架构。

双流层级搜索空间中每一层有 14 种候选操作, 如表 4 所示。其中 ‘Sepconv’ 是指将标准卷积分为一个通道分离卷积层和一个卷积核大小为 1 的标准卷积。14 种候选操作中, 步长为 1 的有 8 个候选操作, 步长为 2 的有 6 个候选操作。每个分支有 $N(K^1 + K^2)$ 层, 包括步长为 1 的 K^1 层和步长为 2 的 K^2 层。在提出的模型中, RGB 分支的层数设置为 $K_{\text{RGB}}^1 = 5, K_{\text{RGB}}^2 = 2$ 。同样, 姿态热图分支的层数设置为 $K_{\text{Pose}}^1 = 5, K_{\text{Pose}}^2 = 1$ 。

在搜索空间中, 我们定义每层输出通道的数量, 仅搜索每层的操作。对于双流层级搜索空间, 先确定步长为 2 的两层。此时, 对于 RGB 分支而言, 有 C_7^2 种可能性。之后为每个层独立选择操作, 并将它们连接起来以形成一个完整的分支。经过计算, RGB 分支有 $6^2 \times 8^5 \times C_7^2$ 种层级架构类型。同样, 提出的姿态热图分支有 $6^1 \times 8^5 \times C_6^1$ 的可能性。综上所述, 在双流层级神经架构搜索的搜索空间中, 总共有 $6^2 \times 8^5 \times C_7^2 \times 6^1 \times 8^5 \times C_6^1 \approx 3 \times 10^{13}$ 种可能的结构。

以下, 我们将逐步描述整个搜索过程。先初始化一个候选架构集合 $A = \{A_{\text{RGB}}, A_{\text{Pose}}\}$, 通过权重共享策略^[45] 训练整个搜索空间。具体来说, 每次迭代时, 从 A 中随机选择一种架构序列, 并生成相应的两个分支 $\{a_{\text{RGB}}, a_{\text{Pose}}\}$ 。然后把两个分支和预定义的 Net_F 连接后组成整个网络, 这样就可以通过优化公式 (3) 进行端到端地训练。在完整训练集上训练了若干个周期后, 评估所有 A 中的候选架构, 并获得其对应评估度量作为反馈, 再通过候选架构及其对应性能训练控制器。控制器的结构如图 3 所示。控制器将架构编码为隐藏状态, 将隐藏状态输入多层感知机 (multilayer perceptron, MLP) 区分性能。当控制器最优时, 它能了解架构与其性能之间的关系。我们需要更高性能对应的架构, 只需要对隐藏状态求性能增加的方向梯度, 沿着该梯度方向更新隐藏状态, 解码器将更新后的隐藏状态重新解码出结构。实际上, 这可以视为当前集合中的架构在隐藏空间整体向更高性能架构的推进。这样, 控制器生成一些更好的架构 A_1 。我们选择 A 中的前 K 个架构作为 A_2 , 并生成一些随机架构 A_3 。将 A_1, A_2, A_3 合并来更新 A 。以这样的流程, 重复地对候选架构和控制器进行训练, 最终获得最佳架构。双流神经架构搜索的完整过程如算法 1 所示。

搜索到的最佳分支称为 $\text{Net}_{\text{TS}} = \{B_{\text{RGB}}, B_{\text{Pose}}\}$, 其具体信息见表 3。

5 实验

本节先描述方法实现细节, 然后在 PARD 测试集上与其他最新方法进行比较, 最后, 针对提出的姿态先验和双流神经架构搜索的影响进行消融实验。

5.1 实验细节

对于每个视频序列, 我们从中随机选择一个起始帧, 取 4 帧作为输入, 将所有图像的大小调整为 112×112 , 同时通过填充上下文保持其原始宽高比。

原始的 VGG11 应用于尺寸为 224 的图像, 为使尺寸匹配, 只需删除原始 VGG11 模型的一个最大池化。此外, 本文采用了更轻量级的分类器, 仅由全局平均池和单个全连接层组成。MFVGG 和 MFVGG_{pose} 总共训练了 300 个周期, 初始学习率设置为 1×10^{-3} , 每 40 个周期衰减为原先的 1/2, 使用 Adam 优化器^[46] 进行优化, 批大小设置为 12。

Algorithm 1 Pseudo-code of two-stream NAS

```

1: Define the number of search epochs  $N_{\text{epoch}}$ ;
2: Define the set of evaluation epochs  $N_{\text{eval}}$ ;
3: Define the gradient-based controller  $C$ ;
4: Define the number of iterations in one epoch  $N_{\text{iter}}$ ;
5: Initialize candidate architectures  $A = \{A_{\text{RGB}}, A_{\text{Pose}}\}$ ;
6: Initialize the pre-defined fusion net  $\text{Net}_F$  as Table 2;
7: for epoch  $\in [1, N_{\text{epoch}}]$  do
8:   for iter  $\in [1, N_{\text{iter}}]$  do
9:      $a \leftarrow \text{random pick}(A)$ ;
10:     $\{a_{\text{RGB}}, a_{\text{Pose}}\} \leftarrow a$ ;
11:    train( $\{a_{\text{RGB}} + a_{\text{Pose}}\} + \text{Net}_F$ );
12:   end for
13:   if epoch  $\in N_{\text{eval}}$  then
14:     reward  $\leftarrow \text{eval}(A)$ ; # Acc in validation set
15:     train( $C$ );
16:     Generate  $A_1$  by  $C$ ;
17:     Reserve the top K architectures in  $A$  as  $A_2$ ;
18:     Randomly generate some architectures  $A_3$ ;
19:     update( $A$ ); # Combine  $A_1$ ,  $A_2$  and  $A_3$ 
20:   end if
21: end for
22: top1  $\leftarrow \text{pick}(A, \text{reward})$ ;
23:  $\{B_{\text{RGB}}, B_{\text{Pose}}\} \leftarrow \text{top1}$ ;
24:  $\text{Net}_{\text{TS}} = \{B_{\text{RGB}}, B_{\text{Pose}}\}$ ;
25: Return  $\text{Net}_{\text{TS}}$ .

```

图 3 (网络版彩图) 控制器的编码解码结构, $L_1, L_2, \dots, L_N$ 是层级网络架构的序列表示方法. 控制器中编码部分与解码部分分别有一个 LSTM 单元. 编码器将架构编码到隐藏状态空间, 然后通过 MLP 拟合隐藏状态空间与结构的评估反馈. 解码结构通过隐藏状态重构出原先的结构序列

Figure 3 (Color online) The encoder-decoder structure of the controller. $L_1, L_2, \dots, L_N$ are the serialized representations of one architecture in our search space. In the controller, there are two LSTM cells, one of which is shared in the encoder and the other is shared in the decoder. The encoder is employed to observe a hidden state of the architecture. Then we fit the hidden state to its reward by MLP. And the decoder decodes the corresponding structure from the hidden state and the pre-structure of the architecture

本文选择 NAO^[24] 作为搜索 Net_{TS} 时的优化方法, 同时为了加速神经架构搜索的执行过程, 采用权重共享策略^[45]. 初始化 400 个候选架构, 在整个训练集上搜索 200 个周期, 并每隔 50 个周期评估候选架构集合 A . 控制器采用批大小为 20 的 Adam 优化器进行优化, 并且在对所有候选架构进行每

表 5 本文方法与当前先进算法在 PARD 测试集上比较的结果 ^{a)}
Table 5 Comparison to current state-of-the-art methods on the PARD test set^{a)}

Method	Input	Params (M)	FLOPs (G)	Runtime (s)	Acc (%)	mAcc (%)	Sit (%)	Walk (%)	Ride (%)	Stand (%)
3D Resnet-34 ^[14]	RGB	63.5	705	0.005	81.9	70.7	45.7	89.2	90.4	57.3
3D Resnet-101 ^[13]	RGB	82.3	1377	0.014	81.8	70.7	45.7	89.0	88.0	59.9
P-3D ^[15]	RGB	24.9	4	0.008	82.9	74.1	54.3	88.4	90.4	63.4
I3D ^[12]	RGB	12.3	7	0.010	84.0	74.4	48.6	88.4	91.2	69.2
LSTA ^[47]	RGB	40.8	5	0.023	83.6	74.8	60.0	91.3	92.8	55.1
MFVGG	RGB	9.2	8	0.001	84.9	77.3	57.1	88.3	90.8	73.1
P-3D ^[15]	RGB+Flow	49.8	8	0.014	87.0	76.6	51.4	92.9	93.6	68.3
I3D ^[12]	RGB+Flow	24.6	14	0.018	86.6	75.4	45.6	92.0	92.4	71.4
Net _{TS} + Net _F	RGB+Pose	10.7	19	0.003	88.1	81.2	65.7	92.6	92.4	74.0

a) Bold indicates the best results. The runtimes are tested on a Nvidia 2080Ti GPU with a batch size of 24.

次评估之后, 在所有评估架构上训练 10 个周期.

在找到最优的双分支结构后, 我们采用与 MFVGG 相同的训练策略来训练搜索到的模型.

本文中所有实验均在一块具有 11 GB 显存的 Nvidia 2080Ti GPU 上进行. 训练一个网络需要 24 h, 神经架构搜索过程大约需要 35 h.

5.2 与先进方法的比较

我们将提出的方法与其他先进的人体动作识别方法 (3D-Resnet^[13], P-3D^[15], I3D^[12] 和 LSTA^[47]) 进行了比较, 所有实验都是在 PARD 数据集上进行的. 对于其他最先进方法的实验, 我们使用它们的开源代码且所有模型的训练策略与本文的基准一致.

根据表 5 中的结果, 有以下结论: (1) 当仅使用 RGB 图像作为输入时, MFVGG 基准已经能取得与之前的人体动作识别方法相当的结果, 同时计算成本较低且易于实现. 与 LSTA 和 3D Resnet-101 相比, MFVGG 的运行速度快约 10 倍. (2) Net_{TS}+Net_F 与 MFVGG 基准相比, 在类平均精度上提高了 3.9 个百分点, 表明姿态先验和双流神经架构搜索对于此任务是有效的. (3) 通过使用姿态先验信息, Net_{TS}+Net_F 在整体精度上达到了 88.1%, 优于其他人体动作识别的先进方法. 与以前的最佳方法 P-3D (RGB+Flow) 相比, 准确率提升 1.1%. 这些结果证明, 姿态信息在行人动作识别上的效果比光流更好. (4) Net_{TS}+Net_F 在不同类别上实现了更加均衡的结果, 达到了 81.2% 的平均准确度, 比 P-3D (RGB+Flow) 高 4.6%. (5) Net_{TS}+Net_F 以更高的运算效率实现了更好的性能. 它的运行速度比 I3D 和 P-3D 快 5~6 倍. 因为 I3D 和 P-3D 都是偏复杂的 3D CNNs 模型, 由于显存的访问时间随着模型的层数增加而增加, 它们即使运算量较低, 运行速度依旧会慢.

另外, 在图 4 中, 我们选择了一些困难样本并展示了 Net_{TS}+Net_F 以及一些其他模型的定性结果. 观察发现, 有了姿态先验, 本文的方法更关注行人, 对于背景干扰更鲁棒. 比如, 在第 1 行, 由于背景变化较大, P-3D 和 I3D 将‘行走’误判为‘骑行’. 在最后一行, P-3D, I3D 和 MFVGG_{pose} 将‘坐着’误判为‘骑行’, 这应该是在观察到的行人旁出现了一辆自行车而引起的. 在使用搜索到的与融合方法匹配的双流编码器之后, Net_{TS}+Net_F 能正确判断行人的运动模式.

Video clips	Ground truth		$\text{Net}_{\text{TS}} + \text{Net}_{\text{F}}$		$\text{MFVGG}_{\text{pose}}$		P-3D		I3D
Sit		4.26E-04		1.21E-04		8.04E-10		5.52E-09
	Walk		8.56E-01		9.94E-01		3.72E-03		2.56E-06
	Ride		1.15E-01		1.22E-03		9.96E-01		1.00E+00
	Stand		2.90E-02		5.06E-03		4.18E-07		8.45E-08
Sit		1.73E-10		1.36E-09		6.02E-11		8.39E-09
	Walk		1.59E-01		8.64E-01		1.00E+00		7.97E-01
	Ride		3.41E-10		3.25E-08		2.23E-13		7.81E-12
	Stand		8.41E-01		1.36E-01		4.88E-04		2.03E-01
Sit		6.06E-04		4.53E-02		1.11E-05		6.96E-09
	Walk		1.49E-02		2.56E-01		9.76E-01		8.16E-05
	Ride		9.78E-01		5.96E-01		9.98E-03		6.14E-05
	Stand		6.03E-03		1.02E-01		1.37E-02		1.00E+00
Sit		6.17E-01		8.96E-05		9.94E-10		1.64E-04
	Walk		3.52E-05		2.05E-04		2.31E-11		1.08E-07
	Ride		3.78E-01		1.00E+00		1.00E+00		1.00E+00
	Stand		5.66E-03		4.55E-05		2.49E-10		8.86E-09

图 4 (网络版彩图) 本文 $\text{MFVGG}_{\text{pose}}$ 和 $\text{Net}_{\text{TS}} + \text{Net}_{\text{F}}$ 方法与一些现有方法在 PARD 测试集上的定性结果. ‘Ground truth’ 列中加粗的部分表示该序列的标签. 数字是每个类的预测概率. 粗体表示每个视频序列的最高概率

Figure 4 (Color online) Qualitative results of our $\text{MFVGG}_{\text{pose}}$ and $\text{Net}_{\text{TS}} + \text{Net}_{\text{F}}$ and some previous state-of-the-art methods on the PARD test set. Bold in the ‘Ground truth’ column indicates the ground-truth labels. The numbers are predicted probabilities for each class. Bold highlights the highest probability for each video sequence

表 6 姿态信息和双流神经架构搜索对模型性能的影响

Table 6 Ablation study on the impact of pose prior and two-stage NAS

Method	Params (M)	FLOPs (G)	Runtime (s)	Acc (%)	ΔAcc (%)	mAcc (%)	ΔmAcc (%)
MFVGG	9.2	8	0.001	84.9	—	77.3	—
MFVGG+Pose	13.8	15	0.002	86.7	+ 1.8	78.9	+ 1.6
MFVGG+Pose+ Net_{TS}	10.7	19	0.003	88.1	+ 3.2	81.2	+ 3.9

5.3 消融实验

在本小节, 我们将分析 $\text{Net}_{\text{TS}} + \text{Net}_{\text{F}}$ 中姿态先验和双流神经架构搜索对行人动作识别的影响. 为了体现它们各自的效果, 我们将其逐步添加到 MFVGG 上并进行了相关实验. 这些模型在 PARD 的训练集和验证集上训练, 在测试集上测试. 从表 6 中的结果可以看出, 使用姿态先验后, 模型在总精度和类平均精度上都有显著提升. 双流神经架构搜索算法搜索到的 Net_{TS} 总共有大于 3 个百分点的提升, 但这也依旧没有大幅增加计算成本, 这表明相比于手工设计的双流模型, 搜索的双流网络结构能更高效地从 RGB 图像和姿态热图中提取显著特征用于行人运动模式识别.

另外, 为了进一步分析姿态先验对模型性能的影响, 在表 7 中, 我们比较了 MFVGG、使用 RGB 图像和光流信息的 MFVGG, 以及使用不同姿态估计方法的 $\text{MFVGG}_{\text{pose}}$ 的性能. 这些模型在 PARD

表 7 姿态先验对模型性能的影响

Table 7 Impact of pose prior to the model using different input information or pose estimation methods

Method	Input	Acc (%)	Δ Acc (%)	mAcc (%)	Δ mAcc (%)
MFVGG	RGB	84.9	—	77.3	—
MFVGG _{flow}	RGB+Flow	86.0	+ 1.1	79.1	+ 1.8
MFVGG _{pose}	RGB+Pose (Resnet) [48]	85.8	+ 0.9	79.8	+ 2.5
MFVGG _{pose}	RGB+Pose (Hrnet) [49]	86.7	+ 1.8	78.6	+ 1.3
MFVGG _{pose}	RGB+Pose (Hourglass) [44]	86.7	+ 1.8	78.9	+ 1.6

的训练集和验证集上训练, 在测试集上测试. 从表 7 中的结果可以看出, 相比于光流, 使用姿态估计作为先验在识别精度上通常会更优. 另外, 无论采用何种姿态估计预训练模型得到姿态先验, 总精度和类平均精度上都有提升, 这说明姿态先验估计有利于行人的动作识别, 但对行人动作识别的提升程度不同. 本文最终采用在 MPII 数据集 [43] 上预训练的 Hourglass [44] 模型, 因为其对行人动作识别表现更优.

为了进一步阐明 NAS 的稳定性和随机性的影响, 我们进行了 3 次搜索, 得到的结构搜索结果在测试集上的平均精度为 $88.1\% \pm 0.26\%$ (mean $\pm$ var), 另外, 我们也分析了数据集划分对 NAS 的影响, 将原训练集与验证集混合并随机打乱样本顺序后, 再按照原本两者的数量关系重新分配样本, 得到的结构搜索结果最终在测试集上的精度 (Acc) 为 $88.2\% \pm 0.11\%$ (mean $\pm$ var), 两者较小的方差共同说明了 NAS 搜索结果的稳定性且表明本文的双流层级 NAS 搜索到的结构比人为定义的 MFVGG_{pose} 更适合行人动作识别.

6 总结

本文提出了城市交通场景中的行人动作识别问题并给出针对性的解决方案. 与一般人体动作识别相比, 由于背景复杂、类间差异小等原因, 该任务变得具有挑战性.

我们先创建了一个新的数据集, 即 PARD, 包含 3864 个交通场景中的不同行人动作的视频序列. 然后, 给出了一个能够以较低的计算成本运行且易于实现的基准模型 (MFVGG) 并获得与之前人体动作识别先进方法相当的结果. 为了进一步有针对性地解决问题, 本文引入姿态先验, 使模型在复杂背景中更关注行人, 且搜索到的双流层级架构更适合行人动作识别任务.

据我们所知, PARD 是第一个行人动作识别的数据集, 有望激发相关主题的更多研究. 同时, Net_{TS}+Net_F 方法对行人运动模式的预测精度超过了其他人体动作识别方法, 获得了目前最先进的结果.

参考文献

- 1 Zhang S S, Benenson R, Omran M, et al. How far are we from solving pedestrian detection? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 1259–1267
- 2 Yang P Y, Zhang G F, Wang L, et al. A part-aware multi-scale fully convolutional network for pedestrian detection. IEEE Trans Intell Transp Syst, 2021, 22: 1125–1137
- 3 Wang H, Li Y L, Wang S J. Fast pedestrian detection with attention-enhanced multi-scale RPN and soft-cascaded decision trees. IEEE Trans Intell Transp Syst, 2020, 21: 5086–5093
- 4 Cao J L, Pang Y W, Han J G, et al. Taking a look at small-scale pedestrians and occluded pedestrians. IEEE Trans Image Process, 2020, 29: 3143–3152

- 5 Brazil G, Liu X M. Pedestrian detection with autoregressive network phases. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 7231–7240
- 6 Hattori H, Lee N, Boddeti V N, et al. Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance. *Int J Comput Vis*, 2018, 126: 1027–1044
- 7 Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. 2014. ArXiv:1406.2199
- 8 Jiang B, Wang M M, Gan W, et al. STM: spatiotemporal and motion encoding for action recognition. In: Proceedings of IEEE International Conference on Computer Vision, Long Beach, 2019. 2000–2009
- 9 Wang L M, Tong Z, Ji B, et al. TDN: temporal difference networks for efficient action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2021. 1895–1904
- 10 Lin J, Gan C, Han S. TSM: temporal shift module for efficient video understanding. In: Proceedings of IEEE International Conference on Computer Vision, Long Beach, 2019. 7083–7093
- 11 Li Y, Ji B, Shi X T, et al. TEA: temporal excitation and aggregation for action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, 2020. 909–918
- 12 Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 6299–6308
- 13 Hara K, Kataoka H, Satoh Y. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet? In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 6546–6555
- 14 Hara K, Kataoka H, Satoh Y. Learning spatio-temporal features with 3D residual networks for action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, 2017. 3154–3160
- 15 Qiu Z F, Yao T, Mei T. Learning spatio-temporal representation with pseudo-3D residual networks. In: Proceedings of International Conference on Computer Vision, Venice, 2017. 5533–5541
- 16 Du Y, Wang W, Wang L. Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, 2015. 1110–1118
- 17 Huang Z W, Wan C D, Probst T, et al. Deep learning on lie groups for skeleton-based action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017. 6099–6108
- 18 Baradel F, Wolf C, Mille J. Human action recognition: pose-based attention draws focus to hands. In: Proceedings of IEEE International Conference on Computer Vision Workshops, Venice, 2017. 604–613
- 19 Chéron G, Laptev I, Schmid C. P-CNN: pose-based CNN features for action recognition. In: Proceedings of IEEE International Conference on Computer Vision, Santiago, 2015. 3218–3226
- 20 Kay W, Carreira J, Simonyan K, et al. The kinetics human action video dataset. 2017. ArXiv:1705.06950
- 21 Soomro K, Zamir A R, Shah M. UCF101: a dataset of 101 human actions classes from videos in the wild. 2012. ArXiv:1212.0402
- 22 Kuehne H, Jhuang H, Garrote E, et al. HMDB: a large video database for human motion recognition. In: Proceedings of International Conference on Computer Vision, Barcelona, 2011. 2556–2563
- 23 Heilbron F C, Escorcia V, Ghanem B, et al. ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, 2015. 961–970
- 24 Luo R Q, Tian F, Qin T, et al. Neural architecture optimization. 2018. ArXiv:1808.07233
- 25 Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 8697–8710
- 26 Ghiasi G, Lin T Y, Le Q V. Nas-FPN: learning scalable feature pyramid architecture for object detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 7036–7045
- 27 Nekrasov V, Chen H, Shen C H, et al. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 9126–9135
- 28 Gong X Y, Chang S Y, Jiang Y F, et al. Autogan: neural architecture search for generative adversarial networks. In: Proceedings of IEEE International Conference on Computer Vision, Long Beach, 2019. 3224–3234
- 29 Zoph B, Le Q V. Neural architecture search with reinforcement learning. 2016. ArXiv:1611.01578
- 30 Chen Y K, Meng G F, Zhang Q, et al. Renas: reinforced evolutionary neural architecture search. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 4787–4796

- 31 Xie L X, Yuille A. Genetic CNN. In: Proceedings of IEEE International Conference on Computer Vision, Venice, 2017. 1379–1388
- 32 Choe Y, Alippi C, Kozma R, et al. Artificial Intelligence in the Age of Neural Networks and Brain Computing. Orlando: Academic Press, 2019
- 33 Liu H X, Simonyan K, Yang Y M. Darts: differentiable architecture search. 2018. ArXiv:1806.09055
- 34 Wu B C, Dai X L, Zhang P Z, et al. FBNet: hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 10734–10742
- 35 Zhong Z, Yan J J, Wu W, et al. Practical block-wise neural network architecture generation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 2018. 2423–2432
- 36 Dong X Y, Yang Y. Searching for a robust neural architecture in four GPU hours. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 1761–1770
- 37 Cai H, Yang J C, Zhang W N, et al. Path-level network transformation for efficient architecture search. In: Proceedings of International Conference on Machine Learning, Stockholm, 2018. 678–687
- 38 Leal-Taixé L, Milan A, Reid I, et al. Motchallenge 2015: towards a benchmark for multi-target tracking. 2015. ArXiv:1504.01942
- 39 Milan A, Leal-Taixé L, Reid I, et al. MOT16: a benchmark for multi-object tracking. 2016. ArXiv:1603.00831
- 40 Dollar P, Wojek C, Schiele B, et al. Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell, 2012, 34: 743–761
- 41 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. ArXiv:1409.1556
- 42 Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of International Conference on Machine Learning, Lille, 2015. 448–456
- 43 Andriluka M, Pishchulin L, Gehler P, et al. 2D human pose estimation: new benchmark and state of the art analysis. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014. 3686–3693
- 44 Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. In: Proceedings of European Conference on Computer Vision, Amsterdam, 2016. 483–499
- 45 Pham H, Guan M, Zoph B, et al. Efficient neural architecture search via parameters sharing. In: Proceedings of International Conference on Machine Learning, Stockholm, 2018. 4095–4104
- 46 Kingma D P, Ba J. ADAM: a method for stochastic optimization. 2014. ArXiv:1412.6980
- 47 Sudhakaran S, Escalera S, Lanz O. LSTA: long short-term attention for egocentric action recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 9954–9963
- 48 Xiao B, Wu H P, Wei Y C. Simple baselines for human pose estimation and tracking. In: Proceedings of European Conference on Computer Vision, Munich, 2018. 466–481
- 49 Sun K, Xiao B, Liu D, et al. Deep high-resolution representation learning for human pose estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, 2019. 5693–5703

Pose-guided pedestrian action recognition with two-stream neural architecture searching

Shenjian GONG¹, Shanshan ZHANG^{1*}, Yu GUO¹, Jian YANG¹ & Ye TAO²

1. *Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education and Jiangsu Key Laboratory of Image and Video Understanding for Social Safety, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;*

2. *Horizon Robotics, Beijing 100085, China*

* Corresponding author. E-mail: shanshan.zhang@njust.edu.cn

Abstract Pedestrians are vulnerable on streets and their actions serve as important cues for motion prediction to avoid collisions. In this paper, we address the problem of pedestrian action recognition for the first time. We first introduce a new dataset, namely, the pedestrian action recognition dataset (PAR), which serves as a database for experiments. Then, we provide an efficient baseline method, MFVGG, reaching comparable performance to previous methods at lower costs. To better handle the canonical problem, we further improve the baseline from the following two aspects: first, we leverage the pose prior to enrich the feature representations; second, we propose a two-stream neural architecture search (NAS) method to obtain the optimal network architecture tailored to our task. From the experimental results on PAR, our method outperforms previous top-performing action recognition methods. The dataset and code are publicly available at <https://github.com/Yankeegsj/PAR>.

Keywords deep learning, computer vision, action recognition, neural architecture search, pose estimation