小 型 微 型 计 算 机 系 统
Journal of Chinese Computer Systems
DOI:10.20009/j.cnki.21-1106/TP.2021-0826
2023 年 6 月 第 6 期
Vol.44 No.6 2023

一种融合运动特征嵌入的多目标分割跟踪算法

许营坤[1],陈天阳[1],陈胜勇[2],徐新黎[1]

[1](浙江工业大学 计算机科学与技术学院,杭州 310023)
[2](天津理工大学 计算机科学与工程学院,天津 300384)
E-mail:xyk@zjut.edu.cn

摘 要: 针对现有多目标跟踪算法中存在目标运动模糊和相互遮挡的难点,在单阶段和无锚框的实例分割框架下,提出了一种融合运动特征嵌入的多目标分割跟踪算法.首先,提取当前帧与前后两帧光流场中的运动信息对表观特征进行运动补偿,再利用特征金字塔网络融合含有运动信息的多尺度特征,提高了目标检测性能.其次,通过两个用于提升网络预测性能的损失函数的设计和使用,进一步减少了由于检测器失效和目标遮挡而导致的漏检.最后,关联网络提取目标的外观特征,并通过预测并关联的更新轨迹策略将可靠的跟踪结果合并至轨迹.实验结果表明,本文提出的算法在 MOTS20 训练集上跟踪准确度达到了 66.0%,测试集上达到了 63.1%,与同类算法相比,本文算法表现出更好的有效性.

关 键 词: 多目标跟踪;目标检测;目标分割;特征嵌入
中图分类号: TP301 **文献标识码:** A **文 章 编 号:** 1000-1220(2023)06-1304-07

Multi-object Tracking and Segmentation Algorithm by Fusing Motion Feature Embedding

XU Ying-kun[1], CHEN Tian-yang[1], CHEN Sheng-yong[2], XU Xin-li[1]
[1](School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)
[2](School of Computer Science and Engineering, Tianjin University of Technology, Tianjin 300384, China)

Abstract: In order to solve the problems of motion blur and mutual occlusion in existing multi-target tracking algorithms, this paper proposes a multi-object tracking and segmentation algorithm by fusing motion feature embedding under a single-stage and anchor-free detection framework. Firstly, appearance features are compensated using motional cues through estimating the optical flow field between current and neighboring frames, and feature pyramid network is used to fuse multi-scale features containing motion information, which effectively improves performance of detecting objects. Secondly, two loss functions are designed to improve network prediction performance, which further reduce false negative due to detector failure and object occlusion. Finally, association network extracts the appearance characteristics of the targets and the predicted and associated update trajectory strategy merges reliable tracking results into the trajectory. Experimental results show that the tracking accuracy of proposed method reaches 66.0% on MOTS20 training set and 63.1% on test set. Compared with methods using similar frameworks, the proposed method shows better effectiveness.

Key words: multi-object tracking; object detection; object segmentation; feature fusion

1 引 言

得益于图像视频数据的大规模积累和计算机系统的快速发展,深度学习算法在计算机视觉领域中得到了广泛验证.其中多目标跟踪(MOT)作为计算机视觉方向的一项重要技术,在自动驾驶和视频监控等领域具有重要的应用.

在当前主流的多目标跟踪算法中,基于检测的跟踪策略以其较好的可行性得到普遍地采纳和扩展[1],这类算法是通过矩形框去定位并跟踪目标,但在复杂的场景下,一个矩形框内可能包含两个目标,这会引发歧义性,而基于分割的多目标跟踪算法将分类、检测、分割以及跟踪视为相互关联的任务,其生成无重叠的实例掩码和像素级的跟踪结果不仅消除了歧义性,并且可以更精确的完成多目标跟踪任务.

多目标跟踪算法一般被分解为检测任务和跟踪任务(数据关联):首先检测整个视频序列上所有目标的位置,再通过关联算法为每个目标分配一个独有 ID,以此构建各个目标轨迹,这无疑增加了推理时间.而本文算法是将检测器转换为跟踪器,联合检测任务和跟踪任务,在检测的同时就可以完成跟踪.此外本文改进了 Tian[2] 等人提出的一种基于条件卷积的实例分割网络(CondInst 网络),以实现多目标分割跟踪的目的,并从提升实例分割性能和优化数据关联算法两个层面出发,提出了全新的一种融合运动特征嵌入的多目标分割跟踪算法:1)设计了一种将运动特征嵌入表观特征的网络结构,增强了网络的特征表现能力,减少了误检率和漏检率,并提高

收稿日期:2021-10-18 收修改稿日期:2021-12-15 基金项目:国家自然科学基金项目(62020106004,92048301)资助;浙江省自然科学基金项目(LY20F020029)资助;浙江省科技计划公益项目(LGG20F020017)资助. 作者简介:**许营坤**,男,1979 年生,博士,讲师,CCF 会员,研究方向为计算机视觉、机器学习;**陈天阳**,男,1997 年生,硕士研究生,CCF 会员,研究方向为多目标跟踪;**陈胜勇**,男,1973 年生,博士,教授,博士生导师,CCF 高级会员,研究方向为计算机视觉和图像处理;**徐新黎**,女,1977 年生,博士,副教授,CCF 会员,研究方向为智能计算、医学图像计算、网络表示学习.

了实例分割的性能和跟踪的精度;2)为了减少因检测器失效和目标遮挡而导致的漏检,设计了两个新的损失函数,有效提升了网络预测实例的性能;3)设计了一种关联网络,可为目标分配有区分度的外观特征向量,并提出了一种预测并关联的更新轨迹策略,同时利用目标的外观信息和位置信息,提升了跟踪准确度,生成了可靠的跟踪结果. 本文在 MOTS20 公开标准数据集上与同类算法进行评估对比,实验结果表明本文所提算法具有更高的跟踪准确度.

2 相关工作

目前存在的多目标跟踪算法按照跟踪的方式可分为基于检测的跟踪和基于分割的跟踪.

2.1 基于检测的跟踪

基于检测的跟踪[3-5]是通过矩形框的形式来表示跟踪目标的位置,并通过关联矩形框区域形成目标轨迹. 由于目标间的相互遮挡可能导致关联错误,研究人员[6,7]通过设计不同的关联算法对 ID 切换问题贡献了自己的解决方案.

其中 Zhang[8]等人提出的 FairMOT 采用无锚框的检测器来定位高分辨率特征图上的目标中心,并在网络中添加了一个并行分支,用于预测目标身份的逐像素 Re-ID 特征,在高分辨率特征图上提取的 Re-ID 特征能够更好地与目标中心对齐,通过学习低维度的 Re-ID 特征,不仅减少了计算时间,而且提高了特征匹配的稳健性.

2.2 基于分割的跟踪

基于分割的跟踪是通过像素级的二值掩码来表示跟踪目标的位置,并通过关联分割区域以形成目标轨迹.

目前这类算法研究的热点都是基于 Mask RCNN[9]网络进行改进:Voigtlaender[10]等人提出的 TrackR-CNN 在主干网和候选区域生成网络之间添加了两个 3D 卷积层,该 3D 卷积层的前两个维度是空间信息,第 3 个维度是时间信息,增强了多帧间的信息关联,此外还添加了一个关联分支用于生成每个目标的特征向量,用于数据关联;Yang[11]等人提出的 MaskTrack R-CNN 添加了与边界框分支和掩码分支并行的跟踪分支,此分支由两个全连接层组成,用于为每一个候选框分配一个实例标签以构建轨迹;Porzi[12]等人提出的 MOTS-Net 网络的主干由 ResNet-50[13]和特征金字塔网络(FPN)[14]组成,将 5 种不同分辨率的多尺度特征输入到候选区域生成网络中,并输出图像中实例可能位于的候选区域,再从候选区域中提取实例特征,输入到分割头和跟踪头,额外添加的跟踪头与分割头并行运行,并使用 mask pool 增强目标在时间上的关联性. Lin 等[15]人提出 VIST_MVAEA 则是一种改进后的变分自编码器(VAE)架构,通过构建一个共享编码器和 3 个并行解码器,产生 3 个不相交的分支,分别用于预测未来帧、目标检测框和实例分割掩码,通过引入高斯过程模型放宽传统 VAE 的独立同分布假设并提取变量之间相关性,增强了 VAE 的统计表示,该网络通过学习视频数据中嵌入的空间相关性和运动连续性,可有效生成高质量的分割掩码并跟踪视频中的多个实例.

不同于上述基于 Mask RCNN 网络的方法,Xu[16]等人的 PointTrack 是基于无锚框的实例分割网络 SpatialEmbedding[17]实现,并介绍了一种像素级的特征学习策略,对于每个实例,PointTrack 将分割的前景和背景视为两个无序 2D 点云进行记录,学习有区分的实例嵌入,形成了一种逐点跟踪的方法. 此外,在逐点提取特征时,PointTrack 会添加多模态数据,如偏移量,颜色,类别和位置,以丰富点状特征,增强关联.

3 算法跟踪模块

本文提出的一种融合运动特征嵌入的多目标分割跟踪算法共包含实例分割模块、实例预测模块、实例关联模块和轨迹删除模块 4 部分组成. 其中实例分割模块负责获取每一帧的实例分割结果,实例预测模块负责通过上一帧的实例分割结果预测出当前帧的分割结果,实例关联模块负责提取实例的外观信息和位置信息,并进行实例间的关联,轨迹删除模块负责删除消失的轨迹.

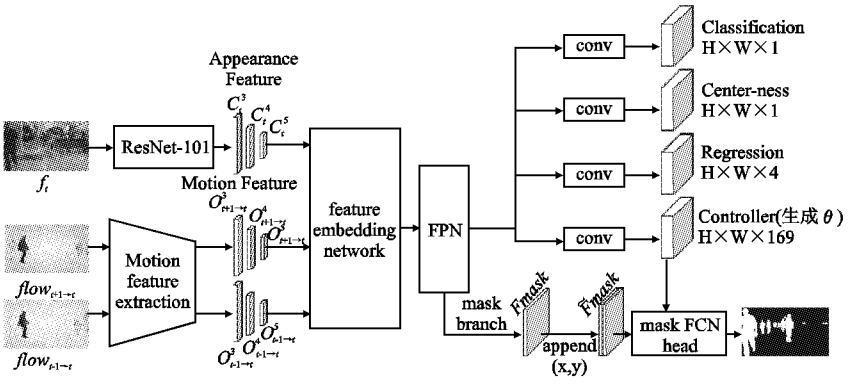


图 1 实例分割网络结构图

Fig. 1 Diagram of instance segmentation network structure

3.1 实例分割模块

实例分割网络的性能很大程度上会影响多目标分割跟踪器的性能,因此对实例分割网络的改进可以有效的提高多目标分割跟踪的准确度. 本文的多目标跟踪算法是基于条件卷积实例分割网络(CondInst 网络),这是一种单阶段、无锚框的全卷积网络,可以生成整个图像的实例掩码,有效去除了分割依赖矩形框的限制,由于其简单的模型结构,可以灵活的扩展为解决多目标分割跟踪任务的网络框架. 但由于 CondInst 网络只利用了单一特征并且未利用视频序列上的时间维度信息,因此本文在 CondInst 网络的基础上添加了运动特征提取

网络和特征嵌入网络.

3.1.1 CondInst 网络

CondInst 网络结构如图 1 所示,其采用 ResNet-101 网络和 FPN 网络作为主干网,假设当前帧图片为 t 帧,ResNet-101 网络共提取 3 层表观特征(Appearance Feature),分别为 C_t^3、C_t^4 和 C_t^5,分类头(Classification)预测实例在(x,y) 位置上的类别概率,中心头(Center-ness)用于抑制远离目标中心的低质量检测,回归头(Regression) 负责预测目标边界框,$\bar{F}_{mask} \in \mathbb{R}^{H_{mask} \times W_{mask} \times (8+2)}$ 是通过结合 $F_{mask} \in \mathbb{R}^{H_{mask} \times W_{mask} \times 8}$ 和坐标图 $(x,y) \in \mathbb{R}^{H_{mask} \times W_{mask} \times 2}$ 得到的,其中 F_{mask} 为掩码分支(mask branch)生成的特征图,坐标图是从 F_{mask} 上的所有位置到掩码头(mask FCN head)产生位置的相对坐标,H_{mask} 和 W_{mask} 分别为网络输入图片高宽的 1/4. 控制器头(Controller)用于预测在(x,y)位置上的实例的掩码头参数 θ,掩码头是一个非常紧凑的全卷积结构,内有 3 个 1×1 卷积,前两个卷积为 8 通道,最后一个卷积为 1 通道,使用 ReLU 作为激活函数,因此掩码头总共有 169 维的参数(169=[(8+2)×8+8]+(8×8+8)+(8×1+1)),网络通过参数 θ 与 $\tilde{F}(x,y)$ 预测出实例的二值化分割掩码.

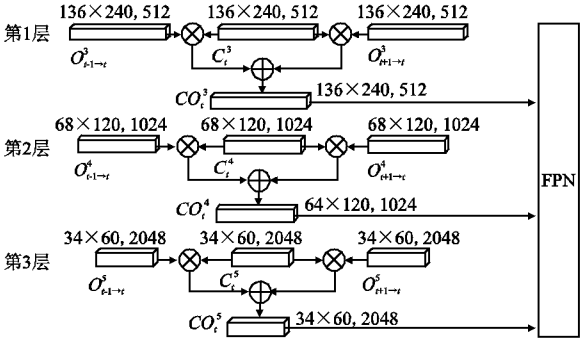


图 2 特征嵌入网络结构图

Fig. 2 Structure diagram of features embedding network

3.1.2 运动特征提取网络

图 1 中的运动特征提取网络(Motion feature extraction)是由卷积核、归一化层和 ReLU 激活函数组成,并与 ResNet-101 的网络层一一对应,负责提取光流场中的运动特征,并与同一层的表观特征的通道和尺寸分别保持一致. 本文首先通过相邻 3 帧作为 VCN[18] 网络的输入,获取每一帧与相邻帧的光流场 $flow_{t-1\to t}$ 和 $flow_{t+1\to t}$,$flow_{t-1\to t}$ 表示 t-1 帧到 t 帧的光流场,$flow_{t+1\to t}$ 表示 t+1 帧到 t 帧的光流场,将两个光流场分别输入到运动特征提取网络提取出 t-1 帧到 t 帧的运动特征 $O_{t-1\to t}^3$、$O_{t-1\to t}^4$、$O_{t-1\to t}^5$ 以及 t+1 帧到 t 帧的运动特征 $O_{t+1\to t}^3$、$O_{t+1\to t}^4$、$O_{t+1\to t}^5$,这些特征都作为特征嵌入网络(Feature Embedding Network)的输入.

3.1.3 特征嵌入网络

图 1 中的特征嵌入网络的具体结构如图 2 所示,本文以 1080×1920 大小的图片作为输入为例说明,输入的 3 层特征的通道数分别为 512、1024 和 2048,尺寸分别为 136×240、68×120 和 34×60,嵌入后的特征表示为 $CO_t^i(i=3,4,5)$,具体如公式(1)所示:

$$CO_t^i = C_t^i \otimes O_{t-1\to t}^i + C_t^i \otimes O_{t+1\to t}^i + C_t^i, i=3,4,5 \quad (1)$$

其中⊗表示逐元素乘法,即在每个通道切片之间应用元素乘法,+表示逐元素相加. 将运动特征与表观特征相乘,可以提取出公共特征部分,滤除无用的运动特征,再将公共部分与表观特征按元素相加,起到补充运动信息的作用,嵌入后的特征 $CO_t^i=(i=3,4,5)$ 同时具有表观信息和运动信息.

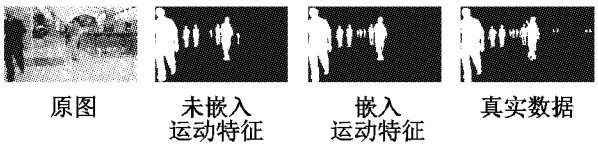


图 3 分割效果的对比

Fig. 3 Comparison of segmentation effects

3.1.4 初始化轨迹

随着视频序列的输入,本文使用图 1 网络结构逐一获取视频序列上每一帧图像的分割结果并同时生成跟踪轨迹. 每一帧的分割结果设为 $D=\{s,b,m,F_{mask},L,\theta\}$,其中的 s 表示置信度得分,b 为边界框信息,m 为实例掩码,F_{mask} 为网络中掩码分支的输出,L 是可以代表目标位置的信息,θ 为掩码头的参数. 当第一帧图像输入网络时,本文用第一帧的分割结果来初始化轨迹,如图 3 所示,相比较未嵌入运动特征的实例分割结果,嵌入运动特征后的实例分割结果边缘更加细腻、精度更高、分割的目标更多,也更加接近真实数据.

3.2 实例预测模块

添加预测模块可有效的减少在跟踪过程中由于检测器失效和目标遮挡而导致的漏检,因此本文选择通过光流去预测上一帧目标在当前帧中的位置. 如图 4 所示,首先从 $flow_{t-1\to t}$ 中获取上一帧(t-1 帧)实例上所有像素点的光流向量集合,取光流集合的中值代表该目标的偏移量,并与目标边界框中心位置的和作为该目标在当前帧(t 帧)上的位置 L',并结合该目标的 F_{mask} 以及参数 θ,预测出当前帧的分割结果 $\{s',b',m'\}$,并将 s' 大于阈值 0.6 的目标保存至"集合 P"中,同时将 s' 小于阈值 0.6 的目标保存至不活跃轨迹"集合 UT"中.

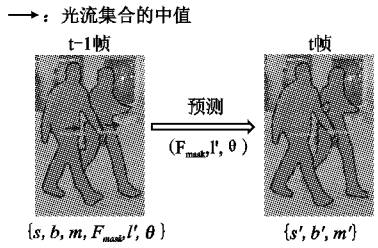


图 4 实例预测示意图

Fig. 4 Diagram of instance prediction

3.3 实例关联模块

在目标关联算法中,目标的位置信息可以用于区分不同目标,但当目标之间相互遮挡严重时,仅凭位置信息往往不能正确的关联目标,导致 ID 切换频繁,而目标的外观特征可有效的缓解上述问题. 在本文中的实例关联模块同时利用了目标的外观特征和位置信息,有效提升了关联目标的性能.

3.3.1 关联网络

如图 5 所示,本文在孪生神经网络[19]的基础上,利用 ResNet50 作为主干网,网络的输入除了目标的边界框 b 以外,额外添加实例掩码 m,将图片和掩码图中边界框 b 所包围的区域裁剪出来,再通过相乘提取出前景部分,去除了背景部分带来的影响,并将尺寸调整为 256×128 输入到主干网中,提取 4 层表观特征,尺寸分别为 64×32、32×16、16×8 和 8×4,通道数分别为 256,512,1024 和 2048. 为了提取更具有区分度的特征,本文使用光流场 $flow_{t-1\to t}$ 和 $flow_{t+1\to t}$ 作为输入,与上述步骤一致,同样通过裁剪和尺寸调整,提取 4 层运动特征,并利用公式(1)将运动特征嵌入表观特征,再通过多层的特征融合,最终在 64×32 的特征图上为每个目标分配 128 维度外观特征向量,并使相同目标的外观特征向量之间欧式距离较小,而不同目标的外观特征向量之间欧式距离较大.

3.3.2　关联流程

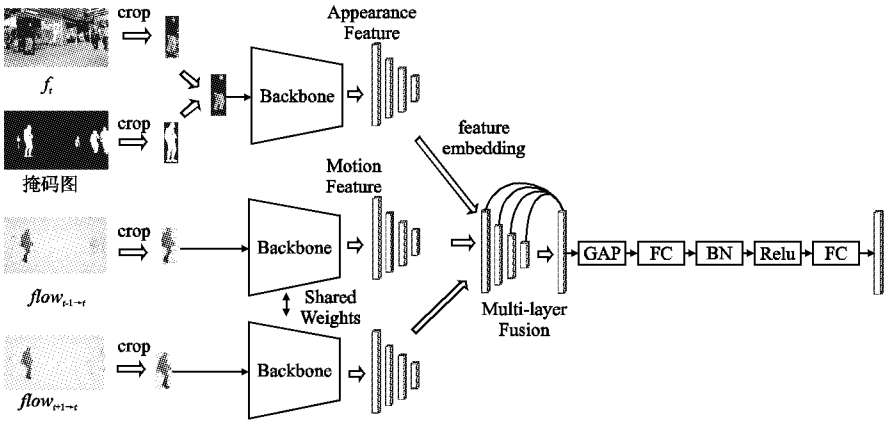


图 5　关联网络结构图

Fig. 5　Diagram of associated network structure

本文将在当前帧上通过光流预测模块得到的分割结果“集合 P”与实例分割模块得到的分割结果“集合 D”进行关联,以此获取可靠的跟踪轨迹. 具体流程如图 6 所示.

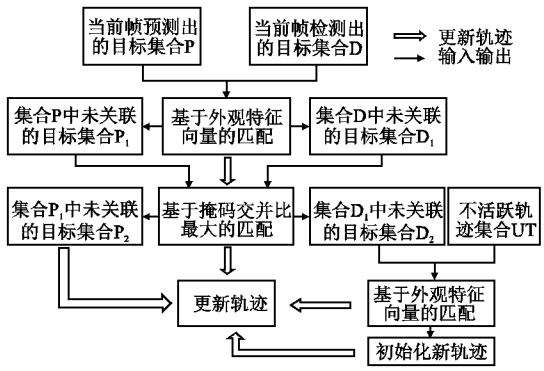


图 6　实例关联模块流程图

Fig. 6　Instance association module flowchart

第 1 步,利用目标外观向量之间的欧氏距离作为关联算法的主要匹配依据,初步获取目标的匹配关系. 以“集合 P”和“集合 D”作为输入,使用关联网络为两个集合中的每个目标分配有区分度的外观特征向量,再通过匈牙利[20]算法将外观特征向量之间的欧氏距离小于 2.0 的目标进行关联,成功关联的目标更新至当前轨迹;反之,将未匹配的目标分别放入“集合 P_1”和“D_1”.

第 2 步,利用目标之间的掩码重叠度作为辅助匹配依据,有效避免了由于第 1 步关联失败而导致一个目标的预测结果和分割结果都被保留的情况. 以“集合 P_1”和“集合 D_1”作为输入,使用基于掩码交并比(mask IoU) 最大的关联方式,将交并比大于 0.2 的目标进行关联,成功关联的目标更新至当前轨迹;反之,对于“集合 P_1”中未匹配的目标放入“集合 P_2”,并直接更新到当前轨迹,“集合 P_1”中未匹配的目标移入“集合 D_2”中.

第 3 步,判断不活跃轨迹“集合 UT”中的目标是否在当前帧中出现. 以“集合 D_2”和“集合 UT”为输入,仍使用基于外观特征向量的关联算法,成功匹配的目标更新到当前轨迹,并将其从“集合 UT”删除. 最终未匹配的目标初始化为新轨迹.

3.4　轨迹删除模块

对于轨迹的消失,本文的跟踪器将考虑以下 3 种情况来删除轨迹:1)如果目标不在图像中或被严重遮挡;2)将非最大抑制(NMS)应用于 IoU 阈值大于 φ 的互斥轨迹;3)设置一个最大阈值 $age=10$,不活跃轨迹中的目标连续 age 帧没有与新目标关联.

3.5　本文算法整体流程

本文算法在 4 个模块之间的整体跟踪流程如算法 1 所示.

算法 1. 多目标分割跟踪算法

输入:图像序列 $I=\{I_t\}_{t=1}^{F}$ 和运动特征 M,其中 F 为总序列数,当前帧设为 t 帧,当前帧图片记作 I_t

输出:图像序列的轨迹 $T_t=\{s,b,m,L,id\}$,其中 s 为置信度,b 为边界框,m 为实例掩码,L 为目标位置,id 为身份信息

```
for I_t in I do
    D_t = {s,b,m,F_mask,L,θ} ← Detect(I_t,M); //实例分割模块
    for d in D_t do
        if s < 检测阈值 then
            将 d 从{D_t}中移除;
        end if
    end for
    D_t ← NMS (D_t);
    If t == 0 then
        利用 D_t 去初始化轨迹 T_t;
    else
        for track in T_{t-1} do
```

```
            optical flow O← VCN(I_{t-1}, I_t); //使 VCN 网络计
            I_{t-1} 到 I_t 的光流场 O
            取 O 的中值作为偏移量 last_v;
            L'←center(b) + last_v;
        end for
        P_t = Predict(F_mask, L', θ, I_t); //实例预测模块
        for p in P_t
            if s < 预测阈值 then
                将 p 移入 UT = {s, b, m, L, id, age};
            end if
        end for
        P_t, UT← NMS(P_t); //将重叠的目标移入 UT
        T_t = associate_1(P_t, D_t); //实例关联模块,使用外观向量和掩码交并比关联目标
        将没有成功关联的目标移入集合 D_2;
        T_t = associate_2(D_2, UT);
    end if
    for ut in UT
        age←age + 1;
        if age > 10 then
            将 ut 从 UT 中移除; //轨迹删除模块
        end if
    end for
end for
```

4 损失函数

为了提升光流预测模块的性能以达到降低漏检率的目的,本文设计了两种损失函数分别用 L_c 和 L_a 表示. 本文算法使用的总损失函数如公式(2)所示:

$$L_{overall} = L_{fcos} + L_{mask} + L_c + L_a \tag{2}$$

其中 L_{fcos} 和 L_{mask} 为 CondInst[1] 工作中的损失函数.

$$L_c = \frac{1}{k}\sum_{i\in X} L_{dice}(M^t_{pred}, M^t_i) \tag{3}$$

$$M^t_{pred} = MaskHead(\tilde{F}^t_{mask}(x^t_i, y^t_i); \theta^{t-1}_i) \tag{4}$$

公式(3)、公式(4)为 L_c 的表达式,用于增强参数 θ 在时间上的关联性,目的是为了当前帧的参数 θ 也可以很精确的用于分割下一帧图像上的目标,以此提升光流预测模块的性能. 其中 $X = \{X_1, X_i, \cdots, X_k\}$ 是两张相邻图像帧上共同目标的集合,K 是集合 X 的基数. $\tilde{F}^t_{mask}(x^t_i, y^t_i)$ 是在 t 帧上 id 等于 i 的目标的网络参数,θ^{t-1}_i 是在 t-1 帧上同一目标的掩码头参数. *MaskHead* 为掩码头,生成预测的实例掩码 M^t_{pred}. M^t_i 是 t 帧上 id 等于 i 实例的标注分割掩码. L_{dice} 是损失函数(dice loss[21]).

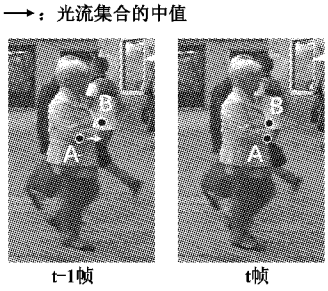


图 7 光流错误示意图

Fig. 7 Diagram of optical flow error

通过光流预测模块获取的位置存在不准确性的问题,如图 7 所示:t-1 帧上目标 A 的光流集合中值可能会使预测在 t 帧上的坐标出现在目标 B 的身上,这会导致预测的分割结果不精确. 因此本文在三元组损失函数的基础上,设计了 L_a 损失函数,具体如公式(5)、公式(6)所示:

$$L_a = \frac{1}{N}\sum_{i\in D} max\{\max_{j\in D}\{DSC(M^{flow}_{pred}, M^t_j)\} - DSC(M^{flow}_{pred}, M^t_i) + \alpha, 0\} \tag{5}$$

$$M^{flow}_{pred} = MaskHead(\tilde{F}^t_{mask}(x^{flow}_i, y^{flow}_i); \theta^{t-1}_i) \tag{6}$$

其中 $D = \{D_1, D_2, \cdots, D_N\}$ 是从公式(3)的 K 集合中选取真实边界框有交集的目标,N 是集合 D 的基数. $\tilde{F}^t_{mask}$ 是结合 F^t_{mask} 和 (x^{flow}_i, y^{flow}_i) 的参数,其中 F^t_{mask} 是在 t 帧上的掩码分支输出,(x^{flow}_i, y^{flow}_i) 是通过预测模块所获得的在 t 帧上的相对坐标图. θ^{t-1}_i 是在 t-1 帧上 id 等于 i 目标的掩码头参数. *MaskHead* 表示掩码头,用于生成预测的实例掩码 M^{flow}_{pred}. M^t_j 是在 t 帧上集合 D 中 id 不等于 i 的目标的标注分割结果. *DSC* 的表达式为 $DSC(A,B) = 2|A\cap B|/(|A| + |B|)$,用于计算 4 预测的实例掩码与真实掩码的重叠程度.

5 实验与结果分析

本文算法在公开标准数据集 MOTS20 跟踪基准上评估实验结果. 此外,还展示了消融实验的研究结果,验证了本文网络的有效性.

5.1 数据集

MOTS20 数据集是从 MOTChallenge 2017[22] 数据集中选取了部分视频序列,训练集和测试集分别有 4 个视频序列. 其中选取了 MOTChallenge 2017 数据集中的第 2 个、第 5 个、第 9 个和第 11 个视频序列添加实例分割标注,分别生成 MOTS20-02、MOTS20-05、MOTS20-09 和 MOTS20-11 作为训练集,共 2862 帧,228 个真实轨迹,26894 个行人. 并选取了 MOTChallenge 2017 数据集中的第 1 个、第 6 个、第 7 个和第 12 个视频序列添加实例分割标注,分别生成 MOTS20-01、MOTS20-06、MOTS20-07 和 MOTS20-12 作为测试集,共 3034 帧,包含 328 个真实轨迹,32269 个行人. MOTS20 数据集与其他数据集相比,主要针对于行人的跟踪,并且行人之间相互遮挡频繁,因此跟踪难度较大,非常具有挑战性.

5.2 评价指标

对于多目标分割跟踪任务的测评,本文使用 Track-RCNN 中提供的评估工具[1],其对于多目标分割跟踪算法的评估指标如表 1 所示,其中 MOTSA 和 MOTSP 是对跟踪准确度和跟踪精度的评估,sMOTSA 是 MOTSA 的软版本,是作为评估多目

表 1 多目标分割跟踪算法指标

Table 1 Metrics used for multi-object tracking and segmentation

Measure	Better	Prefect	Description
sMOTSA	higher	100%	Mask-based Soft Multi-Object Tracking Accuracy
MOTSA	higher	100%	Mask-based Multi-Object Tracking Accuracy
MOTSP	higher	100%	Mask-overlap based variant of Multi-Object Tracking Precision
FP	lower	0	The total number of false positives
FN	lower	0	The total number of false negatives (missed targets)
IDS	lower	0	Number of Identity Switches

[1] https://github.com/VisualComputingInstitute/mots_tools

标分割跟踪任务的主要指标. FP(误检率)和 FN(漏检率)是对检测性能的评估,IDS 为 ID 切换,是对跟踪器是否给目标正确分配 ID 的评估.

5.3 训练过程

本文实验在 Inter Core i9-9900KF 3.6GHz CPU、两张 2080Ti 显卡、内存 16G 的台式电脑上完成. 在训练集的 4 个视频序列上使用留一法(使用其中 3 个视频序列训练得到的模型去评估剩下的视频序列)的方式训练并评估了本文的跟踪结果;对于在测试集上的评估,则使用在全部训练集上训练得到的模型进行验证.

5.3.1 实例分割网络训练过程

实例分割网络使用 ResNet-101 和 FPN 作为主干网络,加载了在 coco 数据集上训练得到的预训练模型,并在 MOTS20 训练集上对网络进行了微调,训练时选择 SGD 优化器,初始学习率设为 1×10^{-4},共迭代 600000 次,分别在 320000 和 480000 处,使学习率下降 10 倍. 此外,为了提高检测和分割质量以及预防过拟合问题,本文使用了常见的数据增广方法,包括水平翻转,随机裁剪,颜色抖动和添加高斯噪声等.

表 2 MOTS20 训练集上结果对比

Table 2 Results on the MOTS20 training set

Train	sMOTSA ↑	MOTSA ↑	MOTSP ↑	FP ↓	FN ↓	IDS ↓
TrackR-CNN	52.7	66.9	80.2	**879**	6602	310
MOTSNet	56.8	69.4	82.7	/	/	/
PointTrack	58.1	70.5	/	/	/	/
MaskTrack R-CNN	50.5	66.7	78.3	1882	7012	/
VIST_MVAEA	59.5	71.5	84.7	1537	5641	/
Ours	**66.0**	**77.9**	**85.7**	1035	**4628**	**291**

5.3.2 关联网络训练过程

关联网络使用 ResNet50 作为主干网,加载了在 ImageNet 数据集上训练得到的预训练模型,通过数据预处理将 MOTS20 训练集中的目标进行裁剪,并调整大小为 256×128,并将同一目标分为一组,每次输入 4 组数据用于关联网络的训练,训练时选择 Adam 优化器,初始学习率为 1×10^{-4},共训练 200 个 epoch.

5.4 实验结果及分析

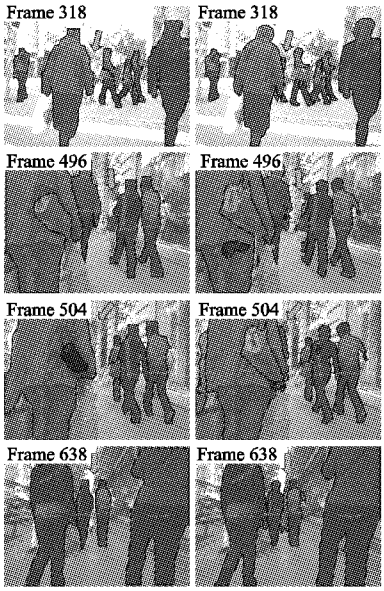
表 2 展示了在 MOTS20 训练集上本文算法与其他算法的结果对比,本文算法在 5 项指标上均领先于其他多目标分割跟踪算法,并且由于实例预测模块以及新增的损失函数在时间维度上增强了同一目标的关联性,因此在 FN 指标上更是远远优于其他算法. 就 sMOTSA 指标而言,本文算法比仅使用表观信息的 TrackR-CNN 高出了 13.3%,同时,在 IDS 指标方面,也比仅使用外观信息关联目标的 TrackR-CNN 低. 在 MOTS20 测试集上,同样将本文算法与最新的同类算法作了对比,结果如表 3 所示,本文算法在各项指标上仍处于前列.

表 3 MOTS20 测试集上结果对比

Table 3 Results on the MOTS20 testing set

Test	sMOTSA ↑	MOTSA ↑	MOTSP ↑	FP ↓	FN ↓	IDS ↓
DD_Vision[23]	**66.6**	**79.7**	**84.4**	**1067**	**5155**	**341**
TraDeS[24]	50.8	65.5	79.5	1474	9169	492
TrackR-CNN	40.6	55.2	76.1	1261	12641	567
Ours	63.1	77.2	82.9	1307	5617	418

图 8 展示了在拥挤的行人场景下,本文算法与 TrackR-CNN 的跟踪效果对比,可以看出,当 TrackR-CNN 无法对某些被遮挡严重的行人进行跟踪时,本文算法依然可以对这些行人进行高质量的跟踪,这也表明了本文所提出的算法更加有效,展现了本文算法的竞争力.



(a)TrackR-CNN的跟踪结果 (b)本文算法的跟踪结果

图 8 MOTS Challenge 数据集上的示例结果

Fig. 8 Sample results on the MOTS Challenge dataset

5.5 消融实验

为了验证了特征嵌入模块、新增损失函数以及关联网络的有效性,本文使用 MOTS20-05、MOTS20-09 和 MOTS20-11 序列对网络进行训练,并在MOTS20-02序列上进行评估. 实验结果如表 4 所示,在训练阶段分别添加 L_c 和 L_a 损失函数后,FN 都得到了有效的降低,同时 sMOTSA 指标也总共提升了 2.5%. 在此基础上添加特征嵌入模块进行训练后,sMOTSA 指标再次提升了 5.9%,FN 和 FP 也都得到了大幅度降低,提升了分割性能.

表 4 MOTS20-02 图像序列上的结果对比

Table 4 Comparison of results on the MOTS20-02 video sequence

L_c	L_a	特征嵌入	sMOTSA ↑	MOTSA ↑	MOTSP ↑	FP ↓	FN ↓	IDS ↓
-	-	-	54.9	68.7	81.2	261	1880	62
√	-	-	56.8	71.3	81.0	300	1659	59
√	√	-	57.4	72.6	80.8	373	1492	67
√	√	√	63.3	77.5	82.5	179	1331	74

表 5 MOTS20 训练集上不同更新轨迹的方式的结果对比

Table 5 Comparison of results on MOTS20 training set for different ways of updating trajectories

光流预测	Mask IoU	关联网络	sMOTSA ↑	MOTSA ↑	MOTSP ↑	FP ↓	FN ↓	IDS ↓
√	-	-	49.2	63.0	81.0	2129	7383	448
-	√	-	62.8	74.3	85.9	1484	5018	408
-	√	√	65.5	76.5	86.2	603	5433	294
√	√	√	66.0	77.9	85.7	1035	4628	291

为验证关联网络以及预测并关联的更新轨迹策略的有效

性,本文在 MOTS20 的训练集上进行了验证,表 5 展示了在本文实验中使用不同的更新轨迹方式的跟踪结果对比,实验表明在 sMOTSA 指标上,仅使用光流预测的方式只达到了 49.2%,添加关联网络比仅使用 mask IoU 的关联方式高出了 2.7%,而预测并关联的方式使 sMOTSA 指标达到了最高的 66.0%,证明了本文所提的关联网络以及预测并更新策略的有效性.

针对多目标分割跟踪任务中不允许存在重叠掩码的要求,本文选择了两种方式进行对比实验:1)将重叠的像素点分配给置信度更大的目标;2)将重叠的像素点分配给在 y 轴(图片的竖轴)上更靠前的目标. 表6展示了两种分配方式在

表 6 重叠掩码的不同处理方式对比

Table 6 Comparison of different processing methods of overlap mask

Train	sMOTSA ↑	MOTSA ↑	MOTSP ↑	FP ↓	FN ↓	IDS ↓
score	66.0	77.9	85.7	1035	4628	291
y	65.8	77.6	85.6	1067	4660	291

训练集上的对比实验结果,score 表示第 1 种方式,y 表示第 2 种方式,实验结果表明选择第 1 种方式可以提高 sMOTSA 指标 0.2%. 因此,本文在跟踪过程中使用了一个简单的标准,即始终将重叠像素分配给具有较高置信度的目标.

6 结 论

本文提出的多目标分割跟踪算法对行人具有很好的跟踪能力,实验结果表明,提出的运动特征嵌入的网络结构,通过嵌入运动信息有效的减少了漏检率和误检率,增强了目标检测器的检测性能. 另外,设计的两个损失函数增强了网络预测实例的性能,进一步的减少了由于检测器失效和目标遮挡而导致的漏检. 同时关联网络和预测并关联的更新轨迹策略很好的构建了目标轨迹,提升了跟踪准确度. 与同类型的算法相比,本文算法表现出了更好的有效性.

References:

[1] Luo W, Xing J, Milan A, et al. Multiple object tracking: a literature review[J]. Artificial Intelligence, 2021, 293, doi: 10.1016/J.ARTINT.2020.103448.

[2] Tian Z, Shen C, Chen H. Conditional convolutions for instance segmentation[C]//European Conference on Computer Vision (ECCV), 2020: 282-298.

[3] Zhou X, Koltun V, Krhenbühl P. Tracking objects as points[C]// European Conference on Computer Vision (ECCV), 2020: 474-490.

[4] Bergmann P, Meinhardt T, Leal-Taixe L. Tracking without bells and whistles[C]//IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 941-951.

[5] Zhang, Jimuyang, Sanping Zhou, et al. Multiple object tracking by flowing and fusing[EB/OL]. https://arxiv.org/abs/2001.11180, 2020.

[6] Wang Z, Zheng L, Liu Y, et al. Towards real-time multi-object tracking[C]//European Conference on Computer Vision (ECCV), 2020: 107-122.

[7] Shuai B, Berneshawi A G, Modolo D, et al. Multi-object tracking with siamese track-RCNN[EB/OL]. https://arxiv.org/abs/2004.07786, 2020.

[8] Zhang Y, Wang C, Wang X, et al. FairMOT: on the fairness of detection and re-identification in multiple object tracking[J]. International Journal of Computer Vision, 2021: 1-19, doi: 10.1007/s11263-021-01513-4.

[9] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[C]//IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017: 2961-2969.

[10] Voigtlaender P, Krause M, Osep A, et al. MOTS: multi-object tracking and segmentation[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019: 7942-7951.

[11] Yang L, Fan Y, Xu N. Video instance segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 5188-5197.

[12] Porzi L, Hofinger M, Ruiz I, et al. Learning multi-object tracking and segmentation from automatic annotations[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 6845-6854.

[13] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 770-778.

[14] Lin T Y, Dollar P, Girshick R, et al. Feature pyramid networks for object detection[C]//IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 2117-2125.

[15] Lin C C, Ying H, Feris R, et al. Video instance segmentation tracking with a modified VAE architecture[C]//IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 13147-13157.

[16] Xu Z, Zhang W, Tan X, et al. Segment as points for efficient online multi-object tracking and segmentation[C]// European Conference on Computer Vision (ECCV), 2020: 264-281.

[17] Neven D, Brabandere B D, Proesmans M, et al. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth[C]//Conference on Computer Vision and Pattern Recognition (CVPR), 2019: 8837-8845.

[18] Yang, Gengshan, Deva Ramanan. Volumetric correspondence networks for optical flow[C]//Advances in Neural Information Processing Systems, 2019: 794-805.

[19] Ristani, Ergys, Carlo Tomasi. Features for multi-target multi-camera tracking and re-identification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018: 6036-6046.

[20] Kuncheva L I. Full-class set classification using the hungarian algorithm[J]. International Journal of Machine Learning and Cybernetics, 2010, 1(1-4): 53-61.

[21] Milletari F, Navab N, Ahmadi S A. V-net: fully convolutional neural networks for volumetric medical image segmentation[C]//4th International Conference on 3D Vision (3DV), 2016: 565-571.

[22] Milan A, Leal-Taixé L, Reid I, et al. MOT16: a benchmark for multi-object tracking[EB/OL]. https://arxiv.org/abs/1603.00831, 2016.

[23] Liu Y, Lyuwei W, Yuan Z, et al. Tracking by segmentation: person-ReID and optical flow based offline tracker for the MOTSChallenge 2020[EB/OL]. https://motchallenge.net/workshops/bmtt2020/papers/DD_Vision_MOTS20.pdf, 2020.

[24] Jialian W, Jiale C, Liangchen S, et al. Track to detect and segment: an online multi-object tracker[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021: 12352-12361.