中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

基于特征增强和历史帧选择的Transformer视觉跟踪算法

侯志强, 杨晓麟, 马素刚, 王云龙, 余旺盛, 王昀琛

引用本文:
侯志强,杨晓麟,马素刚,王云龙,余旺盛,王昀琛. 基于特征增强和历史帧选择的Transformer视觉跟踪算法[J]. 控制与决策, 2024, 39(10): 3506−3512.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.1048

基于特征增强和历史帧选择的Transformer视觉跟踪算法

侯志强[1,2†], 杨晓麟[1,2], 马素刚[1,2], 王云龙[1,2], 余旺盛[3], 王昀琛[1,2]

(1. 西安邮电大学 计算机学院，西安 710121；2. 西安邮电大学 陕西省网络数据分析与智能处理重点实验室，西安 710121；3. 空军工程大学 信息与导航学院，西安 710100)

摘 要: 为进一步提升跟踪算法在历史帧信息利用和目标特征表达方面的性能,提出基于特征增强和历史帧选择的Transformer视觉跟踪算法(feature enhancement and history frame selection based Transformer visual tracking, FEHST). 首先,在骨干网络中引入动态预测模块,通过稀疏化策略提高自注意力机制的计算效率,聚焦目标区域特征;其次,提出特征增强模块,将局部信息与全局信息的优势相结合,提升特征的表达能力;最后,采用自适应历史帧选择策略,提升跟踪器对目标动态信息的关注. 在LaSOT、TrackingNet、GOT-10K和OTB100等数据集上进行了大量的实验,实验结果显示,在LaSOT、TrackingNet、OTB100上分别取得70.1%、83.0%和71.6%的成功率,在GOT-10K上取得71.4%的平均重叠度,并能以27 FPS的速度运行.

关键词: 计算机视觉；视觉跟踪；深度学习；注意力机制；历史帧选择；Transformer

中图分类号: TP391.4 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.1048

引用格式: 侯志强, 杨晓麟, 马素刚, 等. 基于特征增强和历史帧选择的Transformer视觉跟踪算法[J]. 控制与决策, 2024, 39(10): 3506-3512.

Feature enhancement and history frame selection based Transformer visual tracking

HOU Zhi-qiang[1,2†], *YANG Xiao-lin*[1,2], *MA Su-gang*[1,2], *WANG Yun-long*[1,2], *YU Wang-sheng*[3], *WANG Yun-chen*[1,2]

(1. School of Computer，Xi'an University of Posts & Telecommunications，Xi'an 710121，China；2. Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing，Xi'an University of Posts & Telecommunications，Xi'an 710121，China；3. School of Information & Navigation，Air Force Engineering University，Xi'an 710100，China)

Abstract: To enhance the performance of tracking algorithms in utilizing historical frame information and articulating target features, this paper proposes the feature enhancement and history frame selection based Transformer visual tracking (FEHST) algorithm. Firstly, a dynamic prediction module is integrated into the backbone network with a sparsification strategy to enhance the self-attention mechanism's computational efficiency, focusing on the target region's features. Then, a feature enhancement module is introduced, merging local and global information to improve feature representation. Finally, an adaptive history frame selection strategy is adopted to enhance focus on target dynamics and algorithm robustness. Experiments on LaSOT, TrackingNet, GOT-10K, and OTB100 datasets are carried out to validate the algorithm, showing success rates of 70.1%, 83.0%, and 71.6%, and a 71.4% average overlap on GOT-10K, at 27 FPS.

Keywords: computer vision; visual tracking; deep learning; attention mechanism; history frame selection; Transformer

0 引 言

视觉跟踪是计算机视觉的基本任务之一,旨在第1帧中给定目标初始状态的情况下,预测目标在视频后续序列中的状态[1],在人机交互、自动驾驶、视频监控等领域具有广泛的应用. 然而,视觉跟踪在实际应用中受到多种因素影响[2],要实现鲁棒的视觉跟踪

收稿日期: 2023-07-26；**录用日期:** 2024-01-16.

基金项目: 国家自然科学基金项目(62072370)；陕西省自然科学基金项目(2023-JC-YB-598).

责任编委: 夏元清.

†通讯作者. E-mail: hzq@xupt.edu.cn.

*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

依然是一项极具挑战性的任务. 近年来,基于深度学习的视觉跟踪算法以其优越的跟踪性能得到迅速发展. 2016年,由Bertinetto等[3]提出的全卷积孪生网络SiamFC受到广泛关注,该算法能在保持跟踪精度的同时具备较高的运行速度. 随后,许多学者在SiamFC算法的基础上进行了一系列改进工作,例如基于可变形卷积的孪生网络DCSiam[4]、改进融合深浅层特征的IT-AWCR[5],有效提升了跟踪算法的性能.

最近,Transformer模型已成为自然语言处理领域主流的深度学习模型,随后,Dosovitskiy等[6]将Transformer模型推广到计算机视觉领域. Transformer模型作为一种动态信息聚合器,用于处理空间和时间域的任务,其核心是自注意力机制. 近年来,Transformer模型在视觉跟踪算法中也得到了广泛应用. 例如,Chen等[7]提出的TransT算法是将自注意力模块应用于特征融合网络;Hou等[8]采用混合注意力的方式提升Transformer跟踪算法的性能;Chen等[9]利用Transformer模块增强外观特征. 虽然这些算法都取得了优异的跟踪性能,但仍存在一定的局限性. 首先,由于Transformer模型中自注意力机制的复杂度通常与图像块大小呈平方或线性关系,使得模型计算量较大,而有研究表明,有效的稀疏化策略能够去除计算过程中产生的部分相关度较低的冗余tokens,但这方面的工作在视觉跟踪任务中的尝试较少;其次,尽管Transformer架构在全局建模上的能力较为优秀,但对局部信息的建模能力不足,全局建模和局部建模的优势互补可以有效提升跟踪性能;最后,在视觉跟踪任务中,充分利用历史帧信息可以有效应对目标状态的变化,提升跟踪鲁棒性,而以往基于Transformer的跟踪算法通常忽略了历史帧的运用.

为解决以上问题,本文提出基于特征增强和历史帧选择的Transformer视觉跟踪算法(FEHST),具体内容包括:1)针对Transformer模型计算量大、产生较多冗余tokens的问题,引入动态预测模块(dynamic prediction module, DPM),利用稀疏化策略对冗余tokens进行筛选,同时能够将图像中的目标特征进行聚焦,以便于更好地估计目标状态;2)为将局部信息建模与全局信息建模的优势互补,提出特征增强模块(feature enhancement module, FEM),通过通道划分的方式将特征分别经过局部信息建模和全局信息建模,再利用特征融合模块对信息进行充分融合,增强特征表达能力;3)引入历史帧信息以提升跟踪模型应对目标状态变化的能力,并通过自适应历史帧选择策略筛选历史帧模板,获取更为可靠的历史帧,增强算法的鲁棒性.

本文在LaSOT[10]、TrackingNet[11]、GOT-10K[12]和OTB100[2]等多个数据集上进行大量的测试,实验结果显示,本文算法所提策略能够有效提升跟踪器性能,并以27 FPS的速度实时运行.

1　本文算法

本文提出了基于特征增强和历史帧选择的Transformer视觉跟踪算法FEHST,算法整体框架如图1所示. 整体网络架构包含基于Transformer的特征提取网络、基于Transformer的特征融合网络和预测头网络3个部分.

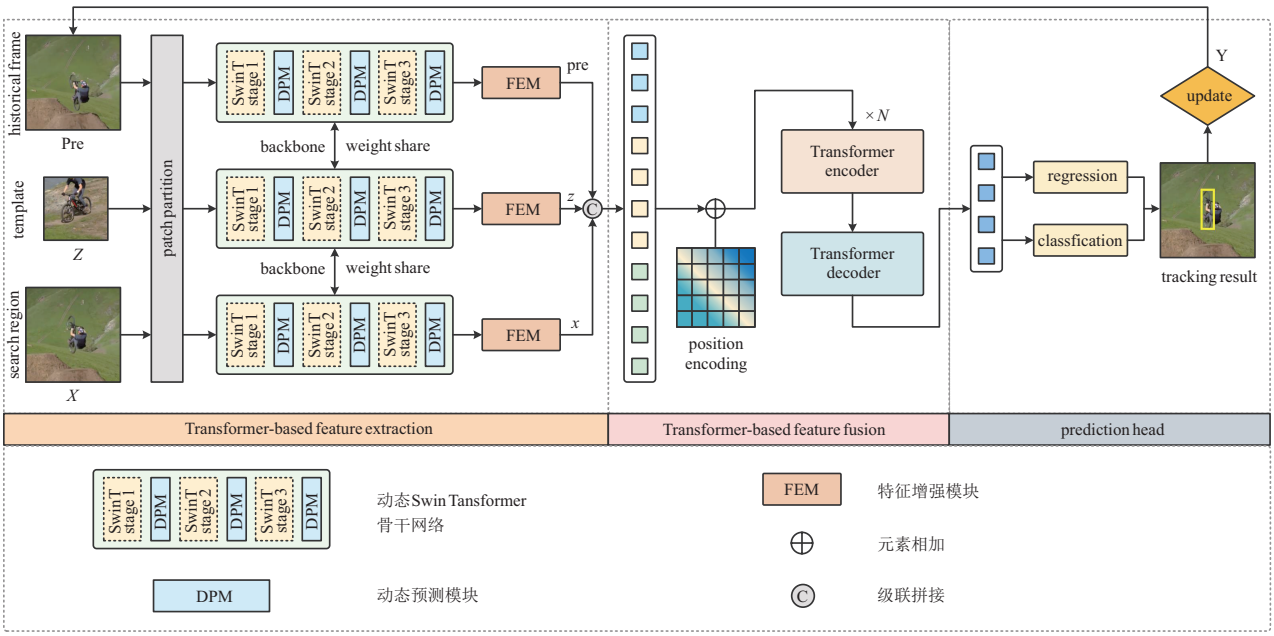

图 1　FEHST网络架构

在基于Transformer的特征提取网络中,历史帧图像pre $\in \mathbb{R}^{H_{\text{pre}}\times W_{\text{pre}}\times 3}$、模板图像$Z \in \mathbb{R}^{H_z\times W_z\times 3}$和搜索图像$X \in \mathbb{R}^{H_x\times W_x\times 3}$作为网络输入,通过基于Transformer的骨干网络提取特征. 骨干网络采用Swin Transformer[13],将其设置为共享权重,第3阶段的输出作为特征提取的最终结果. 与Swin Transformer不同的是,本文引入动态预测模块(dynamic prediction module, DPM),该模块在骨干网络每个阶段之后执行. Swin Transformer骨干网络与DPM模块共同构成动态Swin Transformer骨干网络(dynamic-Swin Transformer backbone). 输入图像经过动态Swin Transformer骨干网络和特征增强模块(feature enhancement module, FEM)之后,历史帧图像输出为pre $\in \mathbb{R}^{\frac{H_{\text{pre}}}{s}\times\frac{W_{\text{pre}}}{s}\times C}$,模板图像输出为$z \in \mathbb{R}^{\frac{H_z}{s}\times\frac{W_z}{s}\times C}$,搜索图像输出为$x \in \mathbb{R}^{\frac{H_x}{s}\times\frac{W_x}{s}\times C}$. 其中:$s$为骨干网络的步幅,$C$为输出特征的通道数.

基于Transformer的特征融合网络先将骨干网络输出图像沿通道维度进行拼接,在拼接后的特征输入到编码器-解码器之前,沿用基线算法SwinTrack[14]中的TUPE位置编码,以便于注意力机制能够知道当前处理的tokens属于哪个分支以及在分支中的位置. 编码器-解码器结构由N个编码器块和一个解码器块组成,编码器通过计算特征之间的相似度来获取全局依赖关系,解码器则通过跨注意力机制生成最终的特征图. 预测头网络包含两个分支:边界框回归分支和分类分支,均接收来自于解码器的输出特征.

接下来,将对本文所提出的动态预测模块(dynamic prediction module, DPM)、特征增强模块(feature enhancement module, FEM)分别进行介绍,并在最后介绍自适应历史帧选择策略.

1.1 动态预测模块(DPM)

动态预测模块的思想来自近年来Transformer架构中的稀疏化操作. 传统卷积操作中,通常利用结构下采样的策略来构建分层架构,例如全局平均池化. 然而,自注意力机制由于其本身特性,可以通过稀疏化操作保留相似性最高的特征信息,从而在保持模型性能的情况下减少模型计算量.

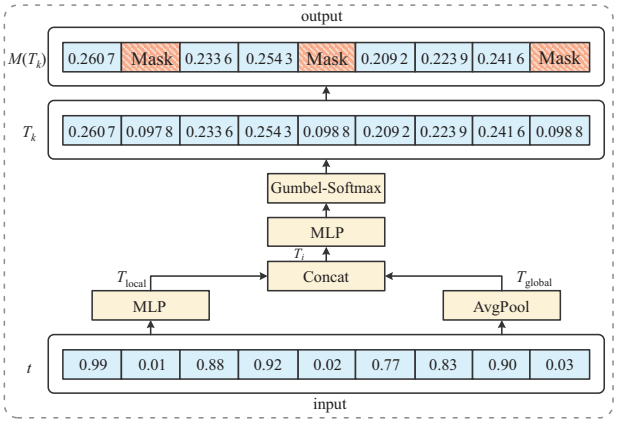

图 2 动态预测模块

动态预测模块在特征提取骨干网络的每个阶段后执行,如图1中骨干网络所示. 该模块采用分层执行的策略,即在骨干网络中,随着计算的执行,逐渐遮蔽相似性较低的tokens,其计算过程如图2所示.

动态预测模块的输入来自于骨干网络每个阶段的输出,记为$t \in \mathbb{R}^{N\times C}$. 其中:$N = H \times W$,$H$和$W$分别表示特征图的高和宽;$C$表示输入特征的通道数. 输入特征$t$分别经过多层感知器(multiple perceptron, MLP)和全局平均池化(AvgPool),从而得到局部编码和全局编码,过程如下:

$$T_{\text{local}} = \text{MLP}(t) \in \mathbb{R}^{N\times C'}, \tag{1}$$

$$T_{\text{global}} = \text{AvgPool}(t) \in \mathbb{R}^{N\times C'}, \tag{2}$$

其中$C' = C/2$. 局部编码包含每个token的信息,全局编码包含整个特征的全局信息,将二者进行拼接,得到包含丰富信息的编码信息,即

$$T_i = [(T)_{\text{local}}, (T)_{\text{global}}]. \tag{3}$$

将得到的编码特征经过MLP和Softmax之后,得到预测丢弃或保留的tokens概率,该过程如下所示:

$$T_k = \text{Softmax}(\text{MLP}(T_i)). \tag{4}$$

此时,输入的特征tokens已被计算出不同的概率分数. 在这里,MLP能够有效地提取和转换特征,增加数据的非线性表达能力;而Softmax操作是通过应用Gumbel-Softmax来突出特征tokens之间的差异性,同时保持操作的可微性. 此外,该操作可以很好地保留原始特征的相对大小关系,从而有助于模型识别和保留重要特征tokens的能力.

常见的稀疏化操作会将概率分数较低的tokens进行丢弃操作,从而降低特征大小,减少计算量. 然而,这一操作会打乱特征排序,导致最终无法重塑为特征图. 概率分数较低的特征tokens对应的是背景区域等与目标无关的区域,本文采用token遮蔽的策略,对需要丢弃的tokens采用零填充,从而能够使特征序列重塑为特征图,不影响后续的分类和回归任务. 遮蔽策略表示如下:

$$M(T_k)_i = \begin{cases} (T_k)_i, & (T_k)_i \geqslant t_r; \\ \text{Masked}, & (T_k)_i < t_r. \end{cases} \tag{5}$$

其中t_r表示决定保留还是遮蔽的阈值,该阈值由设定的稀疏率决定,例如设定稀疏率为0.9,则保留特征

tokens中概率分数前90 %的tokens,其余概率分数较低的tokens执行遮蔽策略,不参与后续计算.

动态预测模块通过稀疏化操作,依赖于MLP和Gumbel-Softmax方法确保了网络的可微性,从而在特征遮蔽时仍能获取有效梯度,保证模型正常训练. 该模块不仅有效地提升了自注意力机制的计算效率,而且有助于集中目标信息,消除干扰目标的噪声. 遮蔽策略主要依靠预设的稀疏率进行,而不是损失函数的直接指导,从而使模型更专注于目标区域,进一步提高了跟踪的准确性和鲁棒性.

1.2　特征增强模块(FEM)

Transformer架构的自注意力机制能够很好地对特征进行全局信息建模,但这种全局建模往往忽视了更为精细的局部特征信息. 为了更好地增强全局和局部信息的特征表达能力,本文提出特征增强模块FEM. 该模块采用通道划分方式,分别对特征进行全局信息和局部信息的增强,并通过融合模块将这些信息进行有效融合. 特征增强模块的结构如图3所示.

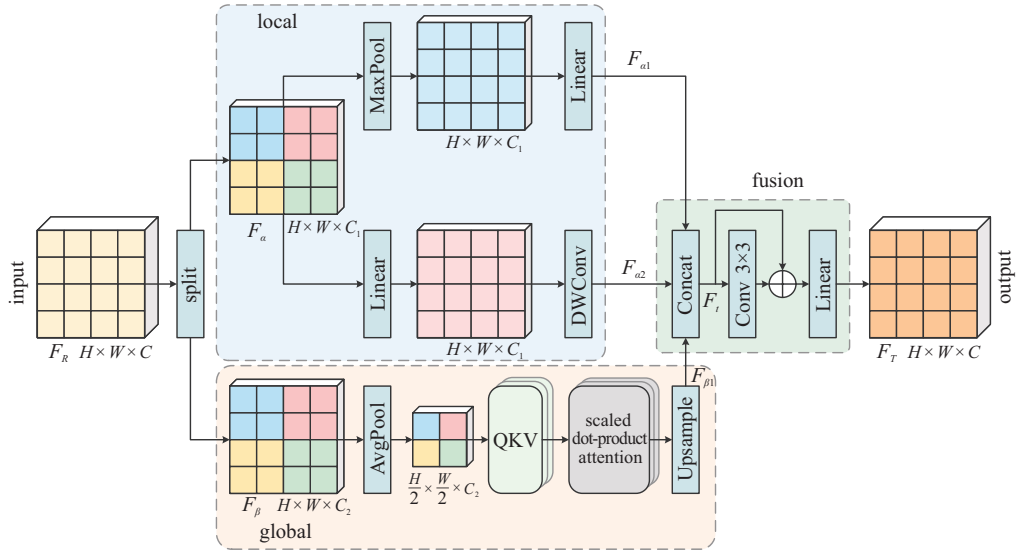

图 3　特征增强模块

该模块将骨干网络的输出特征作为输入$F_R \in \mathbb{R}^{H\times W\times C}$. 其中:$H$和$W$表示特征图的高和宽,$C$表示特征图的通道数. 为了使模型能够同时捕获全局和局部的特征信息,沿通道维度将输入特征进行划分. 首先计算全局建模模块所需要的通道数,可通过将输入特征通道数C除以自注意力头数N_h得到,其余通道则被分配给局部建模模块. 因此,将输入特征沿通道维度进行划分,分别为$F_\alpha \in \mathbb{R}^{H\times W\times C_1}$和$F_\beta \in \mathbb{R}^{H\times W\times C_2}$,$C_1$表示为局部信息增强模块分配的通道数,而$C_2$是用于全局信息建模的通道数,$C_1 + C_2 = C$.

在局部建模模块中,输入特征F_α经过两个并行分支. 其中一个分支首先经过最大池化获取目标特征中更为明显的特征信息,并将该特征经过线性层进行特征激活,得到该分支输出特征$F_{\alpha 1} \in \mathbb{R}^{H\times W\times C_1}$. 虽然最大池化通常用于捕获全局信息,但它同样能够非常有效地在更小的局部范围内捕获显著特征. 在该模块的设计中,最大池化操作是为了从F_α这个局部特征子集中捕获更为显著的特征,因此,将最大池化纳入局部建模模块是基于上述逻辑的考量. 该过程具体如下:

$$F_{\alpha 1} = \text{Linear}(\text{MaxPool}(F_\alpha)). \tag{6}$$

局部建模模块的另一个分支首先将输入特征经过线性层进行激活,然后经过深度可分离卷积对局部特征进行增强,得到该分支输出特征$F_{\alpha 2} \in \mathbb{R}^{H\times W\times C_1}$. 该过程表示如下:

$$F_{\alpha 2} = \text{DWConv}(\text{Linear}(F_\alpha)). \tag{7}$$

全局建模模块采用多头自注意力机制对全局特征信息进行增强,该模块将F_β作为输入. 因自注意力计算具有较大的计算开销,故先将输入特征经过平均池化(AvgPool),再进行缩放点积注意力(scaled dot-product attention, SDPA),最后经过上采样(Upsample)操作得到输出特征$F_{\beta 1} \in \mathbb{R}^{H\times W\times C_2}$. 该过程可以表示为

$$F_{\beta 1} = \text{Upsample}(\text{SDPA}(\text{AvgPool}(F_\beta))). \tag{8}$$

将$F_{\alpha 1}$、$F_{\alpha 2}$和$F_{\beta 1}$沿通道维度进行拼接,得到拼接后的特征F_t,即

$$F_t = \text{Concat}(F_{\alpha 1}, F_{\alpha 2}, F_{\beta 1}). \tag{9}$$

最终将拼接后的特征送入特征融合模块的前馈网络中,得到最终的输出特征F_T. 该过程可以表示为

$$F_T = \text{Linear}(F_t + \text{Conv}(F_t)). \tag{10}$$

特征增强模块通过全局和局部信息建模对骨干网络输出特征进行优化,提供精细的局部特征,从而提升特征的表达能力. 该模块综合利用全局和局部信息,显著增强了Transformer骨干网络的输出特征的利用,进一步提高了目标跟踪算法的性能.

1.3 自适应历史帧选择策略

与传统孪生网络不同的是,本文所提出的算法FEHST中,输入图像除模板图像和搜索图像之外,还引入了历史帧图像,其图像大小与搜索图像相同. 在线跟踪阶段,历史帧图像来自于当前跟踪过程中的搜索图像,通过模板替换策略更新历史帧模板. 历史帧图像的引入,能够为跟踪模型提供潜在的目标状态信息,使跟踪模型能够有效地应对跟踪过程中目标形态变化等复杂挑战.

然而,如果不对更新的历史帧加以限制,无差别的历史帧更新对跟踪性能不仅没有提升,反而由于视频序列中目标形态的变化与初始状态差异较大,对跟踪性能产生负面影响. 常见的更新策略是采用定时更新或者设置质量判断分支. 定时更新策略往往不能应对各种跟踪场景中的复杂变化,而增加质量判断分支无疑增加了训练成本.

针对以上问题,本文提出自适应历史帧选择策略来筛选历史帧图像,符合阈值的图像将被更新. 自适应历史帧选择策略如下式所示:

$$\text{threshold} = \frac{1}{n}\sum_{i=1}^{n} s_i. \tag{11}$$

其中:n为样本数量;s_i为在线更新阶段中的一个样本值,即第i帧图像的置信度分数. 式(11)可解释为在线更新阶段计算更新阈值的方法,即将所有样本值相加并除以样本数量,如果当前帧分数高于阈值,则认为可靠,并将该帧图像进行更新.

自适应历史帧选择策略能够有效地筛选历史帧图像,以避免无差异更新对跟踪性能造成的影响. 此外,该策略还能够尽可能选择靠近当前跟踪状态的可靠参考图像,从而有效提升跟踪性能.

2 实验结果与分析

2.1 实验细节

本文算法在Ubuntu 18.04操作系统上实现,CPU为Intel(R) Xeon(R) Silver 4310 CPU @2.10 GHz,GPU为两块Nvidia GeForce RTX 4090,显存为24 G,深度学习框架为Pytorch 1.11.0,CUDA版本为11.6.

训练数据集来源于LaSOT、TrackingNet、GOT-10K数据集的训练部分和COCO数据集. 整体训练过程为100个周期,每个周期有400 000组训练样本,使用AdamW优化器进行优化,学习率为5e-4,权重衰减为1e-4.

本文构建两种不同的网络设置,分别为FEHST-Tiny和FEHST-Base,具体设置如下所示.

1) FEHST-Tiny.

骨干网络: Swin Transformer-Tiny;

模板图像大小: [112×112];

搜索图像 / 历史帧大小: [224×224];

$C = 384, N = 4$.

2) FEHST-Base.

骨干网络: Swin Transformer-Base;

模板图像大小: [112×112];

搜索图像 / 历史帧大小: [224×224];

$C = 512, N = 8$.

其中C和N分别表示特征提取网络输出特征的通道数和特征融合网络中的编码器数量. 在所有网络设置中,本文均使用骨干网络的第3阶段之后的输出特征进行特征融合.

此外,动态预测模块(DPM)中,稀疏化阶段设置为$S = 3, S$表示DPM阶段数量,即在每个骨干网络各个阶段之后设置DPM模块,每个DPM模块的稀疏率ρ沿用原始模块的渐进式设置,将其设置为$\rho = [0.9, 0.8, 0.7]$. 特征增强模块(FEM)中,自注意力头数N_h与网络中其他自注意力头数相同,均设置为$N_h = 8$. 除特别说明外,本文算法其他网络设置均采用默认设置.

2.2 消融实验

为了测试本文算法所提出模块对跟踪性能的影响,在LaSOT和GOT-10K数据集上,针对FEHST-Tiny进行消融实验,具体如表1所示. 消融实验具体训练设置与2.1节相同.

在表1中,✓表示使用该策略,没有✓表示不

表1 LaSOT和GOT-10K上不同模块性能对比

Baseline	HF	FEM	DPM	Thre.	LaSOT AUC	GOT-10K AO	FPS
✓					0.667	0.718	45
✓	✓				0.668	0.728	30
✓	✓			✓	0.676	0.730	31
✓	✓	✓		✓	0.685	0.738	26
✓	✓		✓	✓	0.684	0.736	28
✓	✓	✓	✓	✓	0.688	0.743	27

使用当前策略. 表 1 中第 1 行数据表示基准算法本地复现结果. 第 2 行数据表示为网络架构中增加历史帧 (historical frame, HF) 信息, 该策略在短时数据集 GOT-10K 上效果提升较为明显, 而在长时数据集 LaSOT 上的效果并不显著. 第 3 行数据表示在上一步的基础上, 增加自适应历史帧选择策略约束历史帧更新, 可以看出, 在增加该策略后, 选取可靠的历史帧能够有效提升跟踪性能. 第 4 行和第 5 行数据分别表示在当前架构下, 增加特征增强模块 (FEM) 和动态预测模块 (DPM) 以后的结果, 可以看出性能均有一定提升. 第 6 行数据表示本文所提出算法整体架构的实验结果, 由实验结果可知, 本文所提出的模块和网络架构能够有效提升跟踪器的跟踪精度和成功率, 并且能够保持实时性.

2.3　定性分析

为定性说明本文算法的优越性, 从 OTB100 中选择 5 个包含多种挑战的序列, 并将本文算法 (FEHST) 与近年流行的跟踪器 (TransT、 SwinTrack) 进行比较. 实验结果表明, 本文算法所提出的策略有效提升了跟踪性能.

2.4　定量分析

为评估本文算法的有效性, 在 LaSOT、TrackingNet、GOT-10K 和 OTB100 上进了行详细分析. 限于篇幅, OTB100 数据集的详细测试结果此处不予罗列.

2.4.1　LaSOT 数据集实验结果

LaSOT 是一个大规模长时跟踪数据集及评价标准, 包含 1 400 个视频序列, 其中训练集 1 120 个序列, 测试集 280 个序列, 平均每个序列 2 500 多帧. LaSOT 数据集的评价标准一般为成功率 (AUC)、精度 (P) 以及归一化精度 (P_{Norm}). 将本文算法与其他具有竞争力的算法相比, 实验结果如表 2 所示. 表 2 中: FEHST-Base* 表示遵循 GOT-10K 协议训练; GOT-10K 中加点结果表示未遵循一次性协议, 因此不参与比较; 最好的 3 个结果分别以加粗、下划线和虚线表示. 从实验结果可以看出, 本文算法 FEHST-Tiny 取得了 68.8 % 的成功率, 相较于基准算法的 Tiny 版本提高了 2.1 %. FEHST-Base 取得了 70.1 % 的成功率, 处于领先优势. 与 TransT、OSTrack-256 等基于 Transformer 的主流跟踪器相比, 本文算法在 3 个评价指标中的表现均较为优异, 体现了本文算法较强的竞争力.

表 2　LaSOT、TrackingNet、GOT-10K 定量分析结果

method	source	LaSOT			TrackingNet			GOT-10K		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AO	$\text{SR}_{0.5}$	$\text{SR}_{0.75}$
MDNet	CVPR2016	0.299	0.303	0.099	0.606	0.705	0.565	0.299	0.303	0.099
ATOM	CVPR2019	0.515	0.576	0.505	0.703	0.771	0.648	0.556	0.634	0.402
SiamRPN++	CVPR2019	0.517	0.616	0.325	0.733	0.800	0.694	0.517	0.616	0.325
DiMP	ICCV2019	0.611	0.717	0.492	0.740	0.800	0.694	0.611	0.717	0.492
SiamR-CNN	CVPR2020	0.649	0.728	0.597	0.812	0.854	0.800	0.649	0.728	0.597
SiamFC++	AAAI2020	0.544	0.623	0.547	0.754	0.800	0.705	0.595	0.695	0.479
Ocean	ECCV2020	0.611	0.721	0.473	–	–	–	0.611	0.721	0.473
PrDiMP	CVPR2020	0.598	0.688	0.608	0.758	0.816	0.704	0.634	0.543	**0.738**
TrDiMP	CVPR2021	0.671	0.777	0.583	0.784	0.833	0.731	0.671	0.777	0.583
TransT	CVPR2021	0.671	0.768	0.609	0.814	0.867	0.803	0.671	0.768	0.609
AutoMatch	ICCV2021	0.582	0.674	0.599	0.760	–	0.726	0.652	0.766	0.543
STARK	ICCV2021	0.688	0.781	0.641	0.820	0.869	–	0.688	0.781	0.641
KeepTrack	ICCV2021	0.671	0.772	0.702	–	–	–	–	–	–
SwinTrack-T	arXiv2021	0.667	0.758	0.706	0.808	0.855	0.779	0.690	0.781	0.624
SwinTrack-B	arXiv2021	0.696	0.786	0.741	0.825	0.870	0.804	0.694	0.780	0.643
ToMP	CVPR2022	0.685	0.792	0.735	0.815	0.864	0.789	–	–	–
CSWinTT	CVPR2022	0.662	0.752	0.709	0.819	0.867	0.795	0.694	0.789	0.654
AiATrack	ECCV2022	0.696	**0.800**	0.632	0.827	0.878	0.804	0.696	0.800	0.632
OSTrack-256	ECCV2022	0.691	0.787	**0.752**	**0.831**	0.878	**0.820**	0.710	0.804	0.682
CTTrack-B	AAAI2023	0.678	0.778	0.740	0.825	0.871	0.803	0.713	0.807	0.703
FEHST-Tiny	ours	0.688	0.778	0.730	0.820	0.867	0.799	0.743	0.850	0.695
FEHST-Base	ours	**0.701**	0.788	0.751	0.830	**0.879**	0.815	0.750	0.854	0.710
FEHST-Base*	ours	–	–	–	–	–	–	**0.714**	**0.817**	0.683

2.4.2　TrackingNet 数据集实验结果

TrackingNet 包含 30 312 个视频序列, 27 种目标类别, 是目前视觉跟踪任务中规模最大的数据集. 该数据集对训练集和测试集进行划分, 其中测试集包含 511 个测试序列. 测试结果提交官方评估服务器来评估结果, 评价标准采用与 LaSOT 类似的成功率 (AUC)、归一化精度 (P_{Norm}) 以及精度 (P). 从表 2 可以看出, 本文算法 FEHST-Tiny 在 3 个评价指标上的结果分别为 82 %、86.7 % 和 79.9 %, 相较于基准算法 SwinTrack-T 分别提升 1.2 %、1.2 % 和 2 %. 本文算法

FEHST-Base与其他主流跟踪算法相比,在成功率上取得了领先优势,在归一化精度和精度上也表现优异.

2.4.3 GOT-10K数据集实验结果

GOT-10K数据集包含560个类别,超过10 000个较短的视频序列. 该数据集遵循训练集与测试集之间对象类零重叠的一次性规则,测试结果提交官方评估服务器来评估结果. 本文所提出算法FEHST-Base*代表遵循GOT-10K协议进行训练和测试评估,取得了71.4%的平均重叠度(AO)分数,相对于基准算法提升了2%;$SR_{0.5}$和$SR_{0.75}$两个评价指标也取得了较为优秀的效果. 为公平比较,FEHST-Base*模型仅针对GOT-10K单一数据集进行了性能比较,与表2中所列其他算法在GOT-10K上结果的训练设置相同.

3 结 论

本文针对目标跟踪算法中历史帧信息的利用和目标特征表达等问题,提出了基于特征增强和历史帧选择的Transformer视觉跟踪算法FEHST. 首先引入动态预测模块,通过稀疏化策略提升自注意力机制的计算效率,并聚焦目标区域,消除背景干扰信息;其次,提出了特征增强模块,通过通道划分对特征分别进行局部信息建模和全局信息建模,有效结合局部建模和全局建模的优势,提高了特征的表达能力;最后,通过自适应历史帧选择策略提升了跟踪器对目标动态信息的关注,该策略能够筛选可靠的历史帧模板,提升了算法的鲁棒性. 大量实验表明,本文算法具有良好性能,并能以27 FPS的速度运行.

参考文献(References)

[1] Javed S, Danelljan M, Khan F S, et al. Visual object tracking with discriminative filters and Siamese networks: A survey and outlook[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(5): 6552-6574.

[2] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]. 2013 IEEE Conference on Computer Vision and Pattern Recognition. Portland, 2013: 2411-2418.

[3] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional Siamese networks for object tracking[C]. European Conference on Computer Vision. Cham: Springer, 2016: 850-865.

[4] 刘如浩, 张家想, 金辰曦, 等. 基于可变形卷积的孪生网络目标跟踪算法[J]. 控制与决策, 2022, 37(8): 2049-2055.
(Liu R H, Zhang J X, Jin C X, et al. Target tracking based on deformable convolution Siamese network[J]. Control and Decision, 2022, 37(8): 2049-2055.)

[5] 陈志旺, 王莹, 宋娟, 等. 特征响应权重自适应的IoU网络跟踪算法改进[J]. 控制与决策, 2022, 37(7): 1752-1762.
(Chen Z W, Wang Y, Song J, et al. Improvement of IoU network tracking with adaptive weighted characteristic responses[J]. Control and Decision, 2022, 37(7): 1752-1762.)

[6] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: Transformers for image recognition at scale[J/OL]. 2020, arXiv: 2010.11929.

[7] Chen X, Yan B, Zhu J W, et al. Transformer tracking[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, 2021: 8126-8135.

[8] Hou Z Q, Guo F, Yang X L, et al. Transformer visual object tracking algorithm based on mixed attention [J]. Control and Decision, 2024, 39(3): 739-748.

[9] Chen Z W, Sun Z B, Lv C H, et al. Tracking algorithm of Siamese network based on parallel multiple appearance features[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2023.0851.

[10] Fan H, Bai H X, Lin L T, et al. LaSOT: A high-quality large-scale single object tracking benchmark[J]. International Journal of Computer Vision, 2021, 129(2): 439-461.

[11] Müller M, Bibi A, Giancola S, et al. TrackingNet: A large-scale dataset and benchmark for object tracking in the wild[C]. European Conference on Computer Vision. Cham: Springer, 2018: 310-327.

[12] Huang L H, Zhao X, Huang K Q. GOT-10K: A large high-diversity benchmark for generic object tracking in the wild[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1562-1577.

[13] Liu Z, Lin Y T, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]. 2021 IEEE/CVF International Conference on Computer Vision. Montreal, 2021: 10012-10022.

[14] Lin L, Fan H, Zhang Z, et al. SwinTrack: A simple and strong baseline for transformer tracking[J/OL]. 2021, arXiv: 2112.00995.

作者简介

侯志强(1973−), 男, 教授, 博士生导师, 从事图像处理、计算机视觉等研究, E-mail: hzq@xupt.edu.cn;

杨晓麟(1997−), 男, 硕士生, 从事视觉目标跟踪算法的研究, E-mail: yxlxupt@126.com;

马素刚(1982−), 男, 高级工程师, 硕士生导师, 从事计算机视觉、机器学习等研究, E-mail: msg@xupt.edu.cn;

王云龙(2000−), 男, 硕士生, 从事视觉目标跟踪算法的研究, E-mail: wangyl@stu.xupt.edu.cn;

余旺盛(1985−), 男, 副教授, 博士, 从事计算机视觉、模式识别等研究, E-mail: xing_fu_yu@sina.com;

王昀琛(1991−), 女, 讲师, 博士, 从事图像分割、深度学习等研究, E-mail: wangyunchen@lzb.ac.cn.