计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(7):163−170 [doi: 10.15888/j.cnki.csa.009130]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于多支路的孪生网络目标跟踪①

谢斌红, 于如潮

(太原科技大学 计算机科学与技术学院, 太原 030024)
通信作者: 于如潮, E-mail: s20202011062@stu.tyust.edu.cn

摘　要: 为有效解决目标跟踪在面对大尺度形变、完全遮挡、背景干扰等复杂场景时出现漂移或者跟踪丢失的问题, 本文提出了一种基于多支路的孪生网络目标跟踪算法 (SiamMB). 首先, 通过增加邻近帧支路的网络鲁棒性增强方法以提高对搜索帧中目标特征的判别能力, 增强模型的鲁棒性. 其次, 融合空间注意力网络, 对不同空间位置的特征施加不同的权重, 并着重关注空间位置上对目标跟踪有利的特征, 提升模型的辨别力. 最后, 在 OTB2015 和 VOT2018 数据集上的进行评估, SiamMB 跟踪精度和成功率分别达到了 91.8% 和 71.8%, 相比当前主流的跟踪算法取得了良好的竞争力.

关键词: 目标跟踪; 孪生网络; 邻近帧支路; 鲁棒性增强; 空间注意力网络

引用格式: 谢斌红,于如潮.基于多支路的孪生网络目标跟踪.计算机系统应用,2023,32(7):163–170. http://www.c-s-a.org.cn/1003-3254/9130.html

Siamese Network Target Tracking Based on Multiple Branches

XIE Bin-Hong, YU Ru-Chao

(College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: In order to effectively solve the problem of target tracking drift or loss in the face of large-scale deformation, complete occlusion, background interference, and other complex scenes, a multi-branch Siamese network target tracking algorithm (SiamMB) is proposed. First, the method of enhancing the network robustness of adjacent frame branches is used to improve the discrimination ability of target features in the search frame and strengthen the robustness of the model. Secondly, the spatial attention network is fused, and different weights are applied to the features of different spatial positions. In addition, the features that are beneficial to target tracking in spatial positions are emphasized, so as to improve the discriminability of the model. Finally, evaluation is performed on OTB2015 and VOT2018 datasets, and the results show that the tracking accuracy and success rate of SiamMB reach 91.8% and 71.8%, respectively, which makes SiamMB more competitive than the current mainstream tracking algorithms.

Key words: target tracking; Siamese network; adjacent frame branch; robustness enhancement; spatial attention network

目标跟踪一直以来都是一项具有挑战性的工作[1], 也是计算机视觉领域的热点之一, 受到了学者的广泛关注. 目前, 目标跟踪广泛应用于视频图像、智能交通、行为分析等领域[2]. 尽管近年来跟踪技术取得了长足的进步, 许多高效算法陆续被提出, 但目标跟踪任务仍面临着目标尺度形变大、相互遮挡和相似背景干扰等诸多挑战, 使其仍然成为当前研究热点[3].

孪生网络目标跟踪算法作为当前主流算法之一, 该算法将目标跟踪任务建模为一种相似性度量的问题, 其通过目标模板帧与搜索帧进行相似度计算, 得到一张响应图, 然后根据响应图的得分判断目标的位置信息. 这种算法的优点是网络结构简单, 算法速度较高[4].

① 基金项目: 山西省基础研究计划 (20210302123216); 吕梁市引进高层次科技人才重点研发项目 (2022RC08)
收稿时间: 2022-12-07; 修改时间: 2023-01-06; 采用时间: 2023-01-16; csa 在线出版时间: 2023-05-19
CNKI 网络首发时间: 2023-05-22

然而, 由于孪生网络目标跟踪算法仅使用第 1 帧的特征信息与搜索帧进行相似度匹配, 因此当目标面对复杂场景时, 目标跟踪就会受到严重影响, 从而降低模型的跟踪性能[5].

针对上述问题, 本文提出了一种基于多支路的孪生网络目标跟踪算法 (Siamese network tracker based on multiple branches, SiamMB), 其贡献如下.

1) 增加邻近帧支路, 提取与搜索帧更加接近的目标信息, 提高了跟踪器对目标的判别能力, 降低了目标在遭受到遮挡、形变等挑战时跟踪失败的风险.

2) 融合了空间注意力机制, 对不同空间位置的特征施加不同的权重, 从而获得更具有判别能力的特征.

1 相关工作

目前, 目标跟踪算法主要分为基于相关滤波的跟踪算法和基于深度学习的跟踪算法[6].

早期的单目标跟踪算法以相关滤波为主. 2010 年, Bolme 等人首次将相关滤波应用于目标跟踪领域, 并提出最小输出误差平方和跟踪算法 MOSSE[7], 该算法采用灰度特征训练目标区域的外观模型, 并利用离散傅里叶变换将目标与所有候选区域之间的相似度计算转换为频域, 大大提高了目标跟踪速度. 2012 年, Heriques 等人[8] 提出了基于 MOSSE 的循环跟踪核检测结构 CSK, 该算法通过训练样本循环矩阵, 大大增加了训练样本的数量. 在此基础上, KCF 算法[9] 采用了多通道的 HOG 特征取代了 CSK 的灰度特征, 提高了算法的鲁棒性. 此外具有代表性的相关滤波算法还有 HCFT[10] 等. 虽然上述算法在目标跟踪方面取得了良好的效果, 但缺乏对目标尺寸变化的考虑, 在面对遮挡、形变等复杂场景时容易受到外界环境的干扰, 导致跟踪的精度与成功率偏低[3].

与基于相关滤波的目标跟踪方法相比, 基于深度学习的目标跟踪算法在性能上有很大的提高, 已成为当前主流方法[11]. Wang 等人[12] 于 2013 年提出了首个将深度学习成功应用于目标跟踪领域的算法 DLT (deep learning tracker), 该算法使用堆叠去噪自动编码器来学习大图像数据集中的通用图像特征作为辅助数据, 然后将学到的特征传输到在线跟踪任务, 并且 DLT 利用了多个非线性变换, 优化了跟踪的性能. 随着深度学习的发展, 基于孪生网络的目标跟踪算法因其强大的性能成为目标跟踪领域的主流算法. SINT[13] 是第 1 个将孪生网络用于目标跟踪领域的算法, 其简单地将跟踪过程视为一个相似度学习问题, 寻找目标在每个搜索帧中的最优匹配响应, 为目标跟踪开辟了新的研究方向. 2016 年, Bertinetto 等人提出了全卷积孪生跟踪算法 SiamFC[14], 其将 AlexNet 作为特征提取网络, 简化了相似度计算过程, 跟踪实时性也得到了提高. SiamRPN[15] 将区域建议网络应用于孪生网络, 通过分类和回归分支来提高跟踪精度. 2019 年, SiamRPN++ 算法[16] 在 SiamRPN 的基础上, 利用 ResNet 进行特征提取, 提出了一种空间感知采样策略, 实现了多层融合, 进一步提高了跟踪精度. SiamGAT[17] 利用图注意力机制将目标信息从模板特征传播到搜索特征, 同时提出了一种目标感知区域选择机制, 以适应不同对象的长宽比变化.

2 基于多支路的孪生网络目标跟踪

2.1 网络结构

本文提出的目标跟踪算法 SiamMB 是在 SiamRPN 的基础上改进的, 如图 1 所示, 图中 SAM 为空间注意力模块, ⊗表示点乘操作, ⊕表示特征加权融合, ★表示互相关运算, 虚线框为本文改进的模块. 网络结构由 3 个部分组成.

1) 特征提取网络, 其中模板帧图像 Z 和搜索帧图像 X 利用 ResNet50 网络进行特征提取, 邻近帧图像 P 由大小为 5×5 的卷积核进行卷积.

2) 空间注意力网络, 输入来自特征提取网络的特征图, 对目标特征进行聚焦, 增强特征之间的判别性.

3) 分类和回归网络, 这一部分由区域提议网络 (RPN 网络) 实现. 其中分类网络用于区分目标是前景或背景, 回归网络用于确定目标的大小.

2.2 增加邻近帧支路的网络鲁棒性增强方法

在目标跟踪中, SiamRPN 网络仅使用模板帧图像的目标信息, 在面对遮挡和形变等复杂场景时鲁棒性不足, 从而增加了目标丢失的风险[3], 但是目标在前后两帧中的尺寸和空间位置通常不会发生巨大的变化[18], 受此启发, 本文在 SiamRPN 网络的基础上, 对邻近帧跟踪得到的结果图进行裁剪, 提取目标所在区域, 再进行缩放, 这时我们已经得到与搜索帧目标相似的邻近帧图像 P, 将其作为我们网络的邻近帧支路, 提取更加接近搜索帧目标的特征信息, 以增强网络在复杂场景下的鲁棒性.

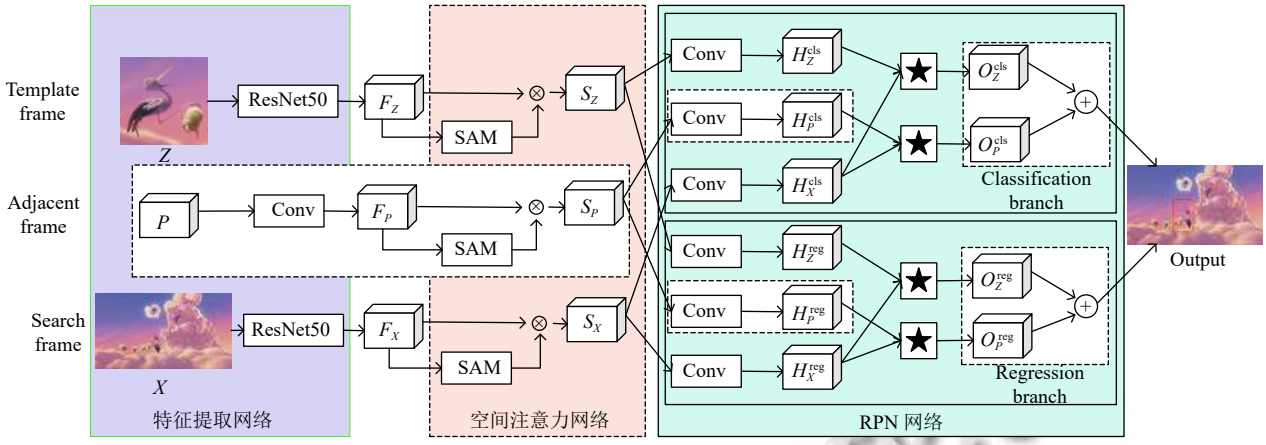


图 1 SiamMB 的网络结构图

首先, 利用 ResNet50 网络对模板帧和搜索帧图像进行特征提取, 分别得到特征图F_Z和F_X:

$$F_Z=\varphi(Z),\ F_Z\in\mathbb{R}^{16\times16\times256} \tag{1}$$

$$F_X=\varphi(X),\ F_X\in\mathbb{R}^{32\times32\times256} \tag{2}$$

其中, φ表示利用特征提取网络提取特征. 然后利用卷积操作对邻近帧图像 P 进行特征提取, 得到特征图F_P, 计算公式为:

$$F_P=\mathrm{Conv}^{5\times5}(P),\ F_P\in\mathbb{R}^{16\times16\times256} \tag{3}$$

其中, Conv 表示卷积操作, 卷积核大小为 5×5. 然后将F_Z、F_P和F_X分别输入到空间注意力网络, 抑制无用信息, 突出目标, 得到特征图S_Z、S_P和S_X, 最后再将其输入到 RPN 网络中进行分类和回归. 在分类网络中, 3 个特征图利用卷积操作提取特征, 改变通道数, 得到H_Z^{cls}、H_P^{cls}和H_X^{cls}, 计算公式为:

$$H_Z^{\mathrm{cls}}=\mathrm{Conv}^{3\times3}(S_Z) \tag{4}$$

$$H_P^{\mathrm{cls}}=\mathrm{Conv}^{3\times3}(S_P) \tag{5}$$

$$H_X^{\mathrm{cls}}=\mathrm{Conv}^{3\times3}(S_X) \tag{6}$$

其中, H_Z^{cls}, $H_P^{\mathrm{cls}}\in\mathbb{R}^{8\times8\times(2k\times256)}$, $H_X^{\mathrm{cls}}\in\mathbb{R}^{32\times32\times256}$, 这里 $2k$ 代表通道数, 分别对应于 k 个 anchors 的前景和背景. 然后将H_Z^{cls}和H_P^{cls}分别与H_X^{cls}按式 (7) 和式 (8) 做互相关运算得到O_Z^{cls}和O_P^{cls}. 最后, 再按式 (9) 进行加权融合, 回归网络中也是同样.

$$O_Z^{\mathrm{cls}}=H_Z^{\mathrm{cls}}*H_X^{\mathrm{cls}} \tag{7}$$

$$O_P^{\mathrm{cls}}=H_P^{\mathrm{cls}}*H_X^{\mathrm{cls}} \tag{8}$$

$$R_{\mathrm{cls}}=\lambda O_Z^{\mathrm{cls}}+(1-\lambda)O_P^{\mathrm{cls}} \tag{9}$$

其中, R_{cls}为分类分支的输出响应图, λ为权重系数.

2.3 空间注意力网络

为了能从复杂的背景下区分出跟踪目标, 需要对目标中的特征进行聚焦, 增加特征之间的判别性[19]. 因此, 本文引入空间注意力机制网络, 突出特征图中重要的目标区域, 抑制无用的背景信息. 空间注意力网络如图 2 所示.

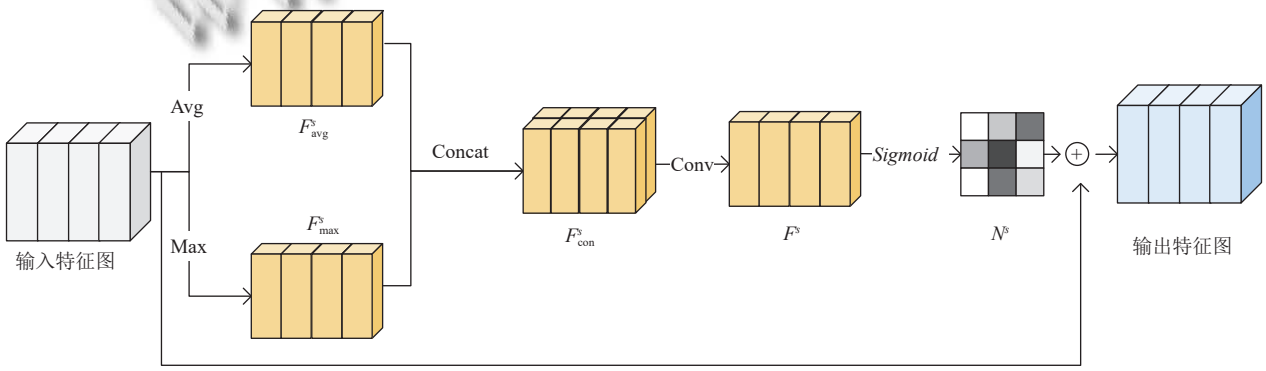


图 2 空间注意力网络结构图

在模板帧支路中, 将F_Z输入空间注意力机制网络, 然后经过最大池化和平均池化操作, 分别得到$F^s_{\text{avg}} \in \mathbb{R}^{16\times16\times1}$和$F^s_{\max} \in \mathbb{R}^{16\times16\times1}$, 再进行 concat 操作:

$$F^s_{\text{con}} = [F^s_{\text{avg}}; F^s_{\max}],\ F^s_{\text{con}} \in \mathbb{R}^{16\times16\times2} \tag{10}$$

接下来使用大小为 7×7 的卷积核进行卷积, 生成空间注意力图F^s:

$$F^s = \text{Conv}^{7\times7}(F^s_{\text{con}}),\ F^s \in \mathbb{R}^{16\times16\times1} \tag{11}$$

再通过 *Sigmoid* 函数进行归一化, 得到$N_s \in \mathbb{R}^{16\times16\times1}$:

$$N_s = Sigmoid(F^s) \tag{12}$$

N_s是可以控制空间分布的权重系数, 最后再将N_s与F_Z点乘, 得到$S_Z \in \mathbb{R}^{16\times16\times256}$:

$$S_Z = N_s \otimes F_Z \tag{13}$$

同样, 邻近帧支路和搜索帧支路分别将F_P和F_Z输入到空间注意力网络, 可以得到$S_P \in \mathbb{R}^{16\times16\times256}$和$S_X \in \mathbb{R}^{32\times32\times256}$.

2.4 损失函数

本文提出的算法训练过程中, 总损失计算公式为:

$$L = L_{\text{cls}} + \alpha L_{\text{reg}} \tag{14}$$

其中, α是平衡两种损失的加权因子, 本文设定为 1.5. L_{cls}为用于分类的交叉熵损失 (CrossEntropyLoss), 具体计算过程为:

$$L(y_i, p_i) = -(y_i \cdot \log(p_i) + (1-y_i)\cdot(1-p_i)) \tag{15}$$

其中, y_i表示第 i 个样本为目标或者背景, 值分别为 1 或者 0, p_i表示第 i 个样本的概率得分, 取值范围为 [0, 1].

若样本总数为N_{cls}, 则分类任务的损失计算值为:

$$L_{\text{cls}} = \frac{1}{N_{\text{cls}}}\sum_{i=1}^{N_{\text{cls}}} L(y_i, p_i) \tag{16}$$

L_{reg}为回归分支损失, 首先将回归距离归一化得到:

$$\begin{cases} \delta[0] = \dfrac{T_x - O_x}{O_w},\ \delta[1] = \dfrac{T_y - O_y}{O_h} \\ \delta[2] = \ln\dfrac{T_w}{O_w},\ \delta[3] = \ln\dfrac{T_h}{O_h} \end{cases} \tag{17}$$

其中, (O_x, O_y, O_w, O_h)表示锚框的中心点坐标和大小, (T_x, T_y, T_w, T_h)表示真实框的中心坐标和大小, 然后通过$SmoothL_1$损失计算, 如式 (18) 和式 (19):

$$SmoothL_1(x,\sigma) = \begin{cases} 0.5\sigma^2 x^2, & |x| < \dfrac{1}{\sigma^2} \\ |x| - \dfrac{1}{2\sigma^2}, & |x| \geqslant \dfrac{1}{\sigma^2} \end{cases} \tag{18}$$

$$L_{\text{reg}} = \sum_{i=0}^{3} SmoothL_1(\delta[i], \sigma) \tag{19}$$

其中, σ代表输入数据的标准差.

3 实验与分析

3.1 实验环境与参数设置

实验环境: 本文算法是使用 Python 版本为 3.6.5 和 PyTorch 1.8.1 框架实现的. 在 CentOS 7 操作系统上, 使用 CPU 型号为 Intel(R) Xeon(R) Bronze 3204 CPU @ 1.90 GHz, GPU 为 GeForce RTX 3080 Ti 进行实验.

训练数据集: 首先将 ResNet50 在 ImageNet 数据集上进行预训练, 然后将预训练后的 ResNet50 在 ImageNet VID 和 GOT-10k 数据集上进行迭代训练. 其中 ImageNet VID 共有 3 862 个视频片段用于训练, GOT-10k 包含了 560 类常见目标和 87 类运动模式.

参数设置: 本文训练周期为 300 个 epoch, 每批次训练图片数量 (batch size) 为 10 张, 其中前 130 个 epoch 学习率为 0.001, 后 170 个 epoch 的学习率衰减至 0.000 01. 使用随机梯度下降法 (stochastic gradient descent, SGD) 优化网络, 动量设置为 0.9. 为了更好地应对目标尺度变化, anchor 设置为 0.33、0.5、1、2、3 共 5 种长宽比.

3.2 数据集介绍

3.2.1 OTB2015 数据集

OTB2015[20] 是单目标跟踪领域最常用的基准之一, 包含了目标跟踪中常见的难点, 包括光照变化、尺度变化、遮挡、形变、运动模糊等 11 个视频属性, 共 100 个跟踪视频序列, 评价指标如下.

1) 跟踪精度: 计算模型跟踪预测的中心点和目标真实位置的中心点之间的欧氏距离:

$$\varepsilon = \sqrt{(x_p - x_g)^2 + (y_p - y_g)^2} \tag{20}$$

其中, (x_p, y_p)为模型跟踪预测的中心点, (x_g, y_g)为目标真实位置的中心点, ε是两个中心点之间的欧氏距离, ε小于设定阈值 T 的视频序列帧数与总帧数的比值即为跟踪精度 (*Precision*):

$$Precision = \frac{P}{N_F} \tag{21}$$

其中, P 代表ε小于设定阈值的视频序列帧数, N_F代表总帧数. 本文中 T 设定为 20.

2) 成功率: 计算模型跟踪预测的目标边界框与真实边界框间的重叠率, 也称为交并比, 计算公式为:

$$IoU = \frac{|R_p \cap R_g|}{|R_p \cup R_g|} \tag{22}$$

其中, R_p表示模型跟踪预测的目标边界框面积, R_g表示跟踪目标的真实边界框面积. 当图像帧的重叠率大于给定的阈值 t 时, 则该帧被视为成功的, 成功的帧数占

总帧数的百分比即为成功率 (*Success*):

$$Success = \frac{S}{N_F} \times 100\% \tag{23}$$

其中, S 代表成功的帧数, N_F代表总帧数. 本文中 t 设定为 0.5.

3.2.2 VOT2018 数据集

VOT2018 数据集[21] 由 VOT (visual object tracking) 挑战赛于 2018 年提出, 包含 60 个不同类别的序列, 评价指标包括: 跟踪准确率 (*Accuracy*), 鲁棒性 (*Robustness*) 和期望平均重叠率 (*EAO*).

1) 跟踪准确率: 用来评价跟踪器跟踪目标的准确度, 数值越大, 准确度越高, 由真实框与预测框的 *IoU* 来定义, 第 t 帧的准确率定义为:

$$\phi_t = \frac{|A_t \cap A_{gt}|}{|A_t \cup A_{gt}|} \tag{24}$$

其中, A_t表示第 t 帧预测框; A_{gt}表示第 t 帧真实框, 而对于整个视频序列的准确率可以定义为:

$$Accuracy = \frac{1}{N_{\text{valid}}} \sum_{t=1}^{N_{\text{valid}}} \phi_t \tag{25}$$

其中,ϕ_t表示第 t 帧的准确率; N_{valid}表示有效视频帧的数量.

2) 鲁棒性: 表示跟踪过程中的错误率, 重叠率为 0 即为跟踪失败, 数值越小, 表面鲁棒性越强, 具体定义为:

$$Robustness = \frac{F}{N_F} \tag{26}$$

其中,F 表示跟踪失败帧数;N_F表示数据集中含有的总帧数.

3) 期望平均重叠率: 是准确率和鲁棒性的综合评估指标, 将所有序列按长度分类, 相同长度的序列测得的准确率取平均. 计算公式为:

$$\widehat{\phi}(N_s) = \frac{1}{M} \sum_{t=1}^{M} \left(\frac{1}{N_s} \sum_{i=1}^{N_s} \phi_i \right) \tag{27}$$

其中, $\widehat{\phi}(N_s)$表示长度为N_s的视频序列的平均准确率, ϕ_i代表第 i 帧准确率, M 为长度N_s视频序列的个数, 然后在$[N_{lo} : N_{hi}]$长度范围上对所有长度序列的准确率平均值进行平均, 得到期望平均重叠率:

$$EAO = \frac{1}{N_{hi} - N_{lo}} \sum_{N_s - N_{lo}}^{N_{hi}} \widehat{\phi}(N_s) \tag{28}$$

3.3 定量分析实验

为了验证本文算法的性能, 在 OTB2015 和 VOT2018 数据集上对 SiamMB 进行了评估, 并与近年来的一些主流跟踪算法进行了比较.

3.3.1 OTB2015 数据集实验分析

式 (9) 中权重参数λ的取值非常重要, 我们通过在 OTB2015 数据集上进行实验确定, 本文赋予模板帧分支响应得分更大的权重, $\lambda \in [0.6, 1)$. 从表 1 可以直观地看出, SiamMB 在λ取 0.7 时, 跟踪精度和成功率都达到了最高值, 所以本文后续实验都在λ取 0.7 基础上进行.

表 1 λ 取不同值时的跟踪结果

λ	*Precision*	*Success*
0.60	0.893	0.705
0.65	0.896	0.707
0.70	**0.918**	**0.718**
0.75	0.897	0.710
0.80	0.887	0.691
0.85	0.872	0.672
0.90	0.866	0.657
0.95	0.853	0.646

在确定 λ 取值后, 为了验证 SiamMB 的性能, 本文选取了 6 个主流算法在 OTB2015 数据集上进行了对比. 对比算法分别是: SiamFC, SiamRPN, DaSiamRPN, SiamFC++, SiamCAR, SiamGAT. 实验结果如表 2 所示, 本文提出的 SiamMB 在对比主流跟踪器中取得了良好的竞争力, 与 SiamGAT 算法相比, 跟踪精度和成功率分别提高了 0.4% 和 0.8%, 与 SiamRPN 相比, 提升效果明显, 跟踪精度和成功率分别提高了 7.1% 和 7.9%, 证明本文算法的改进能有效地提高跟踪的鲁棒性, 提升跟踪的性能.

表 2 不同算法在 OTB2015 数据集上的 *Precision* 和 *Success*

Tracker	*Precision*	*Success*
SiamFC	0.772	0.584
SiamRPN	0.847	0.639
DaSiamRPN	0.880	0.659
SiamFC++	0.907	0.684
SiamCAR	0.910	0.698
SiamGAT	0.914	0.710
SiamMB (ours)	**0.918**	**0.718**

3.3.2 VOT2018 数据集实验分析

我们将提出的 SiamMB 算法在 VOT2018 数据集上进行实验, 并与 SiamFC, SiamRPN, DSiam, SiamVGG, SiamMask, SiamGAT 这 6 种跟踪算法进行比较. 同时我们比较了不同跟踪算法的 3 个评价指标: *Accuracy*、*Robustness* 和 *EAO*, 细节如表 3 所示, 我们可直观地看出, SiamMB 在 *Accuracy*、*EAO* 值上分别比 SiamRPN 提高了 0.3%、2.2%, *Robustness* 值降低了 4.7%. 在 *Accuracy* 值与 SiamMask 相差不大的同时, *Robustness* 和 *EAO* 都优于其他算法, 表现出了良好的竞争力, 说明本文的改进大大提高了跟踪的准确性, 验证了 SiamMB 在面对复杂场景时, 能够展现出较好的尺度适应能力, 保证

算法的鲁棒性.

为了验证模型的通用性, 本文与相关滤波类目标跟踪中具有代表性的算法 KCF、SRDCF、C-COT、ECO、ASRCF、UPDT 在 OTB2015 与 VOT2018 数据集上进行了对比实验. 并分别用 *Precision*、*Success* 和 *Accuracy*、*Robustness*、*EAO* 评价指标进行分析. 具体实验结果如表 4 所示.

表 3　算法在 VOT2018 上的跟踪准确率、鲁棒性和期望平均重叠率

Tracker	*Accuracy*	*Robustness*	*EAO*
SiamFC	0.502	0.583	0.185
SiamRPN	0.584	0.274	0.381
DSiam	0.511	0.651	0.193
SiamVGG	0.531	0.314	0.285
SiamMask	**0.607**	0.280	0.381
SiamGAT	0.595	0.251	0.400
SiamMB (ours)	0.589	**0.229**	**0.405**

由表 4 可以直观看出, 本文算法在 OTB2015 数据集上成功率达到了最高值, 相比 UPDT 算法提高了 1.7%, 跟踪精度仅次于 UPDT 算法, 相比 ECO 算法提高了 1.5%. 在 VOT2018 数据集上, 本文算法鲁棒性与 ECO 算法不相上下, 其余评价指标均达到了最高值, *Accuracy* 与 *EAO* 值相比 UPDT 算法分别提高了 0.8%, 验证了本文算法有效提高了跟踪鲁棒性, 改善了跟踪的性能, 证明本文算法在目标跟踪领域具有良好的竞争力.

表 4　本文算法与相关滤波类目标跟踪算法对比

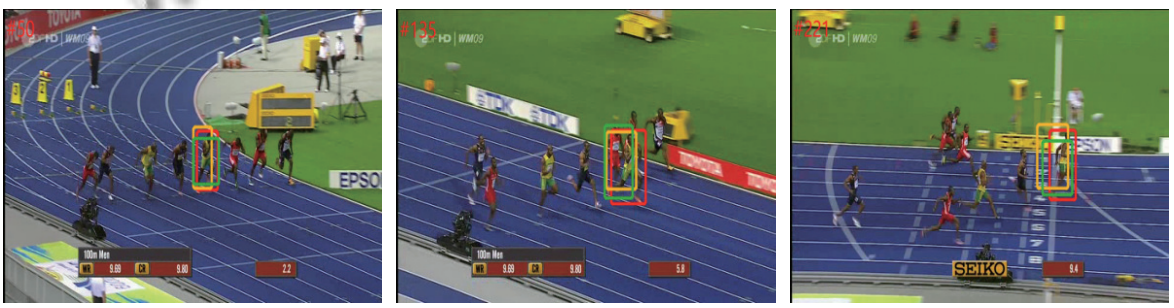
Tracker	OTB2015		VOT2018		
	Precision	*Success*	*Accuracy*	*Robustness*	*EAO*
KCF	0.705	0.507	0.510	0.571	0.202
SRDCF	0.808	0.618	0.537	0.476	0.241
C-COT	0.896	0.667	0.542	0.362	0.321
ECO	0.903	0.685	0.557	**0.227**	0.363
ASRCF	0.913	0.705	0.577	0.272	0.391
UPDT	**0.923**	0.701	0.581	0.243	0.387
SiamMB (ours)	0.918	**0.718**	**0.589**	0.229	**0.405**

3.4　定性分析实验

为了更加直观地展现出本文算法的优越性, 将 SiamMB 与 SiamGAT、SiamRPN 进行对比. 从 OBT2015 中选取 Basketball、Bolt、Girl2 和 Human2 这 4 组具有代表性场景的视频序列. 包含目标快速运动、尺寸变化、目标与背景相似和遮挡等多种复杂场景, 其中, Basketball 和 Bolt 视频序列主要针对快速运动和相似背景干扰情况下的实验验证, Human2 视频序列主要针对大尺度形变情况下的实验验证, Girl2 视频序列主要针对遮挡情况下的实验验证. 几种对比算法的跟踪效果如图 3 所示.

Basketball 和 Bolt: 目标快速移动以及相似背景干扰, 导致模型的判别性不足, SiamGAT 跟踪出现漂移, SiamRPN 跟踪丢失, 而本文算法利用空间注意力机制, 提高了模型的抗干扰能力和提取特征的能力, 表现出了良好的跟踪效果.

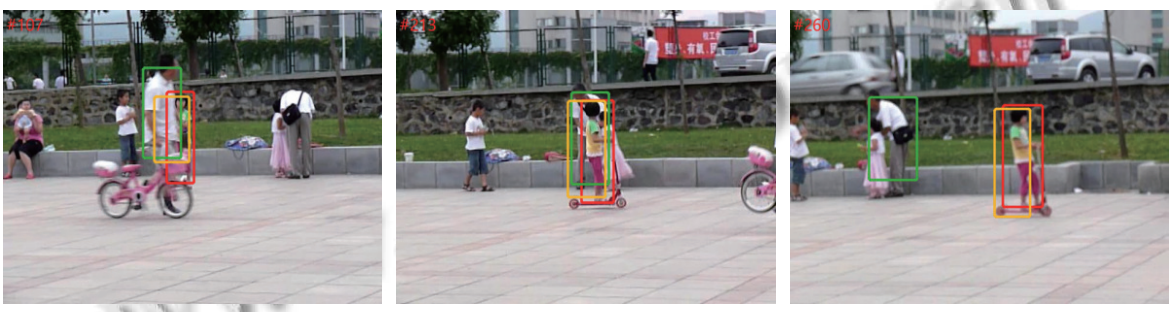
(a) Basketball



(b) Bolt

图 3　SiamMB、SiamGAT 和 SiamRPN 算法的跟踪结果

(c) Human2



(d) Girl2

— SiamMB — SiamGAT — SiamRPN

图 3 SiamMB、SiamGAT 和 SiamRPN 算法的跟踪结果 (续)

Human2: 目标发生大尺度变化, 其他算法都发生了不同程度上的跟踪漂移, 特别是在第 244 帧的时候, SiamRPN 跟踪丢失, 只有本文算法准确定位跟踪目标.

Girl2: 目标发生了完全遮挡或部分遮挡, 其中 SiamGAT 可以跟踪到目标, 但是跟踪过程中出现了漂移, 目标框与目标重叠率较低, SiamRPN 则在第 260 帧完全跟踪丢失, 而本文算法增加邻近帧支路, 提高了算法跟踪的鲁棒性, 所以跟踪效果最好.

通过以上定性实验分析, SimaRPN 在面对复杂场景时出现跟踪漂移甚至完全丢失, 无法完整的跟踪目标. SiamGAT 在跟踪丢失之后能够再次定位目标, 但是目标边界框与目标的重叠率较低, 整体性能较差. 而本文算法在面对以上复杂情况也能够保持良好的跟踪性能, 实现目标的准确定位.

3.5 消融实验

为了验证 SiamMB 改进模块的有效性, 本文在 OTB2015 数据集上进行了消融实验. 表 5 展示了不同模块组合对算法性能的影响.

表 5 中, 基线 (Baseline) 为 SiamRPN 模型, Model1 和 Model2 分别在 SiamRPN 的模板帧支路和搜索帧支路融合了空间注意力机制, Model3 则在 SiamRPN 的基础上增加了邻近帧支路, Model4 在 Model3 的邻近帧支路融合了空间注意力机制, Model5 在 SiamRPN 的基础上增加了邻近帧支路, 并且在 3 个支路中都融合了空间注意力机制.

表 5 OTB2015 数据集上的消融实验结果

方法	跟踪模型	*Precision*	*Success*
Baseline	SiamRPN	0.847	0.639
Model1	SiamRPN+SAM-z	0.854	0.642
Model2	SiamRPN+SAM-x	0.867	0.652
Model3	SiamRPN+PF	0.906	0.683
Model4	SiamRPN+PF+SAM-p	0.909	0.689
Model5	SiamRPN+PF+SAM-a	0.918	0.718

Model1 成功率和跟踪精度在基线模型的基础上分别提高了 0.03 和 0.07, Model2 成功率和跟踪精度在基线模型的基础上分别提高了 0.013 和 0.02, 证明了空间注意力机制能够增强特征之间的判别性, 为目标跟踪提供更加鲁棒的特征信息, 进一步证明了空间注意力机制对模型改进的有效性.

相比基线模型, Model3 的成功率和跟踪精度分别提高了 0.044 和 0.059, 反映了在跟踪过程中, 邻近帧支路能够提供更加具有判别性的特征信息, 增强算法在复杂场景下的鲁棒性, 证明了这种模型改进的有效性.

Model4 在 Model3 的基础上, 成功率和跟踪精度分别提高了 0.006 和 0.003, 证明本文两个改进能够有效结合起来.

Model5 则为 SiamMB 网络, 成功率在 Model4 的基础上提高了 0.029, 跟踪精度提高了 0.009, 与目前主流算法相比, 有较强的竞争力, 尤其在应对复杂环境时, 本文模型体现出了更强的鲁棒性.

经消融实验对比, Model5 的跟踪精度和成功率都达了最高值, 能达到最好的跟踪效果.

4 结论与展望

针对目标形变、遮挡及相似背景干扰等复杂场景下跟踪出现漂移甚至丢失的问题, 本文提出一种基于多支路的孪生网络目标跟踪算法 SiamMB, 增加邻近帧支路以增强网络的鲁棒性, 同时融合空间注意力机制, 有效地抑制无用的背景信息, 突出目标区域. 在 OTB2015 和 VOT2018 数据集上与主流的算法进行了对比, 本文算法相对于其他算法跟踪性能更加稳定.

该算法在跟踪精度与成功率上都有较大的提升, 但仍有进一步提升的空间, 后续我们将从以下两个方面提升算法的性能.

1) 将该算法改进模块与性能更好的孪生网络跟踪模型相结合来提升算法的整体性能.

2) SiamMB 模型在跟踪精度与区分相似干扰物、抗遮挡的能力大大提升, 但是跟踪过程中参考的目标信息仅有模板帧和邻近帧, 未来将加入更多的中间帧来提高对目标变化的适应能力.

参考文献

1 孟琭, 杨旭. 目标跟踪算法综述. 自动化学报, 2019, 15(7): 1244–1260. [doi: 10.16383/j.aas.c180277]

2 Zhao JG, Cao XY. Research on target tracking algorithm in occlusion scene. Journal of Physics: Conference Series, 2021, 1748(3): 032048. [doi: 10.1088/1742-6596/1748/3/032048]

3 程旭, 崔一平, 宋晨, 等. 基于时空注意力机制的目标跟踪算法. 计算机科学, 2021, 48(4): 123–129. [doi: 10.11896/jsjkx.200800164]

4 韩瑞泽, 冯伟, 郭青, 等. 视频单目标跟踪研究进展综述. 计算机学报, 2022, 45(9): 1877–1907. [doi: 10.11897/SP.J.1016.2022.01877]

5 章泉源, 王昱程, 陈曦, 等. 基于全卷积孪生网络的模糊目标跟踪. 武汉大学学报 (理学版), 2021, 67(5): 411–421. [doi: 10.14188/j.1671-8836.2021.0042]

6 陈志良, 石繁槐. 结合双模板融合与孪生网络的鲁棒视觉目标跟踪. 中国图象图形学报, 2022, 27(4): 1191–1203. [doi: 10.11834/jig.200660]

7 Bolme DS, Beveridge JR, Draper BA, *et al*. Visual object tracking using adaptive correlation filters. Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco: IEEE, 2010. 2544–2550. [doi: 10.1109/CVPR.2010.5539960]

8 Henriques JF, Caseiro R, Martins P, *et al*. Exploiting the circulant structure of tracking-by-detection with kernels. Proceedings of the 12th European Conference on Computer Vision. Florence: Springer, 2012. 702–715. [doi: 10.1007/978-3-642-33765-9_50]

9 Henriques JF, Caseiro R, Martins P, *et al*. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583–596. [doi: 10.1109/TPAMI.2014.2345390]

10 Ma C, Huang JB, Yang XK, *et al*. Robust visual tracking via hierarchical convolutional features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(11): 2709–2723. [doi: 10.1109/TPAMI.2018.2865311]

11 陈旭, 孟朝晖. 基于深度学习的目标视频跟踪算法综述. 计算机系统应用, 2019, 28(1): 1–9. [doi: 10.15888/j.cnki.csa.006720]

12 Wang NY, Yeung DY. Learning a deep compact image representation for visual tracking. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 809–817.

13 Tao R, Gavves E, Smeulders AWM. Siamese instance search for tracking. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 1420–1429.

14 Bertinetto L, Valmadre J, Henriques JF, *et al*. Fully-convolutional Siamese networks for object tracking. Proceedings of the 2016 European Conference on Computer Vision. Amsterdam: Springer, 2016. 850–865. [doi: 10.1007/978-3-319-48881-3_56]

15 Li B, Yan JJ, Wu W, *et al*. High performance visual tracking with Siamese region proposal network. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8971–8980.

16 Li B, Wu W, Wang Q, *et al*. SiamRPN++: Evolution of Siamese visual tracking with very deep networks. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 4277–4286.

17 Guo DY, Shao YY, Cui Y, *et al*. Graph attention tracking. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 9538–9547.

18 Wang NY, Shi JP, Yeung DY, *et al*. Understanding and diagnosing visual tracking systems. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 3101–3109.

19 刘嘉敏, 谢文杰, 黄鸿, 等. 基于空间和通道注意力机制的目标跟踪方法. 电子与信息学报, 2021, 43(9): 2569–2576. [doi: 10.11999/JEIT200687]

20 Wu Y, Lim J, Yang MH. Object tracking benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834–1848. [doi: 10.1109/TPAMI.2014.2388226]

21 Kristan M, Leonardis A, Matas J, *et al*. The sixth visual object tracking VOT2018 challenge results. Proceedings of the 2018 European Conference on Computer Vision (ECCV) Workshops. Munich: Springer, 2018. 3–53.

(校对责编: 孙君艳)