高技术通讯 2023 年 第33 卷 第8 期:802-814

doi:10.3772/j.issn.1002-0470.2023.08.003

基于参数自适应与模板更新的孪生网络跟踪算法①

陈志旺②*** 郭金华③* 吕昌昊*** 雷春明* 彭 勇****

(*燕山大学智能控制系统与智能装备教育部工程研究中心 秦皇岛 066004)

(**燕山大学工业计算机控制工程河北省重点实验室 秦皇岛 066004)

(***燕山大学河北省电力电子节能与传动控制重点实验室 秦皇岛 066004)

(****燕山大学电气工程学院 秦皇岛 066004)

摘 要 孪生网络跟踪算法在跟踪过程中网络参数固定,跟踪模板仅仅使用第1帧给定的目标,这导致算法的鲁棒性较差。为此,提出基于参数自适应(PA)与模板更新的孪生网络跟踪算法。首先,利用通道注意力和空间注意力对目标特征进行调整,提高网络对跟踪目标的关注度;其次,利用滤波器参数更新策略滤除背景的干扰,提高网络对当前目标的辨识能力;最后,增加与主网络平行的子网络,通过更新子网络的跟踪模板,使网络能适应目标的变化。在VOT 2018、VOT 2019 2个标准数据集上进行测试,期望重叠率(EAO)分别达到0.455和0.331,验证了本算法的有效性。

关键词 目标跟踪;孪生网络;模板更新;参数自适应(PA);注意力机制

0 引 言

目标跟踪是计算机视觉中的一个基础任务,具有重要的理论意义与应用价值,在车辆导航[1]、隐私保护[2]、疾病预防[3]等方面广泛应用。简而言之,目标跟踪旨在给定任意目标在视频序列中第1帧的位置的前提下,预测出给定目标在后续帧中的位置和大小。尽管目标跟踪算法已经取得了比较大的发展,但在光照变化、尺度变化、遮挡等情况下,实现稳健的跟踪仍然是具有挑战性的任务。

在视觉目标跟踪领域中,现有的跟踪算法大多基于孪生网络(Siamese network)框架,此类算法主要分为2类:含有预定义候选框的跟踪算法和不含预定义候选框的跟踪算法。

(1)含有预定义候选框的跟踪算法。最先提出基于孪生网络的跟踪算法是基于全卷积孪生网络的目标跟踪[4](fully-convolutional Siamese networks for object tracking, SiamFC),该算法使用孪生网络学习跟踪目标与搜索区域的相似性,从而将目标跟踪问题转换为在整个搜索区域上搜索目标问题。该算法首次将孪生网络用于目标跟踪,基于这种结构,衍生出一系列孪生网络跟踪算法。例如,在SiamFC算法的基础上引入目标检测算法[5]中区域候选网络的基于孪生区域候选网络的高性能目标跟踪算法(high performance visual tracking with Siamese region proposal network,SiamRPN)[6],该算法由用于前景-背景估计的分类网络和用于候选框修正的回归网络组成,使用可变宽高比的边界框来实现目标的定位与目标尺寸的估计,从而获得更加精准的边界框。至此之后,出现一批通过预定义候选框定位目标的孪生网络算法,基于干扰感知的孪生网络(视觉)目标跟踪算法(distractor-aware Siamese networks for visual object tracking, Da-SiamRPN)[7]是在SiamRPN

① 国家自然科学基金(61573305)和河北省自然科学基金(F2022203038,F2019203511)资助项目。

② 男,1978年生,博士,副教授;研究方向:多旋翼飞行控制,目标跟踪;E-mail:czwaaron@ysu.edu.cn。

③ 通迅作者,E-mail:1213918096@qq.com。

(收稿日期:2022-08-05)

基础上加入干扰感知模块,提高了算法的精度和辨识能力。基于深度网络的孪生网络跟踪算法(evolution of Siamese visual tracking with very deep networks, SiamRPN ++)[8]在 SiamRPN 的基础上使用更深的 ResNet50 网络代替原来的 AlexNet 网络,并且加入多层融合策略,使用深度交叉互相关操作代替 SiamRPN 中简单的互相关操作,带来了更高的跟踪性能。基于深度和宽度神经网络的目标跟踪算法(deeper and wider Siamese networks for real-time visual tracking, SiamDW)[9]分别在 SiamFC 和 SiamRPN 的基础上,通过在更深的残差网络和更宽的 Inception 网络中加入残差内部裁剪单元(cropping-inside residual ,CIR),消除了简单增加神经单元带来的定位精度下降和网络中的 Padding 带来的负面影响,进一步提高了算法的准确性和鲁棒性。能够进行目标分割的在线孪生网络跟踪(fast online object tracking and segmentation: a unifying approach, SiamMask)[10]将语义分割与目标跟踪相结合,在进行目标跟踪的过程中生成被跟踪目标的二进制掩膜,进而得到目标的边界框,大幅度提高了算法的精度。上述算法虽然取得了比较好的跟踪效果,但是需要人工设定 5 种不同宽高比的候选框,这导致在训练过程中正负样本不平衡,减缓了网络的训练速度;其次候选框宽高比是根据特定样本设定,导致跟踪网络的泛化性下降。

(2)不含预定义候选框的跟踪算法。为消除候选框宽高比设定不合适带来的影响。全卷积无锚框孪生网络目标跟踪算法(fully conventional anchor-free Siamese networks for object tracking, FCAF)[11]最先提出无候选框跟踪算法,该算法通过融合 ResNet50 特征提取网络第 3、4、5 层的输出,并且采用与 SiamRPN ++ 相同的网络分支实现目标跟踪。因舍去候选框,网络无法获得先验知识,该算法增加目标检测算法中的 Center-ness 分支[12]来估计目标的中心点,利用回归网络直接预测出目标边界框与锚点的距离。基于目标估计的精确目标跟踪算法(towards robust and accurate visual tracking with target estimation guidelines,SiamFC ++)[13]在 SiamFC 的基础上将特征提取网络改为 GoogleNet[14],并且采用与 FCAF 相同的分支结构来实现目标的定位,该算法提升了跟踪速率。自适应孪生网络视觉跟踪算法(Siamese box adaptive network for visual tracking, SiamBAN)[15]采用传统的分类与回归分支,通过不同的训练策略(正样本的选定方式)使回归分支可以直接预测出跟踪目标边界框与锚点的距离,精简了跟踪模型,证明孪生网络跟踪算法可以在没有先验知识的情况下,只采用分类与回归分支也可以实现目标跟踪。不同于传统算法中利用回归分支来直接预测目标的边界,关键点预测网络视觉跟踪算法(Siamese key point prediction network for visual object tracking, SiamKPN)[16]通过引入关键点预测头网络(keypoint prediction head)来生成粗略的跟踪目标热度图,再通过级联特征提取网络不同层的输出来逐步缩小热度图的范围,从而对目标的边界进行精准定位。随着自然语言处理领域的发展,计算机视觉领域尝试引入 Transformer[17]结构,由于其考虑到局部特征与全局特征的关系,使用自注意力结构为目标跟踪提供了新的解决方案。文献[18]根据 Transformer 结构设计出了新的特征融合模块,代替传统孪生网络跟踪算法中的互相关操作,使跟踪模板和当前帧可以进行全局计算,提高网络对跟踪目标的整体把握。文献[19]回归到跟踪的本质问题,根据 Transformer 中提出的框架重新设计了跟踪任务的回归和分类分支,利用编码解码结构完成特征融合,之后直接预测出目标的边界信息。

上述孪生网络跟踪算法在跟踪过程中网络参数固定,且仅仅使用第一帧给定的目标作为跟踪模板,或者直接更新原网络的跟踪模板,因此存在一定的局限性。其一,由于网络的参数在跟踪过程中是固定的,但是跟踪目标是任意的,可能存在网络未学习过的跟踪目标,这时网络的跟踪性能会严重下降。其二,文献[20]指出网络在跟踪时跟踪模板固定,将会导致其在被跟踪目标发生形变、运动模糊等具有挑战性的条件下无法跟踪上目标,回归分支预测目标边框的精度下降。并且,文献[21]直接根据目标框的分类来更新原模板并不有效。

针对以上基于孪生网络跟踪算法存在的问题,本文提出基于参数自适应与模板更新的孪生网络跟

踪算法。为获得更适用于当前跟踪目标的网络参数,使用参数自适应模块,通过更新滤波器参数,将背景噪声与目标区域分开,提高算法的判别能力;为应对目标的变化引入与主网络平行的子网络,通过不断更新子网络的跟踪模板来提高网络对目标变化的适应性。

1　基于参数自适应与模板更新的跟踪算法

本文跟踪算法整体框架如图 1 所示,主要包括特征提取模块、分类与回归模块、参数自适应模块和模板更新。

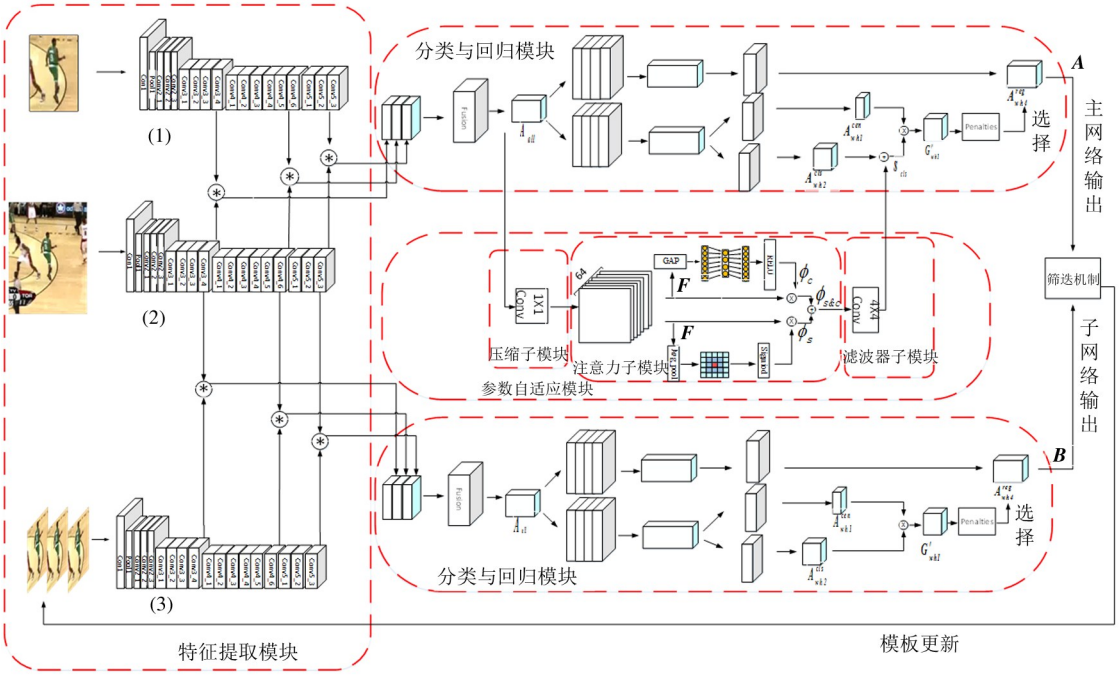


图 1　基于参数自适应与模板更新的孪生跟踪算法整体框图

1.1　特征提取模块

ResNet50 各层输出如图 2 所示,可以发现特征提取模块每层提取的目标特征各不相同,抽象程度逐渐提高:从图 2(a)可以看到目标的轮廓,几乎不包含特定目标的语义信息;从图 2(b)几乎看不到特定目标与背景的不同,目标湮没在背景中;从图 2(c)可以看到目标与背景逐渐分离,但是分离程度不太明显,具有浅层语义信息;从图 2(d)可以看到,相对于图 2(c)来说,目标与背景分离的程度变得更加明显,具有中层语意信息;从图 2(e)可以看到,网络更

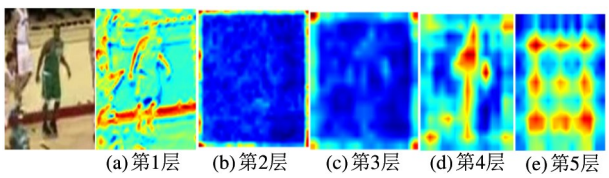


图 2　特征提取模块各层输出

倾向于提取目标的高级语义信息,所以第 5 层具有深层语义信息。

因此,特征提取模块第 1 层提取的是目标的通用特征,第 2 层提取的特征并没有将目标与背景分离,所以不选用第 1、2 层的输出特征。目标跟踪中,高级语义信息(第 4、5 层)在运动模糊、目标旋转、相似性干扰等具有挑战性的场景中比较重要,但是仅仅使用高级语义信息是不够的,随着神经网络深度的加深,特征图的分辨率大大降低,这时目标的位置信息丢失比较严重,因此需要浅层语义信息(第 3 层)来辅助完成目标的精准定位。为提高定位的精准度,在 SiamRPN ++ [8] 中使用 3、4、5 层输出;在无候选框跟踪算法中,FCAF、SiamBAN、SiamKPN 同样采用 ResNet50 网络的 3、4、5 层输出用于后续的跟踪任务。综上考虑,本文算法同样采用第 3、4、5 层的输出特征来实现目标的辨识与定位。在图 1 特征

提取网络中有 3 组 ResNet50 网络:第 1 组提取视频序列第 1 帧中给定跟踪目标的特征作为主网络跟踪模板;第 2 组提取检测帧中目标的特征;第 3 组提取的目标特征作为新的跟踪模板。1 ~ 3 组网络采用相同的网络结构和网络参数,而孪生网络跟踪算法的核心在于使用相同的特征提取网络将模板帧和检测帧映射到相同的特征空间,通过计算模板帧与检测帧特征的相似性实现定位跟踪目标。

1.2 分类与回归模块

分类与回归模块包含特征融合、回归分支、分类分支和得分惩罚 4 个部分。特征融合部分接收来自特征提取网络不同层的输出,实现不同层信息的融合;回归分支编码了目标边界框的回归量;分类分支为每个空间位置编码了其属于目标和背景的分类得分;得分惩罚通过对分类输出进行惩罚,间接对预测框的变化进行限制,实现最终目标框的计算。

由于舍去预定义候选框,本文使用锚点定位目标,其分布情况如图 3 中像素点所示。锚点计算公式如式(1)所示。

图 3 锚点位置示意图

$$\begin{cases} x = 27 + i \times 8 \\ y = 27 + i \times 8 \end{cases} \tag{1}$$

式中,x、y 代表锚点的横坐标和纵坐标,$i = \{1,2,3,\cdots,25\}$。如图 1 所示,特征融合部分具体操作可表示为

$$\boldsymbol{A}_{\text{all}} = \varphi_{11}(\boldsymbol{CONV}_3, \boldsymbol{CONV}_4, \boldsymbol{CONV}_5) \tag{2}$$

式中,φ_{11} 表示卷积核为 1 × 1 的卷积层,$\boldsymbol{CONV}_v$ 表示各层进行互相关之后的结果,$v = \{3,4,5\}$,其中互相关计算方式为 $\boldsymbol{CONV}_v = \boldsymbol{F}_{zv} * \boldsymbol{F}_{xv}$。式中,$\boldsymbol{F}_{zv}$、$\boldsymbol{F}_{xv}$ 分别为模板帧、检测帧第 $\boldsymbol{v}$ 层特征,* 为卷积操作。在文献[8]中,算法使用加权求和的方式对 ResNet50 网络第 3、4、5 层的输出进行融合,相对于文献[8]中对同一层特征的每个通道简单地赋予相同的权重,本文采用卷积操作进行融合,采用卷积融合可以对特征中每个通道的二维空间元素赋予不同的权重,得到更加精细的目标特征。

对于回归分支,特征 $\boldsymbol{A}_{\text{all}}$ 经过 4 层卷积之后,再次经过卷积核大小为 3 × 3 的卷积层,输出结果 $\boldsymbol{A}_{wh4}^{\text{reg}}$,$w$、$h$ 分别代表特征图的宽和高,4 个通道分别代表回归边界框相对于锚点的距离。

对于分类分支,特征 $\boldsymbol{A}_{\text{all}}$ 经过由 4 层卷积组成的卷积块,再次分为 2 个分支:分类子分支与中心度分支。对于分类子分支,对卷积块的输出使用卷积核大小为 3 × 3 的卷积层进行操作,得到输出结果 $\boldsymbol{A}_{wh2}^{\text{cls}}$,$w$、$h$ 代表分类得分图的宽和高,2 个通道分别代表该锚点属于前景和背景的分类得分;对于中心度分支,对卷积块的输出使用卷积核大小为 3 × 3 的卷积层进行操作,得到输出结果 $\boldsymbol{A}_{wh1}^{\text{cen}}$,$w$、$h$ 代表每个像素点属于目标中心得分图的宽和高,1 个通道代表锚点属于目标中心点的得分,得到中心度得分之后,取经过重新分配权重的分类子分支中代表锚点属于前景的得分 $\boldsymbol{S}_{\text{cls}}$ 与其作哈达玛积,得到每个锚点属于目标定位点的得分 $\boldsymbol{G}'_{wh1}$。这里中心度分支的作用是辅助分类分支对目标中心点进行更加精确地定位。将 $\boldsymbol{G}'_{wh1}$ 中得分最高的锚点作为跟踪目标的定位点,再根据与其对应的回归量预测出大致的预测边界框。

对于得分惩罚部分,由于相邻帧之间目标的边界框的尺寸大小变化和尺度变化很小,因此使用尺度和尺寸惩罚项对 25 × 25 个锚点的得分图进行重新排序,间接达到惩罚尺度和尺寸的目的。惩罚项 G_{penalty} 可表示为

$$G_{\text{penalty}} = \exp\{-k(G_{sc} - 1)\} \tag{3}$$

$$G_{sc} = \max\left(\frac{r}{r'}, \frac{r'}{r}\right) \times \max\left(\frac{s}{s'}, \frac{s'}{s}\right) \tag{4}$$

式中,k 为超参数;r 为当前帧的宽高比 $\frac{w_{\text{current}}}{h_{\text{current}}}$;$r'$ 表示上一帧的宽高比;s、s' 为当前帧与上一帧的目标框等效总长度,并且 s 满足

$$s = \sqrt{(w_{\text{current}} + p) \times (h_{\text{current}} + p)} \tag{5}$$

其中 $p = \frac{(w_{\text{current}} + h_{\text{current}})}{2}$。在 25 × 25 个锚点得分的基础上与式(3)的惩罚项相乘,得到新的得分图:

$$\boldsymbol{G}_{\text{new}} = G_{\text{penalty}} \times \boldsymbol{G}'_{wh1} \tag{6}$$

这里,“×”代表矩阵对应元素相乘。由于跟踪目标在相邻两帧间的位置变化不大,因而本文引入余弦窗($\boldsymbol{W}_{\text{cosine}}$)来抑制大的目标位移变化进而排除干扰物。

$$\boldsymbol{G}_{\text{final}} = \boldsymbol{G}_{\text{new}} \times (1 - k_{wi}) + \boldsymbol{W}_{\text{cosine}} \times k_{wi} \tag{7}$$

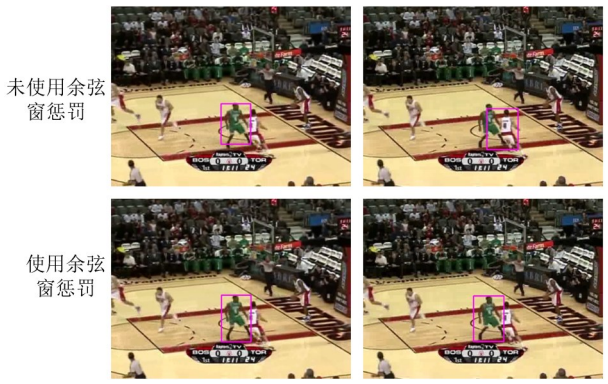
其中,k_{wi} 为超参数,用于决定余弦窗惩罚的程度。

基于式(7),找到得分图中得分最高的分数位置,并找到与其相对应的回归量,得到预测框。最后对预测边界框进行平滑处理,平滑公式为

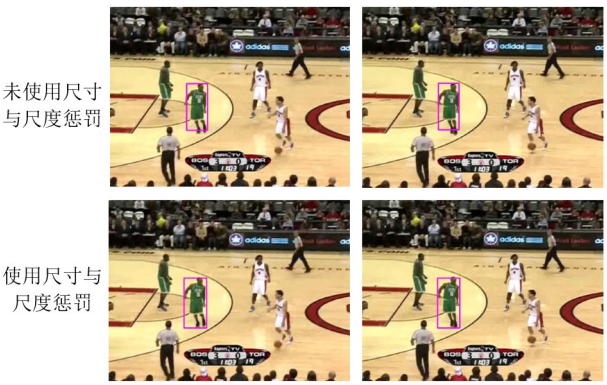
$$\begin{cases} w_{\text{final}} = lr \times w_{\text{current}} + (1 - lr) \times w_{\text{last}} \\ h_{\text{final}} = lr \times h_{\text{current}} + (1 - lr) \times h_{\text{last}} \end{cases} \tag{8}$$

式中,$lr = g_{\text{new}} \times \alpha_{lr}$,$\alpha_{lr}$ 为超参数,g_{new} 为 $\boldsymbol{G}_{\text{new}}$ 中的最高得分,w_{final}、h_{final} 分别为最终目标框的宽和高,w_{last}、h_{last} 分别为上一帧最终的目标框的宽和高。

余弦窗惩罚效果如图 4(a)所示,不加入余弦窗惩罚时,在发生目标遮挡的情况下,算法容易跟踪上干扰物,进行尺度惩罚之后,由于限制相邻帧目标位置预测的变化,提高了算法的鲁棒性;尺寸尺度惩罚效果如图 4(b)所示,不加入尺寸尺度惩罚时,预测的目标框可以随意变化,导致预测出的目标框无法完全框选目标,加入惩罚之后,相邻帧之间的目标框不会发生剧烈变化,提高了算法的精度。



(a) 余弦窗惩罚



(b) 尺寸尺度片罚

图 4　惩罚项作用示意图

1.3　参数自适应模块

因孪生网络采用完全离线的方式进行训练,并且仅在初始帧中学习目标的模板特征,这将导致网络难以适应剧烈的跟踪目标表观变化,从而降低目标跟踪的鲁棒性。因此一个可行的办法是设计一个参数自适应模块,在跟踪过程中进行较少样本的训练,以此来调整网络参数。

参数自适应模块如图 1 中所示,主要包括 3 个子模块:压缩子模块,注意力子模块,滤波器子模块。参数自适应模块的参数求解可看成一个优化问题,可通过求解以下优化目标来获取:

$$\begin{cases} L(w) = \min(\varphi_f + \varphi_p) \\ \varphi_f = \sum_{j=1}^{m} \gamma_j \cdot r_C(f(\boldsymbol{x}_j, \boldsymbol{w}_j), \boldsymbol{y}_j) \\ \varphi_p = \sum_{k=1}^{c} \lambda_k \| \boldsymbol{w}_k \|^2 \end{cases} \tag{9}$$

式中,$\boldsymbol{x}_j$ 为参数自适应模块的输入特征;m 为样本池的容量大小,f 为计算模块输出;$\boldsymbol{y}_j$ 为与训练样本对应的标签;$r_C(f(\boldsymbol{x}_j, \boldsymbol{w}_j), \boldsymbol{y}_j)$ 为计算每个空间位置的残差函数,γ_j 是每个样本对应的权重值,用于控制每个样本的影响程度,$\boldsymbol{w}_k$ 为模块中卷积层的权重;c 为卷积层的层数,这里 $c = 4$;λ_k 为对应的 $\boldsymbol{w}_k$ 的正则化系数。本文将式(9)中的 r_C 设置为 L_2 损失,对滤波器子模块的输出进行调整,进而生成特定于当前目标的特征。

$$r_C(f(\boldsymbol{x}_j; \boldsymbol{w}_j), \boldsymbol{y}_j) = \| f(\boldsymbol{x}_j, \boldsymbol{w}_j) - \boldsymbol{y}_j \|^2 \tag{10}$$

将式(9)重新定义为残差向量的平方范数形式 $L(w) = \sum_{l=1}^{m+c} \| \boldsymbol{g}_l \|^2$,这里 $g_j = \sqrt{\gamma_j}(f(\boldsymbol{x}_j; \boldsymbol{w}_j) - \boldsymbol{y}_j)$,其中 $j \in \{1, \cdots, m\}$,$\boldsymbol{g}_{m+k} = \sqrt{\lambda_k}\boldsymbol{w}_k (k = 1,2,3,4)$;至此,式(9)转化为正定二次型问题,本文沿用文

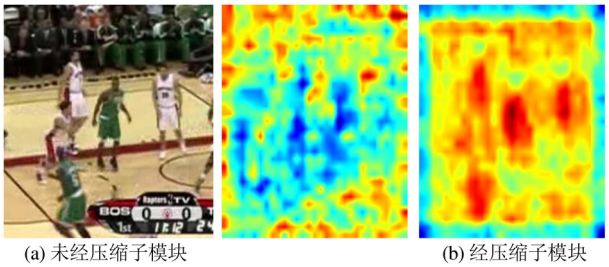
献[22]中的牛顿-高斯下降法实现损失函数的快速收敛。

1.3.1　压缩子模块

文献[23]利用自编码器模型实现输入的压缩与解压缩,自编码器包含编码器和解码器,自编码器通过神经网络学习输入数据的潜在空间表征,利用这种表征重构输出。

压缩模块的输入由图片经过卷积神经网络实现,卷积神经网络中最核心的操作为卷积层,卷积操作可以将原始图像信号中的某些图像特征点进行增强。卷积神经网络中的 d 层卷积可以看作不同的特征提取器,那么压缩子模块的输入可以看作 d 组特征。本文利用卷积层作为压缩子模块,学习输入特征的潜在表征,减少来自特征提取模块的特征通道数,提高目标区域的权重,使特征更加适用于分类任务。压缩子模块采用固定压缩比,压缩之后的特征通道数从 256 减少到 64。

压缩子模块具体作用如图 5 所示。图 5(a)为未经过压缩子模块特征的可视化结果,可以看出,未经过压缩子模块的特征中背景区域占据较大权重,算法忽略了目标区域;图 5(b)为经过压缩子模块的特征可视化结果,可以看出,虽然压缩子模块减少了特征的通道数,但是对输入特征进行重构之后,提高了对目标区域的关注,有利于后续任务的实现。



(a) 未经压缩子模块　　(b) 经压缩子模块

图 5　压缩子模块作用示意图

1.3.2　注意力子模块

参数自适应模块需要在在线跟踪过程中更新参数学习目标的特征。由文献[24,25]可知,在在线参数更新中,若直接使用标准卷积,由于特征所有像素中,属于背景特征的像素比较多,这将导致对背景区域的学习占据主要位置。由图 5 知,虽然压缩子模块使目标从背景中分离出来,但是相似物产生的干扰也愈发明显。由文献[26]知,这是因为将用于目标检测任务的离线训练网络应用于跟踪任务时,只有少数卷积核在表征目标时处于激活状态,由此产生的特征中只有很少的通道可以感知到当前跟踪目标,因此需要注意力机制对特征作进一步的调整。

综上考虑,本文采用双重注意力机制,包括空间注意力和通道注意力机制,对特征进行调整,解决训练和前向计算过程中特征的通道和空间维度数据失衡问题,以此来提高网络对特定目标的关注。

如图 6(a)所示,通道注意力机制首先对输入特征 $\boldsymbol{F}$ 进行全局平均池化,形成每个通道的特征,扩大网络的感受野;再通过 2 个全连接层,利用训练得到的网络参数对通道的权重进行调整。在调整权重时第 1 个全连接层将权重映射到较低的维度,其目的为间接压缩通道,得到较重要的通道信息;第 2 个全连接层将通道权重映射到较高维度,其目的是得到原特征信息每个通道的权重;最后通过 Sigmoid 激活函数将通道权重映射到 0~1 之间,得到最终每个通道的权重,形成通道注意力特征,提高网络对特定种类目标的关注度。为提高计算速度,全连接层由矩阵乘法实现。通道注意力计算过程如式(11)所示。

$$\varnothing_c = \delta(\boldsymbol{W}_1(\boldsymbol{W}_0(\mathrm{GAP}(\boldsymbol{F})))) \times \boldsymbol{F} \tag{11}$$

式中 GAP 为全局平局池化,$\boldsymbol{W}_0$、$\boldsymbol{W}_1$ 为 2 个全连接层的权重,$\boldsymbol{F}$ 为注意力机制输入特征,δ 为 Sigmoid 激活函数。

图像特征由卷积操作得到,而特征的不同通道是由参数不同的卷积核提取得到。由于卷积计算是滑窗操作,不同通道的二维空间信息将被保留下来,其中包含属于跟踪目标特征的空间位置信息。空间注意力机制就是利用卷积这种特性,为特征的每个二维空间位置重新分配权重,达到为不同特征分配权重的目的。如图 6(b)所示,空间注意力机制对输入特征每个二维空间位置进行平均池化,再经过 Relu 激活操作形成二维空间注意力特征,得到特征位置权重。空间注意力机制计算过程如式(12)所示。

$$\varnothing_s = \sigma(\mathrm{AvgPool}(\boldsymbol{F})) \times \boldsymbol{F} \tag{12}$$

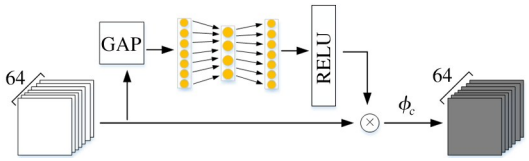
式中 AvgPool 表示平均池化操作,σ 表示 Relu 激活

函数。

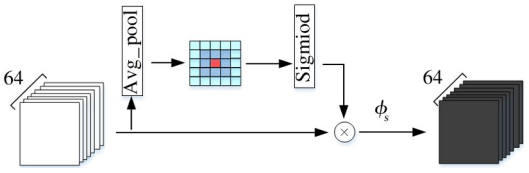
如图 6 中(c)所示,本文利用通道和空间注意力机制的加和实现双重注意力机制,达到同时利用特征空间和通道信息的目的。双重注意力机制操作过程如式(13)所示。

$$\varnothing_{s\&c} = \varnothing_c + \varnothing_s \tag{13}$$

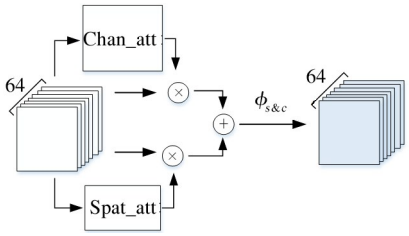
式中 + 为矩阵加法。



(a) 通道注意力模块



(b) 空间注意力模块



(c) 双重注意力机制

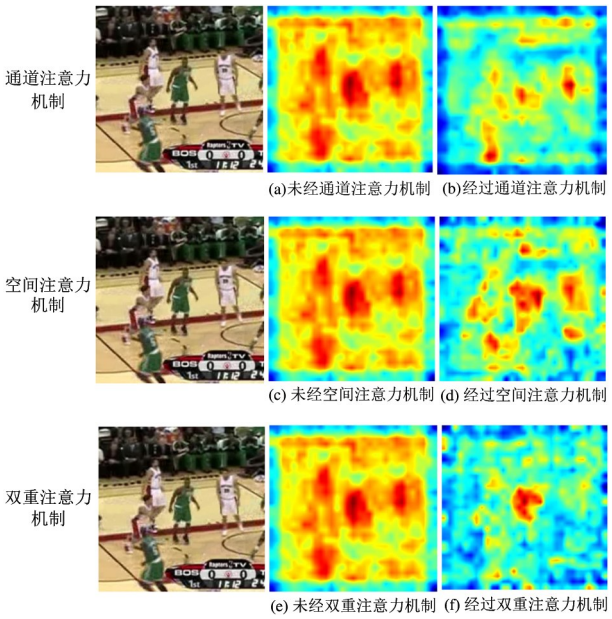
⊗代表哈达玛积　⊕代表加权求和

图 6　通道和空间注意力示意图

双重注意力机制、通道注意力机制、空间注意力机制作用如图 7 所示。从图 7(a)、(b)可以看出,经过通道注意力机制之后,网络将关注的区域集中到所有的篮球运动员,可见通道注意力可以增强网络对特定类别的关注;从图 7(c)、(d)可以看出,经过空间注意力之后,网络将关注的区域集中到要跟踪的篮球运动员,可见空间注意力机制可以提高网络对跟踪目标区域的关注;从图 7(e)、(f)可以看出,经过双重注意力之后,目标从背景中分离出来,可见双重注意力机制兼具空间和通道注意力两者的优点。

1.3.3　滤波器子模块

文献[7]指出,即使提取到能对干扰物感知(特定于当前目标)的特征之后,基于孪生网络结构的



(a)未经通道注意力机制　(b)经过通道注意力机制

(c) 未经空间注意力机制　(d) 经过空间注意力机制

(e) 未经双重注意力机制　(f) 经过双重注意力机制

图 7　注意力模块作用示意图

跟踪算法在跟踪过程中也容易被相似物所干扰。产生这种现象的原因在于跟踪过程中网络的权重是固定的,无法对噪声产生抑制作用。针对以上问题,本文引入在线更新的滤波器子模块,抑制在线跟踪过程中的背景噪声。滤波器子模块由卷积实现,卷积核的参数在跟踪过程中不断更新。因训练样本在跟踪过程中根据网络跟踪结果动态提取样本,而在跟踪过程中无法避免存在定位误差,所以训练样本中难免存在背景噪声,因此滤波器模块必须输出样本的置信度,用于确定是否将样本放入训练样本池。

(1) 滤波器训练样本。滤波器的样本池大小为 m, 其中有 30 个样本是由第 1 帧给定的真实目标经过数据增强得到。样本在网络开始预测时,用于压缩子模块、注意力子模块和滤波器子模块参数的训练,并且在样本池中保持不变,不会随着样本池的更新被替换掉。当样本池中样本未达到 m 时,直接保存受到相似物干扰和正常情况的样本。当样本达到 m 时,采用替换策略,利用新的合适样本替换最旧的样本。对于训练样本的标签,本文采用在跟踪过程中根据目标框预测出的目标位置设置的具有高斯分布的标签值。

(2) 滤波器输出结果。特征经过滤波器之后,输出参数自适应模块最终的结果,结果包含 2 部分:

1) 检测帧每个像素点的得分 $\boldsymbol{f}$。

2) 关于得分的置信度,其中置信度可以分为4种情况:

① 未找到目标位置(最高得分低于0.25);

② 受到相似物的干扰(以最高得分点坐标为中心点,根据上一帧目标框大小,若在目标框大小范围之外有得分(次高分)超过最高得分的0.8倍);

③ 不确定(次高分超过最高分的0.5倍,并且次高分大于0.25);

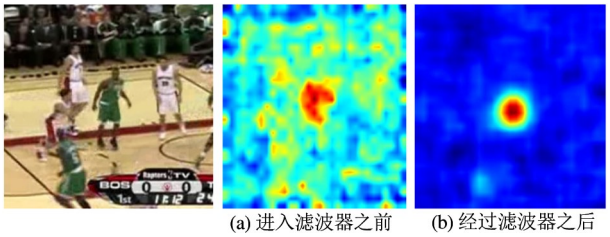
④ 正常情况(未出现以上情况)。

(3) 滤波器输出结果与分类融合。获得滤波器子模块的输出之后,使用三次插值将输出调整到与分类子分支输出相同的大小,然后通过加权求和融合在一起,得到分类子模块的输出得分,表达式可表示为

$$\boldsymbol{S}_{\mathrm{cls}} = \beta_c \boldsymbol{f}(x_j,\ w_j) + (1-\beta_c)\boldsymbol{A}_{wh2}^{\mathrm{cls}} \tag{14}$$

式中,β_c 为超参数。

滤波器子模块的作用如图8所示,图中(a)表示进入滤波器之前的特征的热力图,其中存在相似物的干扰,这会影响到网络的跟踪效果;(b)表示经过滤波器之后的特征的热力图,滤波器在滤除噪声的同时目标区域的响应达到最大。



(a) 进入滤波器之前　(b) 经过滤波器之后

图8　滤波器子模块作用示意图

1.4　模板更新

基于孪生网络的目标跟踪算法通过将模板帧图片和检测帧图片中的目标特征映射成高维特征,之后通过模板帧特征匹配检测帧特征完成跟踪目标的搜索,因此模板帧图像和检测帧图像相似是完成目标跟踪的重要前提。而在实际目标跟踪过程中,跟踪目标会发生剧烈的变化,跟第1帧时的外观完全不同,因此,需要在线上测试时更换跟踪模板,以适应目标剧烈的外观变化。

尽管STMTrack[27]中给出了比较简单的模板更新策略,但在一些比较有挑战性的跟踪场景中存在几个缺点:(1)当上一帧目标跟丢时,在跟踪当前帧目标时仍会使用上一帧图像的特征,此时跟踪模板的质量会大大降低,进而影响跟踪效果;(2)除了第1帧和当前帧的前一帧,其他历史帧的选择方式是固定的,无法判断其图像中目标的跟踪质量,导致模板的质量无法确定,进而影响后续对跟踪目标的预测。

在线更新跟踪模板可以实时获得特定目标的信息,但是文献[20]指出,孪生结构的跟踪算法仅仅使用第1帧或者更新原始模板会使模型的性能下降,特别是在大的形变、运动模糊等具有挑战性的跟踪场景下,因为跟踪过程中噪声的加入会使模板丢失目标的特征。考虑到以上问题及其他算法中模板更新策略的缺陷,本文引入一个与孪生网络并行且相互独立的子网络。如图1所示,该网络同主网络一样包含相同的结构(特征提取网络、分类与回归模块),但跟踪模板会被更新,并且同主网络一样输出跟踪目标的预测框,子网络预测框与主网络输出进行比较,选择合适的预测框作为最终的目标框。子网络和主网络输出预测框使用交并比进行相似度的衡量,交并比(intersection over union, IoU)公式如下所示:

$$IoU = \frac{\boldsymbol{A} \cap \boldsymbol{B}}{\boldsymbol{A} \cup \boldsymbol{B}} \tag{15}$$

式中,$\boldsymbol{A}$ 表示主网络输出预测框,$\boldsymbol{B}$ 表示子网络的输出预测框。

预测框筛选机制原则如下,若子网络输出预测框与主网络输出预测框的交并比超过0.6并且子网络预测框得分 S_B 比主网络输出得分 S_A 高0.5,此时选择子网络输出作为最终结果,否则选择主网络输出结果。

子网络跟踪模板选择方式如下。

自视频第1帧开始,子网络每5帧更新一次模板,对于模板帧的选择,网络采用替代原则保存最好的模板图像,替代原则如下所述。(1)本次目标框分类得分大于已保存的图像;(2)本次目标框的预测结果未受到相似物的干扰。子网络模板更新示意图如图9所示。

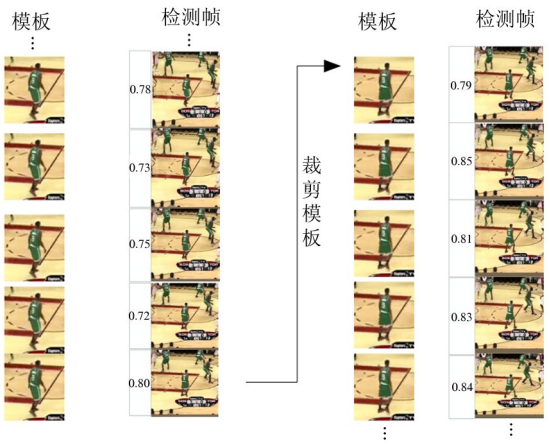


图 9　模板更新示意图

2　算法步骤

本文提出基于参数自适应与模板更新的孪生网络跟踪算法,具体步骤如下。

(1)输入目标跟踪图像序列。输入要跟踪的目标图像序列,根据给定的目标真实边界框获取第 1 帧中目标的中心位置和尺寸大小。

(2)根据给定的目标位置与尺寸大小裁剪图片,将裁剪的图片送入特征提取模块,得到网络第 3、4、5 层的输出作为跟踪时需要的模板。

(3)初始化子网络。将第(2)步中第 3、4、5 层的输出作为子网络的跟踪模板。

(4)初始化参数自适应模块。利用式(9)定义的损失函数进行训练,得到模块中的网络参数。

(5)计算主网络和子网络的跟踪结果。以上一帧预测得到的目标中心点位置为中心点裁剪得到 255×255 的检测帧图像,送入特征提取模块,得到特征提取模块中第 3、4、5 层输出的特征。将特征与第(2)步得到的跟踪模板送入分类与回归模块,根据式(2)计算互相关特征,根据式(3)~(8)和式(14),得到主网络输出预测框。将特征送入子网络根据式(2)~(8),得到另一个预测框。

(6)根据式(17)计算第(5)步得到的 2 个预测框的相似度,经过筛选机制进行筛选,得到最终的目标框。

(7)为参数自适应模块增加训练样本。根据滤波器子模块中的样本筛选条件,选择满足条件的训练样本。

(8)更新子网络的跟踪模板和滤波器模块参数。在跟踪开始后,每 5 帧更新一次子网络的模板,并且根据式(9)训练滤波器参数。

(9)若满足重复条件,跳转到第(5)步,否则跳转到第(10)步。

(10)输出跟踪结果。

3　实　验

为验证本文算法的有效性,实验采用 VOT 2018[28] 和 VOT 2019[29] 作为评价数据集。VOT 2018 包含 60 个具有更精细人工标注的目标跟踪图像序列,即每个真实框由 4 个坐标点组成,含有遮挡、光照变化、运动变化、尺寸变化和相机移动等跟踪难点。VOT 2019 是通过替换 VOT 2018 中跟踪难度比较小的 20% 目标跟踪图像生成得到,其跟踪难度更高。

3.1　实验平台

本文算法所作实验均在台式机上进行,处理器为 Intel core(TM)i7-8700K,主频为 3.70 GHz,内存为 32 GB,显卡为 Nvidia GTX1080Ti;操作系统为 64 位 Ubuntu 16.04,编程环境为 Python 3.7。

3.2　实验参数设置

3.2.1　超参数设置

在实际应用过程中,对于不同的数据集,需要采用不同的参数设置才能获得更好的性能表现。因此,针对不同的数据集 VOT 2018、VOT 2019,需要不同的超参数。本文采用 SiamBAN 算法中针对 VOT 2018、VOT 2019 数据集的超参数,即对于 VOT 2018,$k=0.08$, $k_{wi}=0.46$, $\alpha_{lr}=0.44$;对于 VOT 2019,$k=0.001$, $k_{wi}=0.33$, $\alpha_{lr}=0.46$。

3.2.2　其他参数设置

关于参数自适应模块中的参数设置,训练样本权重 γ_j 为 0.01,当邻近目标受到干扰时设置为 0.02;为有效地融合分类得分,对于式(14),β_c 采用以下步骤寻优:(1)设置搜索区间为[0.1,1.0],步长为 0.1;(2)根据第(1)步找到的性能第 1 与第 2 的参数,重新设置搜索区间[0.7,0.9],步长为 0.01,最终找到的最优超参数为 0.7。

3.3 对比实验

在 VOT 2019 标准数据集上设置对比实验,用于评估引入的参数自适应模块、模板更新子网络的作用。采用期望重叠率(expected average overlap, EAO)、准确性(accuracy, A)、鲁棒性(robustness, R)、跟丢次数(lost number, LN)、每秒帧数(frames per second, FPS)这 5 个评价指标对改进的算法进行评估。

3.3.1 使用参数自适应模块

参数自适应模块(parameter adaptive, PA)包含压缩子模块、注意力子模块、滤波子模块,效果如表 1所示。在 VOT 2019 数据集上,期望重叠率(expected average overlap, EAO)从 0.302 提高到 0.312,提高了 1%;大幅度减少了跟丢次数,从 SiamCAR 的 93 次减少到 64 次,获得了稳健的跟踪性能。

表 1 加入参数自适应模块

Tracker	A	R	LN	EAO	FPS
SiamCAR	0.594	0.467	93.0	0.302	34
+ PU	0.572	0.321	64.0	0.312	20

3.3.2 使用模板更新

在加入参数自适应模块的基础上,进一步加入模板更新(template update, TU),在 VOT 2019 数据集上,做对比实验。因模板更新间隔帧数是一个重要的超参数,首先对模板更新间隔帧数 F_g 做具体的实验,找到最佳值。如表 2 所示,当 F_g 为 5 时,算法的性能最好。这是因为算法在跟踪过程中,对目标框的预测总会有偏差,当过于频繁地更新跟踪模板时,可挑选的模板相对较少,模板质量无法保证;而当模板更新不频繁时,算法会无法适应目标的变化。如表 3 所示,加入模板更新,EAO 提升到 0.331,跟丢次数减少到 58 次,进一步提升了算法的鲁棒性,获得了较好的跟踪性能。

表 2 模板更新间隔帧数

F_g	2	3	5	7	10
EAO	0.304	0.318	0.331	0.313	0.307

表 3 加入模板更新

Tracker	A	R	LN	EAO	FPS
SiamCAR	0.594	0.467	93.0	0.302	34
+ PA + TU	0.586	0.291	58.0	0.331	20

3.4 VOT 数据集实验结果与分析

3.4.1 VOT 2019 实验

尽管孪生网络跟踪算法体现了深度神经网络强大的表征能力,但当前基于孪生网络的跟踪算法仍然会在面临相似物干扰、遮挡和较大形变时性能下降。本文算法引入参数自适应模块和模板更新子网络,为验证本文算法的有效性,引入在 VOT2019 上表现比较好的 SiamMargin[29]、SiamFCOT[29]、DiMP[30]、SiamBAN、DCFST[29]、SiamDW _ ST[29]、ARTCS[29]、SiamMask、SiamRPN + +、SPM[31]、SiamCRF-RT[29]、ATOM[32]、SiamCAR[33] 等 13 种跟踪算法,采用期望重叠率、准确率、鲁棒性、跟丢次数、每秒帧数这 5 个指标对 13 种性能优异的跟踪算法进行比较,结果如表 4 所示。

表 4 不同算法在 VOT2019 上的结果对比

Tracker	A	R	EAO	LN	FPS
SiamMargin	0.579	0.321	0.366	65	46
SiamFCOT	0.601	0.386	0.350	77	-
SiamBAN	0.602	0.396	0.327	79	40
DiMP	0.582	0.371	0.321	74	40
DCFST	0.585	0.376	0.317	75	-
SiamCAR	0.594	0.467	0.302	93	34
SiamDW-ST	0.600	0.467	0.299	93	-
ARTCS	0.602	0.482	0.287	96	-
SiamMask	0.594	0.461	0.287	92	35
SiamRPN ++	0.599	0.482	0.285	96	35
SPM	0.577	0.507	0.275	101	120
SiamCRF-RT	0.549	0.346	0.262	69	-
ATOM	0.579	0.557	0.240	111	30
本文算法	0.586	0.291	0.331	58	20

本文提出的算法在 VOT 2019 测试集上跟踪速率和最好算法存在差距,这是因为算法在测试过程中需要在线训练,不断优化参数自适应模块中的卷积参数,这会拖慢算法的执行效率,但本文算法也达

到实时需要的 20 FPS,虽然算法的 EAO 表现不是最高,但是丢帧次数是最少的,说明本文采用的参数自适应模块和模板更新子网络是有效的,可以提高网络的稳健性。

3.4.2　VOT 2018 实验

为验证算法在其他数据集上的有效性,在 VOT 2018 上进行测试。VOT 2018 数据集相对于 VOT 2019 跟踪挑战难度略有下降,但是仍然可以被认为是较全面的测试集。

图 10 是从 VOT 2018 数据集中选取 5 个具有各种跟踪难点的视频图像序列,可视化本文算法与 SiamCAR(下文简称 CAR)实际跟踪效果(图中未标注目标框的算法代表该算法在当前帧中已丢失目标)。

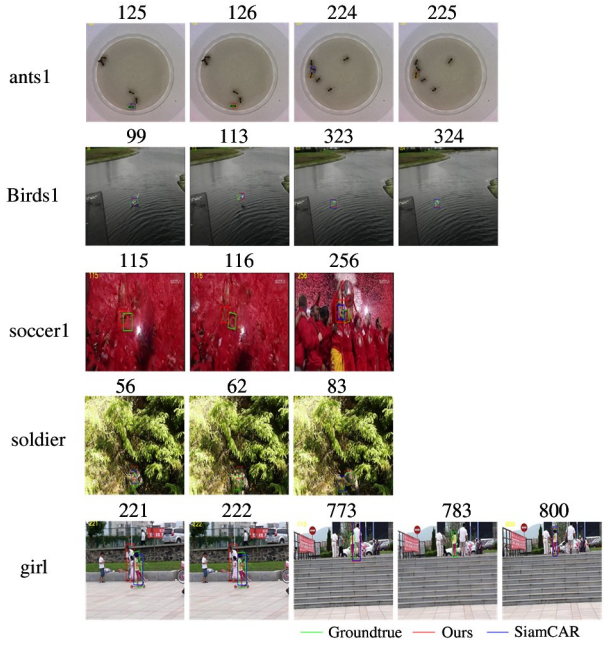


图 10　VOT 2018 部分序列结果对比图

对于 ants1 测试视频的跟踪,其主要跟踪难点在于被跟踪目标尺寸较小和相似目标的干扰。由可视化图可知,CAR 算法在第 125、126 和第 224、225 帧受到与跟踪目标相同类别物的干扰,并且在第 126、225 帧丢失跟踪目标,说明 CAR 算法抵抗相似物干扰的性能不强,而本算法可以在有相似物干扰的情况下,稳健地跟踪目标,体现了本算法鲁棒性的提高。

对于 birds1 测试视频的跟踪,其主要难点是大的形变和相似目标的干扰。由图可知,在第 99 和 113 帧中,跟踪目标与其相似的干扰距离过近或者在图像中显示部分区域重叠,此时 CAR 算法受到干扰,并且在下一帧丢失目标,而本文算法依然能够比较准确地跟踪目标。在第 323 帧中本文算法受到目标大的形变的干扰,但在第 324 帧可以及时地调整目标框,跟踪上目标。这说明参数自适应模块与模板更新的引入可以学习跟踪目标的特征,有效地调整跟踪器在跟踪过程中的状态。

对于 soccer1 测试视频的跟踪,其主要难点是遮挡和模糊。由图可知,在第 115 帧中,由于遮挡与图像模糊同时发生,CAR 算法已经丢失跟踪目标,本文算法也受到这种跟踪挑战的影响,几乎丢失目标,但是在第 116 帧中,当遮挡情况转好,并且模糊有所改善时,本文算法可以及时地调整目标框,跟踪上目标。在第 256 帧中,此时跟踪目标出现模糊,并且此时出现相似物的干扰,CAR 算法几乎丢失跟踪目标,而本文算法仍然可以稳健地跟踪目标。由此可以看出本文算法在面临遮挡与模糊的情况下依然可以实现跟踪。

对于 soldier 测试视频,跟踪对象为士兵的头盔,其主要跟踪难点在于被跟踪目标的背景比较杂乱且存在目标遮挡。从第 56 帧开始,CAR 算法和本文算法均受到背景的干扰,并逐渐偏离真实目标框,到第 62 帧 CAR 算法跟丢目标,再到第 83 帧,CAR 算法又一次跟丢目标,而本文算法从偏离目标到逐渐跟踪上目标。可见本文提出的算法不仅可以区分相似目标的干扰,还可以随着跟踪的进行,逐渐学习目标的特征,不断完善跟踪的效果。

对于 girl 测试视频的跟踪,跟踪目标为小女孩,其跟踪难点在于背景(非跟踪目标区域)中的人对跟踪目标的遮挡以及由此带来的相似物干扰。由可视化图可知,从第 221 帧本文算法受到相似物干扰,并在第 222 帧偏离跟踪目标,但是在第 224 帧又重新跟踪上目标。同样在第 773 帧 CAR 算法与本文算法同样受到相似干扰,而 CAR 算法在第 783 帧丢失跟踪目标,而本文算法仍能跟踪目标,并且在第 800 帧将目标框收缩到跟踪目标。可见本文算法可以在偏离目标的时候自适应地跟踪上目标,达到减

少丢帧的效果,进而提高了算法的稳健性。

4 结 论

传统孪生网络跟踪算法网络参数由线下训练得到,并且在跟踪时参数固定,这使得算法缺少特定于跟踪目标的信息;而且在跟踪时跟踪模板仅仅使用第1帧给定的目标,对实时变化的目标的适应性较差,这导致算法的鲁棒性较差。为此,本文提出基于参数自适应与模板更新的孪生网络跟踪算法,通过引入参数自适应模块、模板更新子网络提高了算法在光照变化、尺度变化,遮挡等具有挑战性的情况下的稳健性。参数自适应模块通过空间注意力机制提高了网络对跟踪目标空间位置的关注度,通道注意力机制提高了网络对特定目标类别的关注。通过在线训练,更新滤波器参数策略滤除了干扰,提高了算法的判别能力;模板更新子网络通过不断更新跟踪模板,得到候选预测框,利用筛选机制挑选出合适的目标框,提高了网络对跟踪目标变化的适应性。在VOT 2018、VOT 2019 2个数据集上对算法进行测试,EAO分别达到0.331和0.455,再分别与其他算法进行比较,本文算法的性能同样具有竞争力,验证了本文算法的有效性。

参考文献

[1] AMADO J, GOMES I P, AMARO J, et al. End-to-end deep learning applied in autonomous navigation using multi-cameras system with RGB and depth images[C]// IEEE Intelligent Vehicles Symposium. Paris: IEEE, 2019:1626-1631.

[2] XIONG Z, CAI Z, HAN Q, et al. ADGAN:protect your location privacy in camera data of auto-driving vehicles [J]. IEEE Transactions on Industrial Informatics,2020, 17(9):6200-6210.

[3] VOINEA G D, GIRBACIA F. Vision-based system for driver posture tracking to prevent musculoskeletal disorders[C]//2020 International Conference on E-health and Bioengineering. LASI: IEEE, 2020:1-4.

[4] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully convolutional SIAMESE networks for object tracking [C]// European Conference on Computer Vision. Amsterdam: ECCV, 2016:850-865.

[5] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016,39(6):1137-1149.

[6] LI B, YAN J, WU W, et al. High performance visual tracking with Siamese region proposal network[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Wellington: IEEE,2018:8971-8980.

[7] ZHU Z, WANG Q, LI B, et al. Distractor-aware Siamese networks for visual object tracking[C]// Proceedings of the European Conference on Computer Vision. Munich: ECCV, 2018:101-117.

[8] LI B,WU W,WANG Q,et al. SiamRPN++: evolution of Siamese visual tracking with very deep networks[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2019: 4282-4291.

[9] ZHANG Z, PENG H. Deeper and wider Siamese networks for real-time visual tracking[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019:4591-4600.

[10] WANG Q, ZHANG L, BERTINETTO L, et al. Fast online object tracking and segmentation: a unifying approach[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE,2019:1328-1338.

[11] HAN G, DU H, LIU J, et al. Fully conventional anchor-free siamese networks for object tracking[J]. IEEE Access, 2019,7:123934-123943.

[12] TIAN Z, SHEN C, CHEN H, et al. FCOS:fully convolutional one-stage object detection[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019:9627-9636

[13] XU Y, WANG Z, LI Z, et al. Siamfc++: towards robust and accurate visual tracking with target estimation guidelines[C]// Proceedings of the AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020: 12549-12556.

[14] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition. Boston: IEEE, 2015:1-9.

[15] CHEN Z, ZHONG B, LI G, et al. Siamese box adaptive network for visual tracking[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. San Francisco: IEEE, 2020:6668-6677.

[16] LI Q, QIN Z, ZHANG W, et al. Siamese key point prediction network for visual object tracking [EB/OL]. (2020-06-07)[2022-12-11]. http://arxiv.org/pdf/2006.04078.pdf.

[17] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017:6000-6010.

[18] WANG N, ZHOU W, WANG J, et al. Transformer meets tracker: exploiting temporal context for robust visual tracking[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Kuala Lumpur: IEEE, 2021:1571-1580.

[19] YAN B, PENG H, FU J, et al. Learning spatiotemporal transformer for visual tracking[C]// Proceedings of the

IEEE/CVF International Conference on Computer Vision. Kuala Lumpur: IEEE,2021:10448-10457.
[20] 卢湖川, 李佩霞, 王栋. 目标跟踪算法综述[J]. 模式识别与人工智能, 2018,31(1):61-76.
[21] LI X, MA C, WU B Y, et al. Target-aware deep tracking [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019:1369-1378.
[22] ZHOU J, WANG P, SUN H. Discriminative and robust online learning for Siamese visual tracking[C]//Proceedings of the AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2020:13017-13024.
[23] 任杰. 基于深度学习的图像压缩方法研究[D]. 哈尔滨:哈尔滨工业大学, 2017.
[24] LI X, MA C, WU B, et al. Target-aware deep tracking [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019:1369-1378.
[25] YANG K, SONG H, ZHANG K, et al. Hierarchical attentive Siamese network for real-time visual tracking[J]. Neural Computing and Applications, 2020, 32(18): 14335-14346.
[26] 陈志旺,张忠新,宋娟,等. 基于目标感知特征筛选的孪生网络跟踪算法[J]. 光学学报,2020,40(9): 0915003.
[27] FU Z, LIU Q, FU Z, et al. Stmtrack: template-free visual tracking with space-time memory networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Kuala Lumpur,: IEEE, 2021:13774-13783.
[28] KRISTAN M,LEONARDIS A,MATAS J,et al. The sixth visual object tracking VOT 2018 challenge results[C]//Proceeding of the European Conference on Computer Vision Workshop. Munich: ECCV, 2018:3-53.
[29] KRISTAN M, MATAS J, LEONARDIS A, et al. The seventh visual object tracking VOT 2019 challenge results [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. Seoul: IEEE, 2019:1-36.
[30] BHAT G, DANELLJAN M, GOOL L V, et al. Learning discriminative model prediction for tracking[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019:6182-6191.
[31] WANG G, LUO C, XIONG Z, et al. SPM-tracker: series-parallel matching for real-time visual object tracking [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019:3643-3652.
[32] DANELLJAN M, BHAT G,KHAN F S,et al. ATOM: accurate tracking by overlap maximization[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019: 4660-4669.
[33] GUO D, WANG J, CUI Y, et al. SiamCAR: Siamese fully convolutional classification and regression for visual tracking[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2020:6269-6277

Siamese network tracking algorithm based on parameter adaptive and template updating

CHEN Zhiwang[*,**], GUO Jinhua[*], LV Changhao[***], LEI Chunming[*], PENG Yong[****]
([*]Engineering Research Center of the Ministry of Education for Intelligent Control System and Intelligent Equipment, Yanshan University, Qinhuangdao 066004)
([**]Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004)
([***]Key Laboratory of Power Electronics for Energy Conservation and Drive Control of Hebei Province, Yanshan University, Qinhuangdao 066004)
([****]School of Electrical Engineering, Yanshan University, Qinhuangdao 066004)

Abstract

The network parameters of the Siamese network tracking algorithm are fixed during the tracking process, and the tracking template is only from the first frame, which make the robustness of algorithm poor. Therefore, a Siamese network tracking algorithm based on parameter adaptive (PA) and template updating is proposed. Firstly, the target feature is adjusted by channel attention and spatial attention to improve the attention of network to tracking target; secondly, the filter parameter update strategy is used to filter out the interference of background, which leads to identify the current target; finally, a subnetwork parallel to the main network is added, and the network can adapt to the change of the target by updating the tracking template. The expected average overlap (EAO) reaches 0.455 and 0.331 respectively on the VOT2018 and VOT2019 benchmarks, which verifies the effectiveness of the algorithm.

Key words: object tracking, Siamese network, template updating, parameter adaptive(PA), attention mechanism