基于目标感知特征筛选的孪生网络跟踪算法

陈志旺[1,2], 张忠新[1*], 宋娟[3], 罗红福[1], 彭勇[4]

[1]燕山大学工业计算机控制工程河北省重点实验室, 河北 秦皇岛 066004;

[2]燕山大学国家冷轧板带装备及工艺工程技术研究中心, 河北 秦皇岛 066004;

[3]国网黑龙江省电力有限公司佳木斯供电公司, 黑龙江 佳木斯 154002;

[4]燕山大学电气工程学院, 河北 秦皇岛 066004

摘要 孪生网络跟踪算法是利用离线训练好的网络提取目标特征并进行匹配,从而实现跟踪。而离线训练深度特征在表征任意形式目标时将目标从背景中分离开的性能较差。为此,提出一种基于目标感知特征筛选的孪生网络跟踪算法。将经过裁剪处理后的模板帧和检测帧送入到 ResNet50 的特征提取网络分别提取目标和搜索区域的浅层、中层、深层特征;在目标感知模块中,通过设计一个回归损失函数来学习对目标敏感的特征,根据反向传播的梯度来确定每个卷积核的重要性程度,并以此来激活相对重要的卷积核筛选较重要的目标感知特征;将筛选得到的特征送入到 SiamRPN 模块,进行目标、背景的二分类判别和边界框的坐标回归,从而得到一个精确的目标边界框。在 OTB2015 和 VOT2018 两个标准数据集上进行测试实验,结果表明该算法可以实现对目标的稳健性跟踪。

关键词 机器视觉; 目标跟踪; 孪生网络; 目标感知

中图分类号 TP391.4 **文献标志码** A **doi**: 10.3788/AOS202040.0915003

Tracking Algorithm for Siamese Network Based on Target-Aware Feature Selection

Chen Zhiwang[1,2], Zhang Zhongxin[1*], Song Juan[3], Luo Hongfu[1], Peng Yong[4]

[1]*Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao, Hebei* 066004, *China*;

[2]*National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Yanshan University, Qinhuangdao, Hebei* 066004, *China*;

[3]*Jiamusi Electric Power Company, State Grid Heilongjiang Electric Power Co., Ltd., Jiamusi, Heilongjiang* 154002, *China*;

[4]*School of Electrical Engineering, Yanshan University, Qinhuangdao, Hebei* 066004, *China*

Abstract Tracking algorithms implemented in Siamese networks utilize an offline training network to extract features from a target object for matching and tracking. The offline-trained deep features are less efficient for distinguishing targets with arbitrary forms from the background. Therefore, we proposed a tracking algorithm for a Siamese network based on target-aware feature selection. First, the cropped template and detection frames were sent to a feature extraction network based on ResNet50 to extract the shallow, middle and deep features of the target and search regions. Second, in the target-aware module, a regression loss function was formulated for target-aware features and an importance scale for each convolution kernel was obtained based on backpropagated gradients. Then, the convolution kernels with large importance scales were activated to select target-aware features. Finally, the selected features were inputted into the SiamRPN module for target-background classification and the bounding box regression was applied to obtain an accurate bounding box of the target. Results of experiments on OTB2015 and VOT2018 datasets confirm that the proposed algorithm can achieve robust tracking of the target.

Key words machine vision; object tracking; Siamese network; target-aware

OCIS codes 150.1135; 100.4996; 100.4999

收稿日期: 2019-12-19; **修回日期**: 2020-01-14; **录用日期**: 2020-01-19

基金项目: 国家自然科学基金(61573305)

* **E-mail**: ZZXin00016@163.com

1 引　言

目标跟踪是计算机视觉领域中具有广泛应用的基本问题之一，例如应用于自动驾驶[1]、智能视频监控[2]、无人机自主跟随[3]等。通用的目标跟踪是估计任意目标在视频序列中的状态信息(位置信息、尺度信息)，具体方法为：在第1帧图像中使用边界框框选中一个特定目标，利用给定的目标跟踪算法在后续帧中确定目标的位置，并使用边界框对该目标进行标识[4-5]。尽管目标跟踪最近已经取得了很大进展，但由于光照变化、尺度变化、形变、背景杂乱和遮挡等诸多因素，目前仍然被认为是一项具有挑战性的任务[6]，除此之外，实时性要求也成为跟踪算法投入实际应用的一大瓶颈。

现代的目标跟踪算法主要分为两个分支。一种是基于相关滤波器的算法，它利用循环卷积[7]的特性训练一个回归器并且在傅里叶域执行运算，与此同时，在跟踪过程中在线更新权重，该算法速度快但精度差。另一种是基于深度学习的跟踪算法，这种算法通常用于提取强大的深度特征[8-9]，虽然跟踪精度高但速度慢。最近，孪生网络跟踪算法因其在跟踪精度和跟踪速度上能够达到很好平衡而备受关注。

孪生网络跟踪算法将视觉对象跟踪转换成相似性得分图问题，使用卷积神经网络提取目标模板和搜索区域的特征，并对两者执行相关操作来描述目标模板和搜索区域的特征相似性，其中特征表示可以通过端到端的卷积神经网络学习得到。为了确保在线跟踪的效率，离线训练好的孪生网络相似度量函数通常在运行时是固定的。基于全卷积孪生网络的目标跟踪(SiamFC)[10]利用完全卷积的网络结构进行相似性预测，获得超过100 frame/s的高跟踪速度。基于残差注意力机制的孪生网络跟踪(RASNet)[11]算法通过学习一种残差注意力机制，使跟踪模型适应用于当前目标。基于区域提议网络的目标跟踪(SiamRPN)[12]算法在孪生网络之后引入了区域提议网络(RPN)[13]，允许使用可变宽高比的边界框估计目标位置和目标尺寸大小，并将目标、背景分类和边界框坐标回归联合起来一起进行目标跟踪，从而获取一个更加准确的边界框。基于干扰物感知的孪生网络跟踪(DaSiamRPN)[14]算法进一步引入了干扰物感知模块，并提高了模型的辨别能力。基于更深和更宽网络的孪生网络跟踪(SiamDW)[15]算法分别在SiamFC、SiamRPN的基础上，通过在更深的残差网络(ResNet)、更宽的Inception网络中引入残差块内部裁剪单元(cropping-inside residual，CIR)，进一步提高了跟踪的准确性和鲁棒性。基于深度网络的孪生网络跟踪(SiamRPN++)[16]算法在SiamRPN的基础上，使用更深的特征提取网络ResNet50[17]替代原来的AlexNet，并且加入多层融合的策略，使用逐通道互相关(depth-wise cross-correlation)操作代替了SiamFC中简单的互相关操作，从而实现更高的跟踪精度。能够进行目标分割的在线孪生网络跟踪(SiamMask)[18]算法将目标跟踪和视频语义分割统一起来，在进行目标跟踪的同时，对被跟踪目标生成一个二进制掩模，进而得到一个自适应掩模的预测边界框，大幅度提高了跟踪的准确性。

上述孪生网络跟踪算法使用的都是离线训练好的深度特征，尽管离线训练好的深度特征能够有效应用于目标检测任务，但用于目标跟踪任务时还有许多问题需要解决。

首先，在对象识别任务中，离线训练的深度模型包含有限个类别的特定对象信息，而进行在线跟踪中的目标可以是任意形式的，例如，在离线训练模型的训练集中没有出现过的对象或者是一个特定对象的某个部分的特征信息并没有包含在对象识别任务中。也就是说，来自对通用图像的离线训练模型对于感兴趣的目标是不可知的，因而将被跟踪目标从图像背景中分离出来的性能受限。其次，即使在离线训练模型的训练集中包含目标对象，但从最后的卷积层获取得到的深度特征通常获得的是关于目标对象类别相关的高级语义信息，这些信息对于跟踪问题中的精确定位或尺度估计并不有效[8]。第三，由于离线训练模型提取得到的深度特征是高维的，这些高维的特征信息更多地是在训练集中出现的目标对象通用特征信息，对于实际跟踪的单个目标对象来说，不仅存在大量的信息冗余，也使得深度学习跟踪器需要较高的计算负荷，严重影响了跟踪算法的实时性，为此，在目标跟踪阶段，从全部特征通道中筛选出与目标相关的有效特征显得尤为重要。

在SiamRPN++算法的ResNe50网络中，特征提取模块离线训练的深度特征由每个类别标签稀疏地激活，在实际的跟踪过程中，只有少数卷积核在表征目标时处于激活状态，大部分的卷积核包含冗余和无关信息。为此，本文引入目标感知模块，以获取与当前目标相关并且判别力更强的特征，从而有助于区分被跟踪目标和相似干扰。

2　本文方法

引言中经典算法之间的发展进程如图 1 所示(图中实线表示前后算法之间的继承关系,虚线表示前后算法之间的替代关系)。

根据图 1,对孪生网络用于目标跟踪各算法的优缺点进行分析。本文算法以 SiamRPN++算法为基础,引入目标感知模块,提出基于目标感知特征筛选的孪生网络跟踪算法,整体框架如图 2 所示,主要包括特征提取模块(feature extraction module)、目标感知模块(target-aware module)和 SiamRPN 模块(SiamRPN module)。

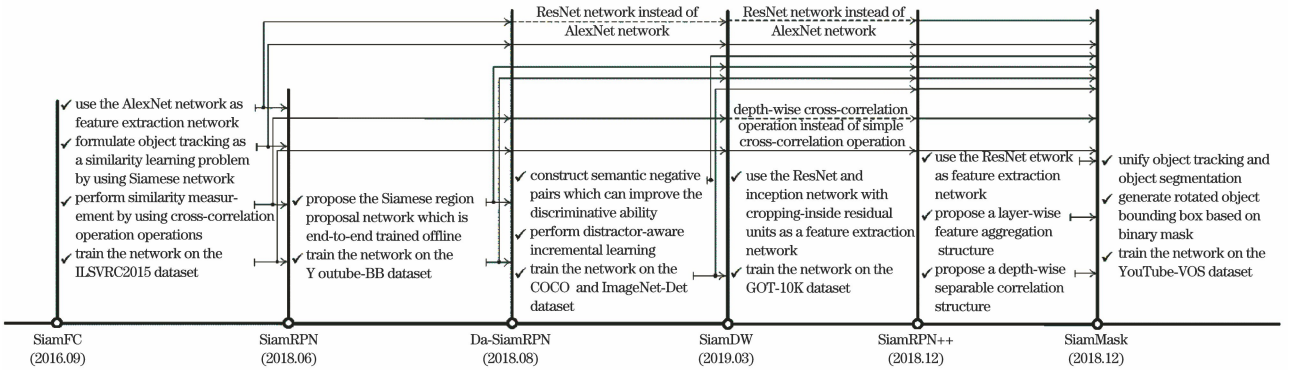

图 1　基于孪生网络的目标跟踪算法发展进程图

Fig. 1　Development process diagram of object tracking algorithm based on Siamese network

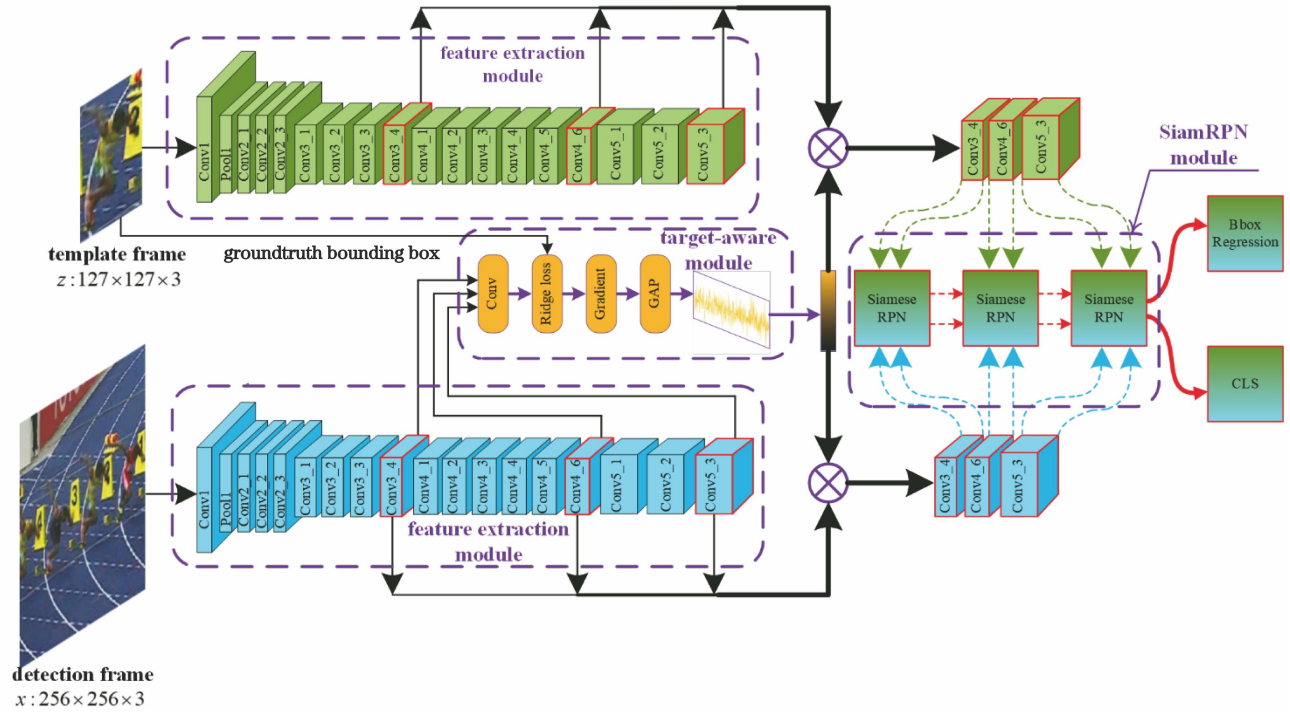

图 2　基于目标感知特征筛选的孪生网络跟踪网络结构图

Fig. 2　Framework of tracking algorithm with Siamese network based on target-aware feature selection

2.1　通用特征提取模块

2.1.1　特征提取网络

本文算法的特征提取模块如图 2 中 feature extraction module 所示,使用的特征提取网络是 ResNet50,ResNet50 各个层之间感受野变化很大:较浅的层主要提取一些低级信息,如颜色、形状等信息,这些特征信息对于定位目标位置十分重要,但是这些特征信息缺乏语义信息,难以用于目标的辨识;而较深的层提取到的特征具有丰富的语义信息,在运动模糊、目标形变严重等一些充满挑战的场景中可能是有用的。进一步通过可视化特征提取模块中各层输出的可视化结果,进行对比,如图 3 所示。可以发现:layer1、layer2 的输出特征图主要提取图片中通用对象的形状、纹理信息;从 layer3 的输出特

征图开始，提取被跟踪目标的关键信息，逐渐将具有相同类别（如图 3 中人这一类别）的对象从图片背景中分离出来，所以第 3 层具有浅层特征；layer4 的输出特征图相比 layer3 提取的特征信息更加明显，具有初步的轮廓信息，所以第 4 层具有中层特征；layer5 的输出特征图更倾向于提取目标高级的语义信息，所以第 5 层具有深层特征。由于 layer1、layer2 的输出特征图提取的是通用对象的形状、纹理信息，并没有将具有关键类别的人从背景中分离出来，所以不选用 layer1、layer2 的输出特征图。本文最终选定它的第 3 层、第 4 层、第 5 层的输出值来表征涵盖目标浅层、中层、深层的特征信息，用于在视频跟踪序列中目标的辨识和定位。SiamRPN＋＋算法中也选定第 3 层、第 4 层、第 5 层的输出值来表征涵盖目标浅层、中层、深层的特征信息。

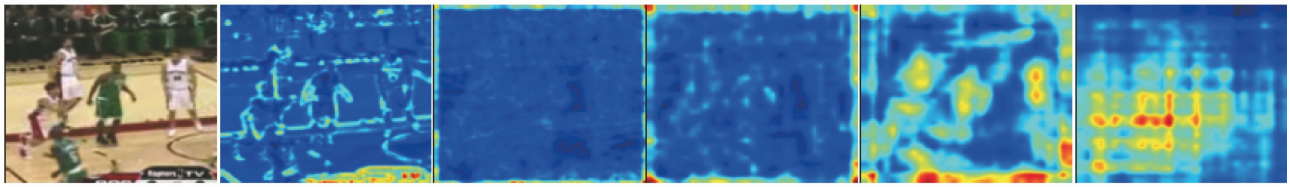

图 3　特征提取模块中各层原始特征对比实验结果

Fig. 3　Comparison experiment results of original features of each layer in the feature extraction module

如图 2 中 feature extraction module 所示，特征提取模块主要接收两种输入：一种输入为模板帧 z，通过第 1 帧图像以目标中心位置为基准结合目标尺寸大小裁剪得到，大小为 127×127；另一种输入为检测帧 x，在当前帧图像（这里假设为第 k 帧）以前一帧（第 k-1 帧）预测的目标中心位置和目标尺寸大小为基准裁剪得到，大小为 255×255。将 z 和 x 输入到特征提取模块中，通过 ResNet50 网络的第 3 层、第 4 层、第 5 层提取得到模板帧和检测帧的深度特征 $\boldsymbol{f}_l(x)(l=3,4,5)$ 和 $\boldsymbol{f}_l(z)(l=3,4,5)$。在 2.2 节中的目标感知特征筛选模块中，将模板帧和检测帧特征送入特征筛选网络，进行特征筛选。

2.1.2　模板帧特征图的获取方式

模板帧特征图的获取方式一共有两种。第一种方式[19]为通过间接方式获得（定义为模板帧 1），根据第 1 帧中目标的中心位置和尺寸大小，裁剪得到 255×255 的检测帧图像，如图 4(a)所示；将其输入到特征提取网络中提取得到第 3 层的 31×31×512、第 4 层的 31×31×1024、第 5 层的 31×31×2048 特征图，如图 4(c)、(d)、(e)所示。在得到的检测帧特征图的基础上以特征图中心位置为中心分别裁剪得到 15×15×512、15×15×1024、15×15×2048 的模板帧特征图，如图 4(f)、(h)、(j)所示。第二种方式为通过直接方式获得（定义为模板帧 2），根据第 1 帧中目标的中心位置和尺寸大小，裁剪得到 127×127 的模板帧图像，将其输入到特征提取网络中提取得到第 3 层的 15×15×512、第 4 层的 15×15×1024、第 5 层的 15×15×2048 的模板帧特征图，如图 4(g)、(i)、(k)所示。

对比可以发现，在第 3 层、第 4 层输出的特征图中，相对于整张特征图来说，图 4(f)和(g)、(h)和(i)中响应值较高的区域分别一一对应，并没有明显的差别；在第 5 层输出的特征图中，通过直接方式获取得到的模板帧特征图中响应值较高的区域整体居于整张特征图的中心区域，而在通过间接方式获取得到的模板帧特征图中，响应值较高的区域倾向于整张特征图的左下区域，这会造成一种学习偏差，因而这种获取方式效果较差；与此同时，通过实验（见 4.3.1）发现这种模板帧特征图的间接获取方式在后续 SiamRPN 模块进行相似性判别时实际的跟踪效果会出现大幅度下降，因而本文在利用第 1 帧图像的检测帧得到对当前目标敏感的卷积核后，重新按照直接方式获取模板帧的特征图。另外，重新计算模板帧特征图时额外消耗的时间是微小的，可以忽略不计。

2.2　目标感知特征筛选模块

在特征提取模块中的 ResNet50 网络中，离线训练的深度特征由每个类别标签稀疏地激活，将用于目标检测任务的离线训练卷积神经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，大部分的卷积核包含冗余和无关信息，这将导致高计算负荷和过拟合[19]。针对以上问题，本文建立目标感知模型，如图 2 中的 target-aware module 所示。

2.2.1　构建回归损失函数

在一个预先训练的分类网络中，每个卷积核用

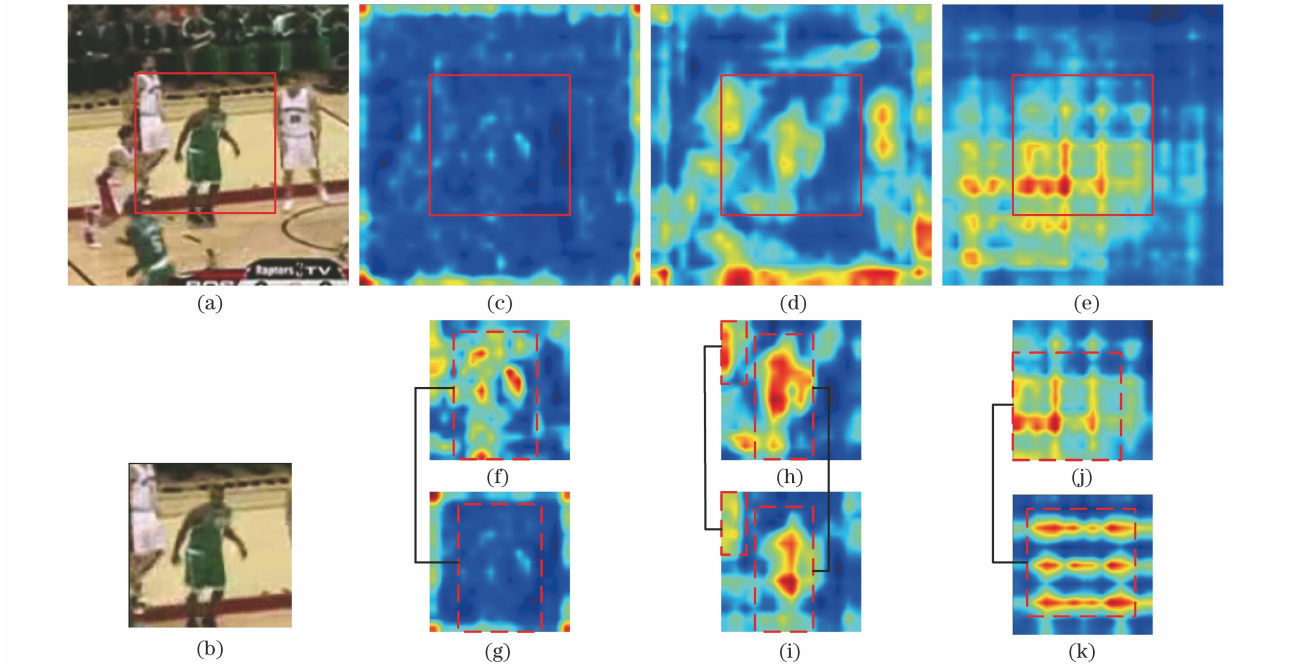

图 4　使用两种方式获取模板帧特征图对比实验结果。(a)(b)模板帧图像和检测帧图像;(c)(f)(g)第 3 层特征图;(d)(h)(i)第 4 层特征图;(e)(j)(k)第 5 层特征图

Fig. 4　Visualized results of the original features and the target-aware features. (a)(b) Template frame and detection frame; (c)(f)(g) feature maps of layer3; (d)(h)(i) feature maps of layer4; (e)(j)(k) feature maps of layer5

于获取特定的特征模式,并且所有卷积核可构建出包含不同对象的先验特征空间。一个经过离线训练的网络主要基于这些卷积核的子集来识别特定的对象类别。对于目标跟踪任务,可以通过识别对目标区域敏感的对象来获得与目标有关对象信息的卷积核,并且要求得到的卷积核对背景信息并不敏感。为此,设检测帧中提取得到的深度特征为$\boldsymbol{X}_n^l(l=3,4,5)$(等同于前文中的$\boldsymbol{f}_l(\cdot)(l=3,4,5)$,其中,$n$ 为离线训练深度特征的第 n 个特征通道,l 为 ResNet50 的第 l 层输出特征),将深度特征为$\boldsymbol{X}_n^l$ 的元素$X_n^l(i,j)(l=3,4,5)$回归到高斯标签图 $\boldsymbol{Y}$ 上,高斯标签图 $\boldsymbol{Y}$ 的元素可表示为

$$Y(i,j)=\exp\left(-\frac{i^2+j^2}{2\sigma^2}\right),\tag{1}$$

式中:(i,j)为相对于目标中心位置的偏移量;σ 为高斯核的宽度值。

通过构建岭回归损失函数,将特征提取模块的特征回归到由高斯函数(式(1))生成的软标签上,即

$$L_{\mathrm{reg}}=\|\boldsymbol{Y}-\boldsymbol{W}*\boldsymbol{X}_n^l\|^2+\lambda\|\boldsymbol{W}\|^2,\tag{2}$$

式中:$\|\cdot\|$为 L^2 范数;$*$ 为卷积操作;$\boldsymbol{W}$ 为回归器的权重;λ 为正则化系数。每个卷积核的重要性程度可以根据其对拟合高斯标签图的贡献来计算,贡献的大小可以根据回归损失的梯度来计算[20-21]。根据链式法则和(2)式,进行回归损失的梯度计算,表达式为

$$\begin{aligned}\frac{\partial L_{\mathrm{reg}}}{\partial \boldsymbol{X}_n^l}&=\left[\frac{\partial L_{\mathrm{reg}}}{\partial X_n^l(i,j)}\right]_{m\times n}=\\&\left[\sum_{p,q}\frac{\partial L_{\mathrm{reg}}}{\partial X_{\mathrm{o}}^l(p,q)}\frac{\partial X_{\mathrm{o}}^l(p,q)}{\partial X_n^l(i,j)}\right]_{m\times n}=\\&2\boldsymbol{W}^{\mathrm{T}}(\boldsymbol{X}_{\mathrm{o}}^l-\boldsymbol{Y}),\end{aligned}\tag{3}$$

式中:$\boldsymbol{X}_{\mathrm{o}}^l$为第 l 层预测的输出值;$\frac{\partial L}{\partial \boldsymbol{X}}$为 L 关于 $\boldsymbol{X}$ 的矩阵偏导;m、n 为 $\boldsymbol{X}_n^l$ 的尺寸大小。

2.2.2　特征筛选策略

由文献[20]可知,各通道对表征目标对象的贡献程度可通过(3)式生成的梯度信息利用全局平均池化函数进行计算,即

$$\Delta_n^l=G_{\mathrm{AP}}\left(\frac{\partial L_{\mathrm{reg}}}{\partial \boldsymbol{X}_n^l}\right)=\frac{1}{Z_n}\sum_{i,j}\frac{\partial L_{\mathrm{reg}}}{\partial X_n^l(i,j)},\tag{4}$$

式中:Δ_n^l 为第 l 层第 n 个通道的重要性程度;$G_{\mathrm{AP}}(\cdot)$为全局平均池化函数;$\boldsymbol{X}_n^l$ 为第 n 个卷积核的输出特征;Z_n 为第 n 个卷积核的输出特征图大小。

具体筛选策略如下:

1) 根据经验或跟踪对象确定 ResNet50 网络的第 l 层特征通道的保留个数 n_l;

2) 针对 ResNet50 网络的第 l 层,根据(4)式计算各通道的重要程度;

3) 根据各通道重要程度大小进行排序；

4) 根据排序结果保留前 n_l 个通道的特征信息。

上述筛选之后的特征称为目标感知特征 $\boldsymbol{F}$。

通过后续实验(见 4.3.2)发现：对于不同的跟踪对象，不同的 n_l 所带来跟踪性能的提升是不同的。如果 n_l 值过大，不利于筛选出具有判别力的特征，从而不能有效地将目标从图片背景中分离出来；如果 n_l 值过小，筛选后的特征信息不足以表征被跟踪目标，从而不利于目标的辨识。

本文发现与离线训练的深度特征相比，目标感知特征具有以下优点：使用一组对于特定目标敏感的卷积核来筛选得到具有判别力的目标感知特征，不仅可以减少特征的数量，而且还可减少卷积运算中使用的元素乘法以及加和操作，从而有效减少了计算负荷，同时解决了模型过度拟合的问题。目标感知特征能够有效表示训练集中的任意目标和没有出现过的对象，并使用目标感知得到的深度特征在分离与预先训练的深度特征具有相同语义标签的不同目标对象(如图 5 中，目标人与背景人有相同语义标签)方面更有效。图 5 通过平均所有通道，可视化地比较了目标感知特征和原始特征的区别，第 3 层从原来的 512 个特征通道经过特征筛选得到 256 个对当前目标敏感的特征通道，如图 5(b)所示，第 4 层从原来的 1024 个特征通道经过特征筛选得到 800 个对当前目标敏感的特征通道，如图 5(c)所示，第 5 层从原来的 2048 个特征通道经过特征筛选得到 1500 个对当前目标敏感的特征通道，如图 5(d)所示。对比模板帧第 3 层和第 4 层的输出特征图，可以发现目标感知特征有助于将目标从背景区域中分离出来；对比检测帧和模板帧第 5 层的输出特征图，可以发现目标感知特征在减少特征数量的同时，并没有削弱深层特征的表征能力。

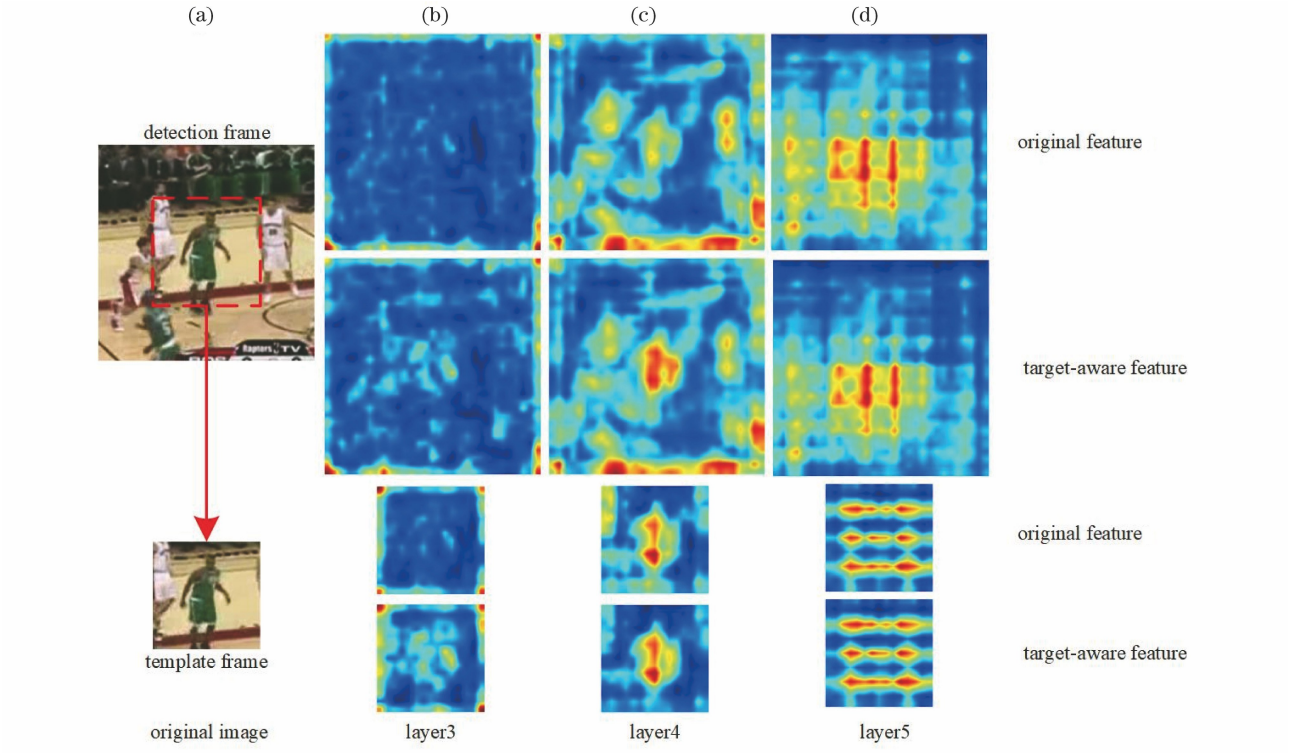

图 5　目标感知特征与原始特征对比实验结果。(a)原始图像；(b)第 3 层；(c)第 4 层；(d)第 5 层

Fig. 5　Visualized results of the original features and target-aware features. (a) Original image; (b) layer3; (c) layer4; (d) layer5

2.3　SiamRPN 模块

在 SiamRPN[12]中，为响应得分图的每个空间元素编码了一组 k 个锚点框提议(anchor box proposals)和其相对应的目标、背景的分类分数，以及边界框的回归量。SiamRPN 在并行输出每个提议的锚点框的分类分数和锚点框坐标回归量的基础上，根据各个锚点框的分类得分排名得到与目标边界框最接近的锚点框，并根据对应的锚点框坐标回归量得到最优的预测目标边界框。SiamRPN++[16]在原来 SiamRPN 的基础上，采用多层特征信息融合的策略，将特征提取网络 ResNet50 中第 3 层、第 4 层、第 5 层的输出特征分别送入 3 个 SiamRPN 模

块中，最后将 3 个 SiamRPN 模块的输出进行加权求和，得到更加精确的目标边界框。

单个 SiamRPN 模块具体结构如图 6 所示，由于第 3 层、第 4 层、第 5 层选定的特征通道数分别为 256、1024、1500，并不统一，因而引入自适应调整层，使用 1×1 的卷积核将目标感知特征的通道数统一为 256，方便后续 SiamRPN 模块对 3 层特征信息的分别处理；每个 SiamRPN 模块包含两个分支结构，一条分支用于目标和背景分类，另一条分支用于目标边界框的坐标回归，显而易见，两条分支要完成的任务不同，因而基于不同的任务所需要的特征也是不同的，所以通过模板分支和检测分支分别传递至两个非共享卷积层，不同的卷积层用于完成不同的分支任务，即学习相对应的特征，进而对具有相同数量通道的两个特征图逐通道进行相关操作（如图 6 中 Depthwise_Corr1、Depthwise_Corr2 所示），再添加一个卷积层以融合各个不同的通道输出。最后，添加一个卷积层用于 K 个锚点框的分类或坐标回归输出。

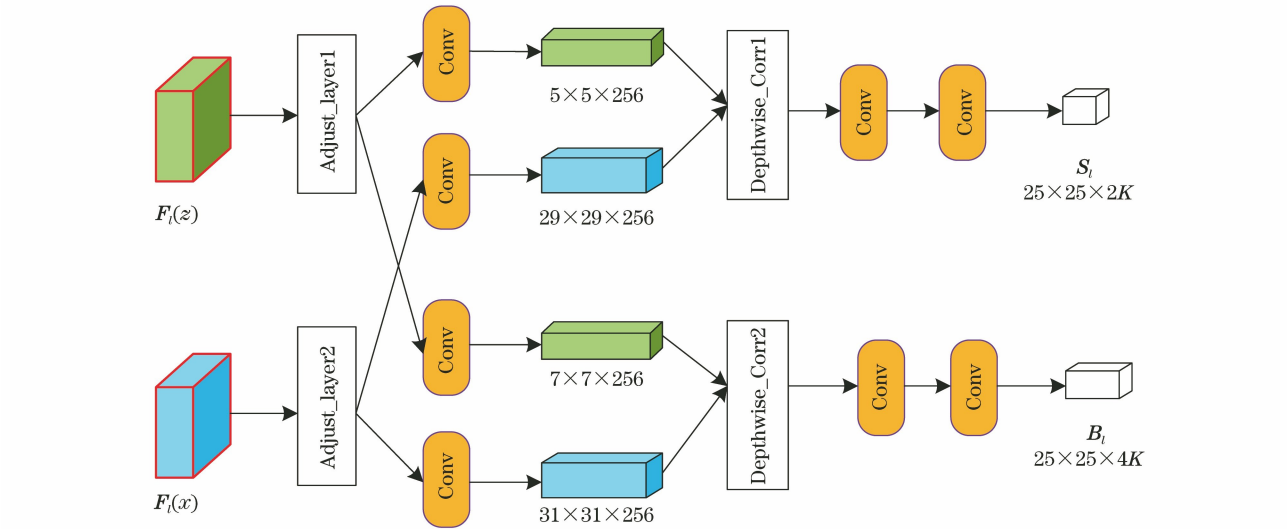

图 6　SiamRPN 模块具体结构图

Fig. 6　Specific structure diagram of SiamRPN module

在图 6 中，$\boldsymbol{S}_l$ 编码了第 l 层输出 $25\times25\times K$ 个锚点框的目标/背景的得分图，$\boldsymbol{B}_l$ 编码了第 l 层输出 $25\times25\times K$ 个锚点框的坐标值回归量 $[\Delta\boldsymbol{c}_x, \Delta\boldsymbol{c}_y, \boldsymbol{w}_{\text{anchor}}, \boldsymbol{h}_{\text{anchor}}]$，其中 $(\Delta\boldsymbol{c}_x, \Delta\boldsymbol{c}_y)$ 为锚点框中心坐标相对于上一帧中心位置的偏移量，$\boldsymbol{w}_{\text{anchor}}$ 和 $\boldsymbol{h}_{\text{anchor}}$ 为锚点框的宽和高。由于 3 个 SiamRPN 模块的输出具有相同的空间分辨率，故将这 3 个 SiamRPN 模块的输出进行加权求和，即

$$\boldsymbol{S}_{\text{all}}=\sum_{l=3}^{5}\alpha_l\cdot\boldsymbol{S}_l,\boldsymbol{B}_{\text{all}}=\sum_{l=3}^{5}\beta_l\cdot\boldsymbol{B}_l,\tag{5}$$

式中：α_l、β_l 分别为 3 个 SiamRPN 模块输出值 $\boldsymbol{S}_l$、$\boldsymbol{B}_l$ 的加权系数值。

根据 3 个 SiamRPN 模块的分类分支加权融合得到的 $25\times25\times2K$ 特征图，得到 $25\times25\times K$ 个锚点框（如图 7(a) 所示，这里选定 5 个锚点框，因而 $K=5$），其中，红色为锚点框，黄色为本文算法预测框，绿色为真实框，目标中心位置的 5 个锚点框如图 7(b) 所示。根据 RPN 原理，在这些生成锚点框（已给定空间位置信息和形状信息）的基础上，找到最接近目标（即 $\boldsymbol{S}_{\text{all}}$ 得分最高）的锚点框，再加上对应锚点框的坐标回归量（$\boldsymbol{B}_{\text{all}}$），从而得到最终的预测边界框。

为了得到更准确的得分图，使用余弦窗和尺寸大小变化惩罚项对 $25\times25\times K$ 个锚点框（如图 7(a) 中的红色框所示）的得分图（$\boldsymbol{S}_{\text{all}}$）进行重新排序。由于相邻帧之间目标边界框的尺寸大小变化和宽高比比率变化很小，因而引入一个惩罚项 $\boldsymbol{S}_{\text{penalty}}$ 来抑制较大的尺寸大小变化，其中，$\boldsymbol{S}_{\text{penalty}}$ 中每个元素为

$$S_{\text{penalty}}=\exp\left[-\kappa\left(\max\left\{\frac{r}{r'},\frac{r'}{r}\right\}\cdot\max\left[\frac{s}{s'},\frac{s'}{s}\right]-1\right)\right],\tag{6}$$

式中：κ 为一个超参数；r 为当前帧对应锚点框的宽高比比率 $\left(\frac{h_{\text{anchor}}}{w_{\text{anchor}}}\right)$；$r'$ 为前一帧目标预测框的宽高比比率 $\frac{h}{w}$；s、s' 为当前帧对应锚点框、上一帧目标预测边界框的等效总尺度，s 满足

$$s=\sqrt{(w+p)\times(h+p)},\tag{7}$$

其中，w、h 可分别为目标边界框或者锚点框的宽和高，p 等同于 $\frac{(w+h)}{2}$。

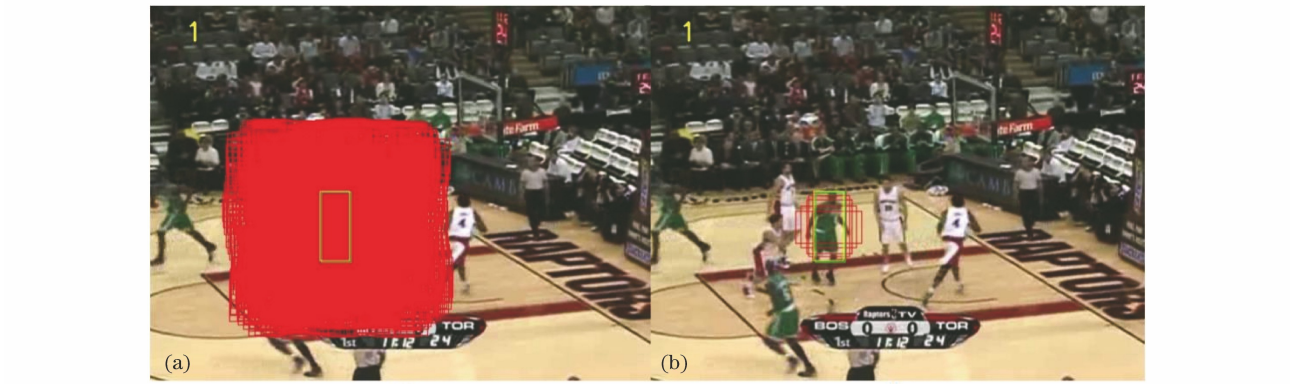

图 7　锚点框的可视化。(a) 25×25×K 个锚点框的可视化;(b) 目标中心位置 5 个锚点框的可视化

Fig. 7　Visualization of anchor bounding boxes. (a) Visualization of 25×25×K anchor bounding boxes; (b) visualization of 5 anchor bounding boxes at center position of target

在 25×25×K 个锚点框的得分图基础上与(6)式的惩罚项相乘,得到新的得分图:

$$\boldsymbol{S}_{\text{new}}=\boldsymbol{S}_{\text{penalty}}\times\boldsymbol{S}_{\text{all}},\tag{8}$$

这里,"×"为对应元素相乘。

由于目标在相邻两帧之间的变化不大,因而引入余弦窗(余弦窗口惩罚 $\boldsymbol{W}_{\text{cosine}}$)来抑制大的目标位置变化以排除干扰物。

$$\boldsymbol{S}_{\text{final}}=\boldsymbol{S}_{\text{new}}\cdot(1-\alpha_{\text{wi}})+\boldsymbol{W}_{\text{cosine}}\cdot\alpha_{\text{wi}},\tag{9}$$

其中:α_{wi} 为超参数,用于表征余弦窗口惩罚的作用程度。

基于(9)式,对新得到的 25×25×K 个锚点框得分图($\boldsymbol{S}_{\text{final}}$)执行非极大值抑制操作,找到得分最高($S_{\text{best}}$)的锚点框就找到了实际跟踪的目标位置,再加上对应锚点框的坐标回归量 $\boldsymbol{B}$ 就找到了当前帧的预测边界框。

最后对预测边界框进行平滑处理,平滑公式为

$$\begin{cases}w_{\text{final}}=w_{\text{last}}\cdot(1-\alpha_{\text{lr}})+w_{\text{current}}\cdot\alpha_{\text{lr}}\\ h_{\text{final}}=h_{\text{last}}\cdot(1-\alpha_{\text{lr}})+h_{\text{current}}\cdot\alpha_{\text{lr}}\end{cases},\tag{10}$$

式中:$\alpha_{\text{lr}}=S_{\text{best}}\cdot\alpha_{\text{LR}}$,这里 α_{LR} 为超参数;w_{last} 和 h_{last} 为上一帧目标边界框的宽度值和高度值;w_{current} 和 h_{current} 为当前帧预测边界框的宽度值和高度值;w_{final} 和 h_{final} 为最终目标边界框的宽度值和高度值。根据(10)式所示,得到最终精确的目标边界框(如图 7 中的黄色框所示)。

3　算法流程

本文提出基于目标感知特征筛选的孪生网络跟踪算法,其具体算法流程如图 8 所示。

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。

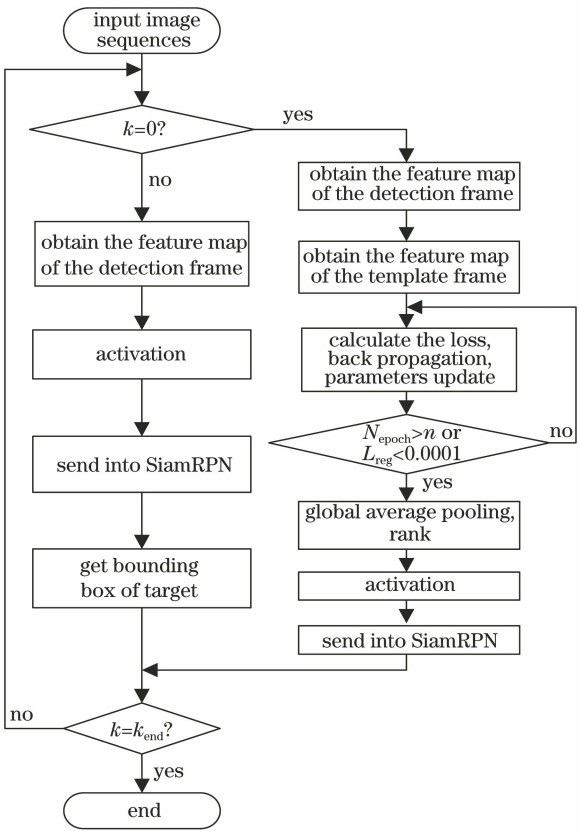

图 8　算法步骤

Fig. 8　Flow chart of proposed algorithm

2) 得到第 1 帧图像的检测帧特征图。根据第 1 帧中目标的中心位置和尺寸大小,裁剪得到 255×255 的检测帧图像,将其输入到特征提取网络中提取得到第 3 层的 31×31×512、第 4 层的 31×31×1024、第 5 层的 31×31×2048 特征图。

3) 得到第 1 帧图像模板帧的特征图。根据第 1 帧中目标的中心位置和尺寸大小,裁剪得到 127×127 的模板帧图像,将其输入到特征提取网络中提

取得到第 3 层的 15×15×512、第 4 层的 15×15×1024、第 5 层的 15×15×2048 特征图。在此，同时计算得到模板帧的特征图大小为 15×15。

4) 计算第 1 帧图像中目标感知模块的损失值、反向传播、参数更新。对第 1 帧图像检测帧的 31×31×512、31×31×1024、31×31×2048 特征图分别使用 7×7×512、7×7×1024、7×7×2048 的卷积核进行卷积得到 3 张 31×31×1 的特征图，再根据(1)式生成的高斯标签图，由(2)式计算损失值，再根据损失值计算反向传播的梯度值，进而更新给定 3 组卷积核的权重信息。

5) 重复步骤 4)，直到 N_{epoch} 满足要求，或者直到由(2)式得到的损失值满足收敛条件。

6) 全局平均池化、排序。对第 3 层、第 4 层、第 5 层按照(3)式对根据收敛损失函数反向传播生成的卷积核梯度值进行全局平均池化，池化结果作为衡量卷积核重要性程度的评判标准。对第 3 层、第 4 层、第 5 层中每个卷积核梯度值的全局平均池化结果按照从大到小进行排序，按照顺序分别选择 n_l 个在该层中比较重要的卷积核，这里 $n_3=256$，$n_4=1024$，$n_5=1500$。

7) 激活。将在步骤 6)中选定的卷积核设置为激活状态，提取得到模板帧第 3 层、第 4 层、第 5 层的目标感知特征。

8) 得到下一帧图像的检测帧特征图。以上一帧预测得到的目标中心位置为中心裁剪得到 256×256 的检测帧图像，将其输入到特征提取网络中提取得到第 3 层的 31×31×512、第 4 层的 31×31×1024、第 5 层的 31×31×2048 特征图，根据步骤 6)中选定的卷积核将其设置为激活状态，得到检测帧的目标感知特征。

9) 送入 SiamRPN 模块中。将模板帧生成的特征图和检测帧的特征图分别送入 SiamRPN 模块中，用于目标的辨识与定位，最终输出一个精确的目标边界框。

10) 继续步骤 8)、步骤 9)，直至该图像序列的目标跟踪结束。

4 实　验

为了验证本文算法的有效性，实验采用 OTB2015[6] 和 VOT2018[22] 标准跟踪数据集作为评估标准。OTB2015 包含 100 个目标跟踪图像序列，含有光照变化、目标尺度变化、目标遮挡、形变、运动模糊、快速运动、平面内旋转、平面外旋转、目标出视野、背景干扰、低分辨率 11 种跟踪难点，可对跟踪算法各种情况下的跟踪效果作出评价。VOT2018 包含 60 个具有更精细人工标注的目标跟踪图像序列，并且更具有挑战性。

4.1 实验平台

依据本文算法在台式机(装有 1 张 Nvidia GTX 1080ti GPU)上进行实验，台式机的操作系统为 64 位Ubuntu16.04，处理器为 Intel core(TM) i7-8700K，主频为 3.70 GHz，内存为 32 GB，编程环境为使用 PyTorch 的 python3.7。

4.2 实验参数设置

在实际的应用过程中，对于不同的数据集，需要采用不同的参数设置才能获取更大的性能增益。因此，算法的具体参数设置需根据不同的数据集重新配置。为了提高算法应用的适应性，总结了较具体的超参数搜索算法和参数影响规律。

4.2.1 超参数设置

针对不同数据集 OTB2015、VOT2018，对(6)式中的 κ、(9)式中的 α_{wi}、(10)式中的 α_{LR} 设置了两组不同的超参数。

参考文献[16]，采用网格搜索的超参数搜索算法，通过循环遍历，尝试每一种可能性，在 OTB2015 数据集上进行评估，以跟踪成功率和跟踪精度为评价指标，将表现最好的参数保留，表现差的参数丢弃，逐步缩小搜索区间，最后得到表现最好的一组参数作为最终的结果。超参数寻优初始化区间设置如下：

1) 对于 OTB2015 数据集设置区间：κ 为[0.1，0.5)，α_{wi} 为[0.1，0.5)，α_{LR} 为[0.1，0.5)，步长为 0.1；

2) 对于 VOT2018 数据集设置区间：κ 为[0.1，0.6)，α_{wi} 为[0.1，0.6)，α_{LR} 为[0.1，0.6)，步长为 0.1。

在初步的超参数搜索之后再根据最好的评估结果逐步缩小区间，缩短步长，最终找到效果最好的参数设置：对于 OTB2015 数据集，设置 $\kappa=0.12$，$\alpha_{\mathrm{wi}}=0.12$，$\alpha_{\mathrm{LR}}=0.48$；针对 VOT2018 数据集，设置 $\kappa=0.13$，$\alpha_{\mathrm{wi}}=0.30$，$\alpha_{\mathrm{LR}}=0.48$。

4.2.2 目标感知模块优化参数设置

关于目标感知模块优化参数设置，理论上讲，对于不同的数据集，不同的参数设置所带来的性能是不一样的，因此通过大量的实验总结出其中的客观规律：在(1)式生成高斯标签图中，参考文献[19]，σ 设置为 1.5；在 2.2.1 小节中，对于(2)式构造的损失函数，λ 为正则化系数，是权衡范数惩罚项 $\|\boldsymbol{W}\|^2$ 和标准目标函数 $\|\boldsymbol{Y}-\boldsymbol{W}*\boldsymbol{X}_n^l\|^2$ 相对贡献的超参数，λ

值越大，对应正则化惩罚越大，使权重 $\boldsymbol{W}$ 更加接近原点（趋向于 0），在这里，需要根据 $\boldsymbol{W}$ 的值对其进行全局平均池化来衡量相应卷积核的重要性程度。为了不改变网络学习 $\boldsymbol{W}$ 的原始特性，设置的值相对来说越小越好，在这里，设置为 0.0001。使用带动量（momentum）的随机梯度下降法（Stochastic Gradient Descent，SGD）进行优化，在实际的优化过程中，随着特征提取的深度增加，优化的难度逐步增大，因而针对不同层提取的特征使用不同的优化策略；在实际的调整动量（M）和学习率（R）的实验中，发现增大动量和学习率会加速损失函数的收敛，但也会容易引起梯度爆炸，第 5 层相比于第 3 层、第 4 层更易发生梯度爆炸；基于此，本文采取的策略是在各层损失函数不发生爆炸的前提下，选择尽量大的动量参数和学习率。在第 1 帧图像的初始训练阶段，当损失值（L_{reg}）低于 0.0001 或者达到最大迭代次数（N_{epoch}）300 时，认为训练的网络满足了收敛的条件。通过大量的实验，参数设置如下：对于第 3 层的特征，$N_{\mathrm{epoch}}=300$，$R=5\times10^{-4}$，$M=0.98$；对于第 4 层的特征，$N_{\mathrm{epoch}}=300$，$R=5\times10^{-4}$，$M=0.98$；对于第 5 层的特征，$N_{\mathrm{epoch}}=300$，$R=4.92\times10^{-5}$，$M=0.90$。

针对特征提取模块中第 3 层、第 4 层、第 5 层提取得到的特征图使用目标感知模块分别筛选得到 256、1024、1500 个对当前目标比较敏感的卷积核。关于这 3 个参数的具体讨论见 4.3.2。

上述设置中没有提到的参数，沿用文献[19]中的参数设置。以下实验如不特殊说明，均采用此参数设置。

4.3　对比实验

4.3.1　模板帧特征图的获取方式

本文分别使用直接方式和间接方式获取两种模板帧特征图（2.1.2 中模板帧 1 和模板帧 2），将其代入本文算法中，在 OTB2015 数据集上进行算法跟踪实验，结果见表 1 所示。

表 1　在 OTB2015 数据集上两种模板帧特征图的实验结果对比

Table 1　Comparison of experimental results of two template frame feature maps on the OTB2015 dataset

Frame	Success	Precision
Template frame 1	0.661	0.878
Template frame 2	0.547	0.786

可以发现通过直接方式获取模板帧特征图相比于间接方式，跟踪成功率提升 0.114，跟踪精度提升 0.092。因而本文选择使用直接方式获取模板帧特征图。

4.3.2　三层特征通道的筛选保留数量

在实验中发现，大幅度减少第 3 层、第 5 层特征图的卷积核数量，在 OTB2015、VOT2018 数据集的最终评估指标不会发生明显的精度损失，然而对于第 4 层特征，大幅度减少其卷积核数量，在整个 OTB2015、VOT2018 数据集会造成严重的精度损失。因此在 OTB2015 数据集上对比分析第 4 层具有不同数量的重要特征通道时的实验结果，如表 2 所示。这里以 Basketball、Bird2 等 15 个视频跟踪序列为例，直接采用文献[16]中的参数设置 $\kappa=0.24$，$\alpha_{\mathrm{wi}}=0.50$，$\alpha_{\mathrm{LR}}=0.25$。表 2 中：$n_3$，$n_4$，$n_5$ 代表第 3 层、第 4 层、第 5 层筛选得到的筛选保留通道数量。这里，固定第 3 层的输出重要通道数为 256，固定第 5 层的筛选保留通道数为 1500，分别改变第 4 层的输出重要通道数，观察各个视频跟踪序列的跟踪成功率和跟踪精度变化情况。

如表 2 所示，在 OTB2015 标准数据集上，以跟踪成功率和跟踪精度作为评价指标，对于不同的视频跟踪序列，使用本文算法所筛选得到的不同数量的重要特征通道带来的增益是不同的。以视频跟踪序列 Bird2 为例，使用（n_3，n_4，n_5）=（256，512，1500）的重要特征通道数，相比于第 4 层的全部特征通道保留（n_3，n_4，n_5）=（256，1024，1500），跟踪成功率提升 0.01，跟踪精度提升 0.002，相比于 SiamRPN++算法，跟踪成功率提升 0.081，跟踪精度提升0.079，这说明对于特定数据集，保留特征通道数量并非越多越好。对于不同的视频跟踪序列，提取不同数量的重要特征通道所带来的跟踪成功率和跟踪精度上的最大提升程度是不同的，这说明本文引入目标感知模块筛选参数大小对目标集有依赖性。n_l 的重要意义在于筛选出用于跟踪目的的有效特征，剔除掉干扰噪声特征。由于不同的视频跟踪序列包含用于跟踪辨识的有效特征数量并不一致，包含的干扰噪声特征数也不相同，因此对于不同的视频跟踪序列，该参数的选取需要具体问题具体分析，以期在不同的跟踪视频序列中获得最大的性能增益。n_l 可以看作超参数，因此需要进行相应的超参数搜索实验。实验准则为以整个数据集或者单个视频序列为测试对象（这里以 OTB100 整个数据集为测试对象），以跟踪精度和跟踪成功率为评价指标，采用控制变量法，固定三层中任意两层特征通道保留个数，

表 2　在 OTB2015 数据集改变第 4 层中重要特征通道数量 n_4 的实验结果对比($n_3=256$,$n_5=1500$)

Table 2　Comparison of experimental results for changing the number of important feature channels n_4 in layer 4 on the OTB2015 dataset ($n_3=256$, $n_5=1500$)

Video sequence	Success, precision				SiamRPN++
	$n_4=512$	$n_4=650$	$n_4=800$	$n_4=1024$	
Basketball	0.420,0.532	0.432,0.548	**0.436**,0.557	0.425,**0.584**	**0.446**,0.562
Bird2	**0.708**,**0.827**	0.701,0.825	0.688,0.813	0.698,0.825	0.627,0.748
Bird1	0.176,0.497	0.211,**0.560**	0.203,0.448	**0.236**,0.502	0.204,0.367
Bolt	0.261,0.335	0.259,0.337	0.257,0.340	**0.650**,**0.883**	0.644,**0.887**
Girl2	**0.614**,0.720	0.586,0.685	0.557,0.645	0.579,0.679	**0.634**,**0.720**
Car4	0.838,0.953	**0.850**,**0.954**	0.849,**0.954**	0.847,0.951	**0.869**,0.953
ClifBar	0.316,0.396	**0.605**,**0.836**	0.577,0.819	0.567,0.795	0.524,0.718
Dancer	**0.779**,**0.871**	0.763,0.852	0.735,0.829	0.741,0.831	0.768,0.861
DragonBaby	0.626,0.748	0.629,**0.750**	0.630,0.743	**0.630**,0.747	**0.681**,**0.830**
FaceOcc1	**0.637**,0.534	0.636,0.528	0.608,**0.560**	0.621,0.559	0.604,0.486
Freeman3	0.811,0.954	0.814,0.955	**0.816**,**0.955**	0.815,0.954	0.809,**0.960**
Human2	0.743,0.773	0.756,0.782	0.768,0.799	**0.782**,**0.806**	0.775,**0.817**
Jumping	0.550,0.804	0.600,0.840	**0.610**,**0.845**	0.578,0.816	**0.670**,**0.882**
Liquor	**0.750**,**0.814**	0.708,0.770	0.701,0.764	0.607,0.653	0.616,0.661
Suv	**0.745**,**0.901**	0.736,0.898	0.679,0.882	0.434,0.519	0.649,0.802
Woman	0.668,0.901	**0.673**,**0.903**	0.670,0.899	0.650,0.890	0.613,**0.906**

改变剩余一层特征通道保留个数，观察实验结果变化情况，从而选择相对最优的特征通道保留个数(选取原则为在跟踪成功率和跟踪精度损失不大的前提下，使得特征通道保留个数尽量小)。对于 OTB2015 和 VOT2018 数据集来讲，本文发现使用(n_3,n_4,n_5)=(256,1024,1500)的综合指标最好，因而对于 OTB2015 和 VOT2018 数据集采用(n_3,n_4,n_5)=(256,1024,1500)的参数设置。

4.4　实验结果与分析

4.4.1　OTB2015 实验

为了保证实验结果的客观性，对于 OTB2015 数据集的 100 组跟踪图像序列，引入近几年热门并且跟踪性能优异的 SiamFC[10]、SiamRPN[12]、DaSiamRPN[14]、SiamRPN++[16]、TADT[19]、ECO[23]、UPDT[24]等 14 种跟踪算法，采用一次跟踪评估(OPE)中的跟踪成功率、跟踪精度、运行速度作为跟踪算法评价指标，对这些跟踪算法的性能进行比较，并且对于能够实现实时跟踪的算法采用一次性评估生成成功率图和精确度图。

4.4.1.1　整体性能分析

表 3 展示了两种不同类别的目标跟踪算法(孪生网络跟踪算法和基于相关滤波器的跟踪算法)在 OTB2015 数据集上进行评估得到的跟踪成功率(Success)、跟踪精度(Precision)和运行速度(V_{FPS})。在实时的跟踪算法($V_{FPS}>25$)中，本文的跟踪算法实现在 OTB2015 数据集上排名第二，实现了较好的性能。

表 3　在 OTB2015 数据集上实验结果对比

Table 3　Comparison of experimental results on the OTB2015 dataset

Tracker name	Success	Precision	V_{FPS}
SiamRPN++	0.695	0.905	35
Ta-SiamRPN++	0.661	0.878	36
SiamRPN	0.643	0.860	71
RASNet[11]	0.642		83
SA-Siam[25]	0.657	0.865	50
CFNet[26]	0.568	0.748	75
SiamFC	0.582	0.771	49
TADT	0.647	0.839	34
DaSiamRPN	0.658	0.880	97
BACF[27]	0.617	0.815	35
ECO	0.694	0.910	3
UPDT	0.702		0.4
STRCF[28]	0.683		3

与孪生网络跟踪算法相比，本文算法的跟踪成功率和精度较高的原因除了前文理论分析外，还包括本文算法中特征提取模块提取了更加丰富有效的特征表征，以及 RPN 模块对目标位置、尺度进行了有效的搜索。结合本文算法提出的目标感知模块，筛选对当前目标敏感的深度特征，充分利用当前被跟踪目标的外观和语义信息，对于目标的外观变化和尺度变化具有一定的鲁棒性。

与当前跟踪性能最好的 SiamRPN＋＋算法相比，两种算法的跟踪成功率与精度仅仅相差 3%，然而在特征提取模块中的第 3 层和第 5 层特征图输出中，本文算法的卷积核使用率分别仅为 SiamRPN＋＋的 50% 和 73%，可以看出使用目标感知模块进行激活的卷积核足以用于对当前跟踪目标的辨识与定位，并且实时性要比 SiamRPN＋＋算法更好。

基于相关滤波器的跟踪算法（ECO[23] 和 UPDT[24]）得益于在线更新和多特征融合方案的优势，在所有被比较的跟踪器中均获得了最佳性能。虽然这些跟踪算法均已获得良好的跟踪成功率和跟踪精度，但是它们的在线训练过程是十分耗时的，并且训练得到的模型容易发生过拟合，这也直接导致了这些算法不能达到实时的效果，精度很难再有提高。而本文算法依托于简洁的孪生网络框架和相对较少的深层特征表示，可以实现实时的跟踪速度（36 frame/s）。由于孪生网络跟踪框架的性能很大程度上取决于所提取特征的判别能力，故本文算法所提出的目标感知功能有效。

图 9 显示了本文算法相对于其他性能优异的实时跟踪算法的性能对比结果。为了简洁，在此图中仅显示了实时的跟踪算法，其他跟踪算法的完整结果可以在表 3 中找到。

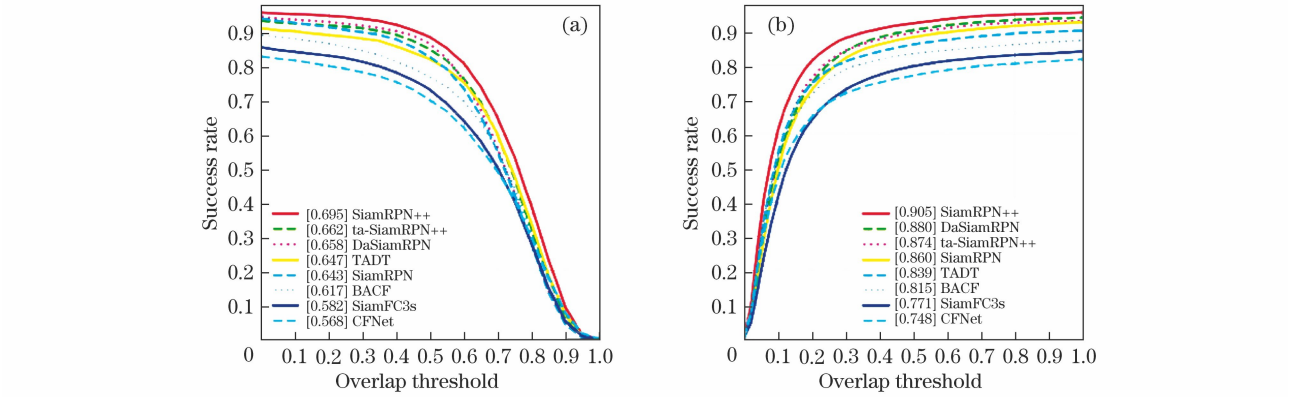

图 9　实时跟踪算法在 OTB100 数据集中的 OPE 成功率和精确度对比图。(a) 成功率图；(b) 精确度图

Fig. 9　Comparison of success rate and precision plots of OPE for realtime trackers on the OTB100 dataset. (a) Success rate plot; (b) precision plot

4.4.1.2　基于数据集属性的性能分析

为了充分评估算法的跟踪性能，利用 OTB2015 跟踪数据集的 11 个属性进一步评估算法的性能。图 10 为 10 种跟踪算法在包含 11 个属性数据集上的成功率图。通过分析曲线可以看出，在 11 个跟踪难点属性中，本文算法在平面内旋转、尺度变化、低分辨率、平面外旋转、超出视野、光照变化、快速移动、运动模糊和尺度变化条件下取得了较好的成绩，分别达到了 0.667、0.651、0.631、0.638、0.601、0.685、0.652、0.655，与 SiamRPN＋＋算法相比，跟踪成功率偏差分别为：－0.033、－0.043、－0.036、－0.045、－0.023、－0.029、－0.026、－0.037，与SiamRPN＋＋算法性能差距不大，但计算量降低很多。相比于其他性能良好的孪生网络目标跟踪算法，本文算法在不同条件下的跟踪成功率均有较大的提高。从图 10 可以看出，通过目标感知模块提取对当前目标敏感特征是有效的，所筛选特征具有对当前被跟踪目标充分表征的能力。

4.4.1.3　定性分析

为了可视化显示实际的跟踪效果，从 OTB2015 数据集中选取 6 个具有各种跟踪难点的视频序列，与 SiamRPN＋＋、DaSiamRPN、SiamRPN、SiamFC、TADT 5 个跟踪算法进行对比验证实验，如图 11 所示。

对于 Basketball 测试视频的跟踪，其主要难点在于相似目标的干扰。参与评测的跟踪算法在第 614 帧产生分化，此时目标周围有较为密集的相似目标干扰，并且有部分相似的背景对目标存在部分重叠，可以发现，这些相似干扰主要为具有相同类别的人与人干扰。SiamRPN＋＋、SiamRPN、TADT

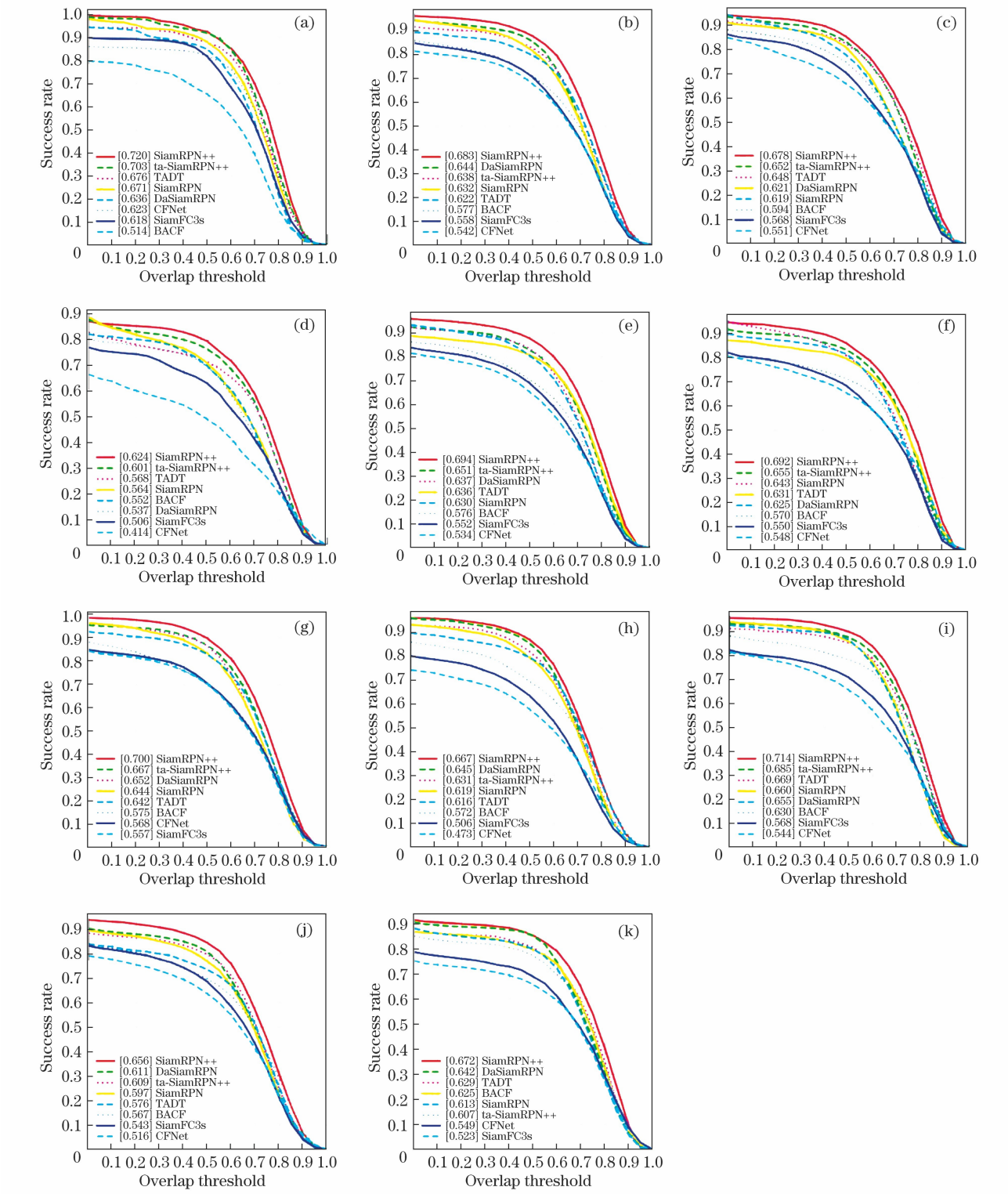

图 10　8 种跟踪算法在 11 个属性的成功率图。(a) 低分辨率；(b) 平面外旋转；(c) 快速移动；(d) 出视野；(e) 尺度变化；(f) 运动模糊；(g) 平面内旋转；(h) 形变；(i) 光照变化；(j) 遮挡；(k) 背景干扰

Fig. 10　Success rate plots with 11 different attributes for 8 trackers. (a) Low resolution; (b) out of plane rotation; (c) fast motion; (d) out of view; (e) scale variation; (f) motion blur; (g) in-plane rotation; (h) deformation; (i) illumination variation; (j) occlusion; (k) background clutters

直接跟丢了目标，而本文算法由于加入目标感知模块，对特征提取模块的通用特征进行进一步筛选，得到对当前目标敏感的特征，故在面对较多相似的具有类内变化的干扰目标时依旧能够达到优

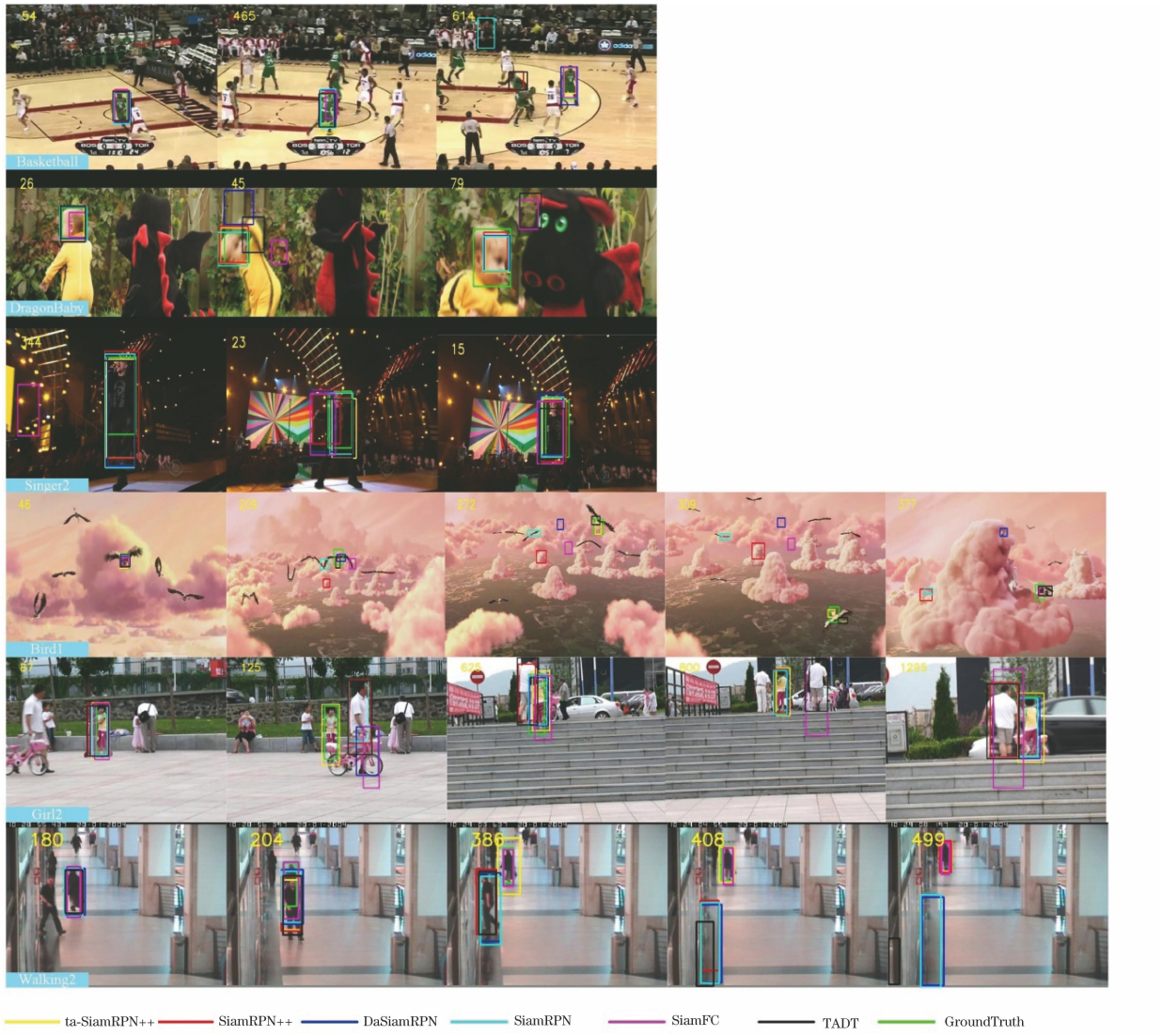

图 11　各个跟踪算法在不同属性的视频中的实际跟踪结果图

Fig. 11　Actual tracking results of each algorithm for videos with different attributes

异的效果。

对于 Girl2 测试视频的跟踪，其主要难点在于背景中的人物对于目标有遮挡，且存在相似干扰。在第 125 帧，出现相似的遮挡，即一个男人遮挡住了小女孩，这里男人和小女孩同时具有人这个类别标签，可以认为是类内相似目标干扰。SiamRPN＋＋、SiamRPN 和 TADT 将遮挡之后的背景人物作为跟踪对象，直接跟丢了目标，与此同时，SiamFC、DaSiamRPN 也直接跟错目标，而本文算法在相似遮挡出现之后依然能够找回被跟踪目标，由此看出本文算法在具有相同类别的相似目标遮挡的情况下依旧能够拥有十分出色的跟踪结果；在第625 帧中，背景中人物与被跟踪目标距离较近，跟踪过程中存在较强的具有类内变化的相似干扰，SiamRPN＋＋和 TADT 直接跟丢目标，而本文算法仍然精准地定位目标，由此可以看出，本文算法在具有较强类内变化的相似目标干扰的情况下仍能表现出很好的跟踪效果。

对于 Bird1 测试视频的跟踪，其主要难点在于目标的形变变化较大、移动快速且出视野。在第 130 帧，跟踪目标出视野，直到第 182 帧，跟踪目标重新回归，此时只有本文算法和 TADT 能够准确跟踪到目标，其他跟踪器均不同程度地偏离目标。在第 272 帧，由于目标快速移动且形变量较大，DaSiamRPN 直接跟丢目标，TADT 也逐渐偏离目标，而本文算法依然能够精准地定位目标，由此可以看出，本文算法在目标快速移动、形变变化较大的情况下依然能较好地跟踪目标。

对于 DragonBaby 测试视频的跟踪，其主要难点在于目标快速移动、尺度变化较大并且存在平面

内、平面外旋转。在第 45 帧，目标快速移动并且进行了平面内、外旋转，此时 DaSiamRPN、SiamFC、TADT 均跟丢目标。在第 79 帧，由于目标尺度变化较大，导致 SiamFC、TADT 均跟丢目标。而本文算法在以上情况均能稳健地跟踪目标，说明本文算法对于目标平面内旋转、平面外旋转、尺度变化较大也有很好的跟踪效果。

另外，在进行 OTB2015 各个视频的评估过程中，以跟踪成功率和跟踪精度作为评价标准，通过对比发现，本文算法在这两个基准上有包括 Basketball、Bird1、Bird2、Girl2、Mhyang、MountainBike 等 29 个评估视频超过了 SiamRPN＋＋，可见，本文算法在一些跟踪场景中，优于 SiamRPN＋＋，并且在 60 多个评估视频中与 SiamRPN＋＋相比并无较大差异（在 10％以内），这说明通过目标感知模块对特征提取网络提取的目标特征进行筛选，得到对当前目标敏感的特征，能够提取对当前目标有效且充分的特征表示。

4.4.2　VOT2018 实验

同样，对于 VOT2018 标准数据集的 60 组跟踪图像序列，引入近几年热门并且具有代表性的 SiamFC、SiamRPN、DaSiamRPN、SiamRPN＋＋、UPDT、SA_Siam_R[29] 等 13 种跟踪算法，采用期望重叠率（EAO）、准确率（Accuracy）、鲁棒性（Robustness）这三个评价指标对 10 种性能优异的跟踪算法进行了性能比较，如表 4 所示。

表 4　在 VOT2018 数据集上实验结果对比

Table 4　Comparison of experimental results on the VOT2018 dataset

Tracker name	Accuracy	Robustness	EAO	Lostnumber	V_{FPS}
SiamRPN＋＋	0.601	0.234	0.415	50	35
DaSiamRPN	0.586	0.276	0.383	59	59
UPDT	0.536	0.184	0.378	39	0.4
RCO[30]	0.507	0.155	0.376	33	0.8
DeepSTRCF[31]	0.523	0.215	0.345	46	3
SA_Siam_R	0.566	0.258	0.337	55	32
SiamVGG[32]	0. 531	0.318	0.286	68	29
ECO	0.484	0.276	0.280	59	4
DSiam[33]	0.512	0.654	0.196	138	10
SiamFC	0.503	0.585	0.187	125	32
DCFNet[34]	0.470	0.543	0.182	116	27
DensSiam[35]	0.462	0.688	0.174	147	19
Ta-SiamRPN＋＋	0.593	0.272	0.360	58	36

表 4 显示了 VOT2018 数据集上的实验结果。相对于该数据集上的最新跟踪算法，本文提出的跟踪算法具有良好的性能。以较高的准确性（0.593）和相对较好的鲁棒性（0.272），获得了排名第二的 EAO（0.360）。在 EAO 这一评价指标上，虽然本文算法的与性能最好的 SiamRPN＋＋相差 5％，但是在精度这一评价指标上只与 SiamRPN＋＋相差 0.07％，说明本文算法已经实现了与 SiamRPN＋＋十分接近的准确率，这也进一步说明了提取对目标敏感的深层特征信息，有助于区分目标和背景。

5　结　论

目前，基于深度学习的目标跟踪算法使用离线训练好的特征提取网络进行目标辨识，这些网络基本来自于对象识别任务，并不适用于泛化性要求高的跟踪问题。为此，本文提出了基于目标感知进行特征筛选的孪生网络跟踪框架，包括特征提取模块、目标感知模块、SiamRPN 模块。在目标感知模块中，针对特征提取模块中提取得到的通用特征信息，设计了特征筛选策略，筛选得到对当前被跟踪目标重要的特征通道信息，在提高跟踪成功率和跟踪精度的同时，减少了计算量。利用 OTB2015 和 VOT2018 两个标准数据集作为评估基准进行测试，分别与当前最先进的多种算法进行了对比，验证了本文所提出算法的有效性。

参　考　文　献

[1] Lee K H, Hwang J N. On-road pedestrian tracking across multiple driving recorders[J]. IEEE Transactions on Multimedia, 2015, 17(9): 1429-1438.

[2] Tang S Y, Andriluka M, Andres B, et al. Multiple people tracking by lifted multicut and person Re-identification[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE, 2017: 3539-3548.

[3] Teutsch M, Kruger W. Detection, segmentation, and tracking of moving objects in UAV videos[C]// 2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, September 18-21, 2012, Beijing, China. New York: IEEE, 2012: 313-318.

[4] Smeulders A W M, Chu D M, Cucchiara R, et al. Visual tracking: an experimental survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(7): 1442-1468.

[5] Yang H X, Shao L, Zheng F, et al. Recent advances and trends in visual tracking: a review[J].

Neurocomputing, 2011, 74(18): 3823-3831.

[6] Wu Y, Lim J, Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.

[7] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.

[8] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 4293-4302.

[9] Song Y B, Ma C, Wu X H, et al. VITAL: Visual tracking via adversarial learning[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT. New York: IEEE, 2018: 8990-8999.

[10] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional Siamese networks for object tracking[M]// Hua G, Jégou H. Computer Vision-ECCV 2016 Workshops. Lecture Notes in Computer Science. Cham: Springer, 2016, 9914: 850-865.

[11] Wang Q, Teng Z, Xing J L, et al. Learning attentions: residual attentional Siamese network for high performance online visual tracking[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT. New York: IEEE, 2018: 4854-4863.

[12] Li B, Yan J J, Wu W, et al. High performance visual tracking with Siamese region proposal network[C]// 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT. New York: IEEE, 2018: 8971-8980.

[13] Ren S Q, He K M, 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, 2017, 39(6): 1137-1149.

[14] Zhu Z, Wang Q, Li B, et al. Distractor-aware Siamese networks for visual object tracking[M]// Ferrari V, Hebert M, Sminchisescu C, et al. Computer Vision-ECCV 2018. Lecture Notes in Computer Science. Cham: Springer, 2018: 103-119.

[15] Zhang Z P, Peng H W. Deeper and wider Siamese networks for real-time visual tracking[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 4591-4600.

[16] Li B, Wu W, Wang Q, et al. SiamRPN++: evolution of Siamese visual tracking with very deep networks[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 4282-4291.

[17] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 770-778.

[18] Wang Q, Zhang L, Bertinetto L, et al. Fast online object tracking and segmentation: a unifying approach[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 1328-1338.

[19] Li X, Ma C, Wu B Y, et al. Target-aware deep tracking[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE, 2019: 1369-1378.

[20] Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: visual explanations from deep networks via gradient-based localization[C]// 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice. New York: IEEE, 2017: 618-626.

[21] Zhou B L, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 2921-2929.

[22] Kristan M, Leonardis A, Matas J, et al. The sixth visual object tracking vot2018 challenge results[C]// Proceedings of the European Conference on Computer Vision (ECCV), September 8-14, 2018, Munich, Germany. Cham: Springer, 2018: 3-53.

[23] Danelljan M, Bhat G, Khan F S, et al. ECO: efficient convolution operators for tracking[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE, 2017: 6638-6646.

[24] Bhat G, Johnander J, Danelljan M, et al. Unveiling the power of deep tracking[C]// Proceedings of the European Conference on Computer Vision (ECCV), September 8-14, 2018, Munich, Germany. Cham: Springer, 2018: 483-498.

[25] He A, Luo C, Tian X, et al. A twofold siamese network for real-time object tracking[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition (CVPR), June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 4834-4843.

[26] Valmadre J, Bertinetto L, Henriques J F, et al. End-to-end representation learning for correlation filter based tracking[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE, 2017: 5000-5008.

[27] Kiani Galoogahi H, Fagg A, Lucey S. Learning background-aware correlation filters for visual tracking[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI. New York: IEEE, 2017: 1135-1143.

[28] Li F, Tian C, Zuo W, et al. Learning spatial-temporal regularized correlation filters for visual tracking[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 4904-4913.

[29] He A, Luo C, Tian X, et al. Towards a better match in Siamese network based visual object tracker[C]// Proceedings of the European Conference on Computer Vision (ECCV), September 8-14, 2018, Munich, Germany. Cham: Springer, 2018: 132-147.

[30] Zhang X M, Wang M G. Robust visual tracking based on adaptive convolutional features and offline Siamese tracker[J]. Sensors, 2018, 18(7): 2359.

[31] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking[C]// Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 58-66.

[32] Li Y, Zhang X. SiamVGG: visual tacking using deeper siamese networks[J/OL]. (2019-02-07)[2019-12-08]. https: //arxiv.org/abs/1902.02804.

[33] Guo Q, Feng W, Zhou C, et al. Learning dynamic Siamese network for visual object tracking[C]// 2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice. New York: IEEE, 2017: 1763-1771.

[34] Wang Q, Gao J, Xing J, et al. Dcfnet: discriminant correlation filters network for visual tracking[J/OL]. (2017-04-13)[2019-12-08]. https: //arxiv.org/abs/1704.04057.

[35] Abdelpakey M H, Shehata M S, Mohamed M M. Denssiam: end-to-end densely-siamese network with self-attention model for object tracking[C]// International Symposium on Visual Computing(ISVC), November 19-21, 2018, Las Vegas, Nevada, USA. Cham: Springer, 2018: 463-473.