在线目标分类及自适应模板更新的孪生网络跟踪算法

陈志旺[1]，张忠新[1]，宋娟[2]，雷海鹏[1]，彭勇[3]

（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. 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；
3. 燕山大学电气工程学院，河北 秦皇岛 066004）

摘 要：针对孪生网络跟踪算法在离线训练阶段学习被跟踪目标和其他对象的嵌入式特征，而这些特征缺少特定于目标的上下文信息，使跟踪算法的稳健性较差的问题，以 SiamRPN++作为基准算法，提出了在线目标分类及自适应模板更新的孪生网络跟踪算法。首先，在离线训练阶段设计了互相关特征图监督模块，以学习更具判别力的嵌入式特征；其次，在线跟踪阶段设计了包含注意力机制的在线目标分类模块，在该模块中使用在线滤波器更新策略滤除背景噪声干扰；最后，设计了一种自适应模板更新模块，使用 UpdateNet 更新目标模板信息。在 VOT2018、VOT2019 这 2 个标准数据集上的实验结果验证了所提算法的有效性，相比基准算法 SiamRPN++分别带来 13.5%和 18.2%（EAO）的性能提升。

关键词：机器视觉；目标跟踪；孪生网络；目标分类；自适应模板更新

中图分类号：TP391.4

文献标识码：A

DOI: 10.11959/j.issn.1000−436x.2021127

Tracking algorithm of Siamese network based on online target classification and adaptive template update

CHEN Zhiwang[1], ZHANG Zhongxin[1], SONG Juan[2], LEI Haipeng[1], PENG Yong[3]

1. Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China
2. Jiamusi Electric Power Company, State Grid Heilongjiang Electric Power Co., Ltd., Jiamusi 154002, China
3. School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China

Abstract: Aiming at the problem that tracking algorithm of Siamese network learned the embedded features of the tracked target and the object in the offline training stage, and these embedded features often lacked the target-specific context information, which made these tracking algorithms less robust, a tracking algorithm of the Siamese network based on online target classification and adaptive template update was proposed, which used SiamRPN++ as the baseline algorithm. Firstly, a cross-correlation feature map supervision module for classification was designed in the offline training phase to learn more discriminative embedded features. Secondly, an online target classification module that included an attention mechanism in the online tracking phase was designed, and the online update filter strategy in the module was used to filter out the background noise. Finally, an adaptive template update module was designed to update the target template information using the UpdateNet. The results of experiments on VOT2018 and VOT2019 datasets verify the effectiveness of the proposed algorithm, which brings 13.5% and 18.2% (EAO) improvement respectively compared with the baseline algorithm SiamRPN++.

Keywords: machine vision, object tracking, Siamese network, object classification, adaptive template update

收稿日期：2021−02−18；**修回日期**：2021−04−10

基金项目：国家自然科学基金资助项目（No.61573305）；河北省自然科学基金资助项目（No.F2019203511）

Foundation Items: The National Natural Science Foundation of China (No.61573305), The Natural Science Foundation of Hebei Province (No.F2019203511)

1 引言

视觉对象跟踪是计算机视觉任务的一个主要分支，具有重要的理论研究意义和应用价值，在车辆视觉导航系统、智能人机交互、智能视频监控系统和智能交通等方面具有广泛应用。简而言之，视觉对象跟踪旨在给定任意感兴趣目标在某一视频图像序列的第一帧中位置和形状信息的前提下，在后续帧中预测被跟踪目标的实际位置和形状大小。

解决视觉对象跟踪问题的方法主要可以分为两类：生成式跟踪算法和判别式跟踪算法。生成式跟踪算法在当前帧中对目标区域进行建模，在下一帧中寻找与模型匹配最相似的区域，从而确定该区域为预测目标位置。判别式跟踪算法将目标跟踪问题转化为一个关于目标和背景的二分类问题，通过训练一个分类器以将目标与背景区分开，从而找到预测目标位置。近年来，随着深度学习的发展，由于基于深度学习的判别式跟踪算法通过深度卷积神经网络学习到的特征具有很强的辨别性并且具有稳健的效果，因此判别式跟踪算法逐渐成为视觉对象跟踪领域中的主流方法。

判别式跟踪算法中具有代表性的是基于相关滤波类跟踪算法。其首先在第一帧中提取模板目标图像特征作为滤波器模板；然后利用后续帧的图像与滤波器模板做相关性卷积，计算后续帧图像不同部分的响应值；最后将具有最大响应值对应的部分作为跟踪的结果，使目标跟踪算法在跟踪精度和速度上均获得了显著提升。其中，比较典型的算法包括最小均方误差输出和（MOSSE, minimum output sum of squared error）滤波器[1]、基于核相关滤波器（KCF, kernelized correlation filter）[2]的目标跟踪算法、空间正则化的判别式相关滤波器（SRDCF, spatially regularized discriminative correlation filter）跟踪算法[3]、基于有效卷积运算目标跟踪（ECO, efficient convolution operator for tracking）算法[4]。

除了相关滤波类跟踪算法，随着深度学习技术的发展，基于孪生网络的跟踪算法由于其在保证实时速度运行的前提下在各种基准跟踪数据集测试中处于领先地位而受到了广泛关注。最先提出的基于孪生实例搜索的目标跟踪（SINT, Siamese instance search for tracking）[5]算法和基于全卷积孪生网络的目标跟踪（SiamFC, fully-convolutional Siamese networks for object tracking）[6]算法使用孪生网络学习目标对象和候选图像块之间的相似性度量，从而将跟踪建模为在整个图像上搜索目标对象的问题，并由此衍生出一系列基于孪生网络的跟踪算法，例如，在 SiamFC 算法的基础上引入区域提议网络（RPN, region proposal network）的基于区域提议网络的目标跟踪（SiamRPN, high performance visual tracking with Siamese region proposal network）[7]算法，它由用于前景−背景估计的分类网络和用于锚点边界框修正的回归网络（即学习与预定义锚点边界框的 2D 坐标偏移量）组成，允许使用可变宽高比的边界框估计目标位置和目标尺寸，从而获取一个更加准确的边界框。随后，基于干扰物感知的孪生网络跟踪（DaSiamRPN, distractor-aware Siamese network for visual object tracking）[8]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, deeper and wider Siamese network for real-time visual tracking）[9]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（CIR, cropping-inside residual）单元，进一步提高了跟踪的准确性和稳健性。基于深度网络的孪生网络跟踪（SiamRPN++, evolution of Siamese visual tracking with very deep network）[10]算法在 SiamRPN 的基础上，使用更深的特征提取网络 ResNet50 代替 AlexNet，并且加入多层融合的策略，使用逐通道互相关操作代替 SiamFC 中简单的互相关操作，从而带来更高的跟踪精度。能够进行目标分割的在线孪生网络跟踪（SiamMask, fast online object tracking and segmentation: a unifying approach）[11]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，大幅提高了跟踪的准确性。

虽然上述基于孪生网络的跟踪算法均取得了当时最优的性能，由于其均只使用离线训练的方法，因此存在一定的局限性。1) 基于孪生网络的跟踪算法忽略了跟踪过程中的背景信息，导致其在面临相似性干扰的情况下判别能力较弱；2) 基于相关滤波器的跟踪算法[11]通过使用手工制作的特征和预先训练得到的用于对象分类的深层特征来学习对象外观的在线模型，相对而言，在基于孪生网络的跟踪算法中使用在线学习机制的思想受到的关

注较少；3) 基于孪生网络的跟踪算法仅使用第一帧作为模板帧，或者仅通过移动加权平均法更新模板帧，导致其在被跟踪目标发生巨大形变、旋转和运动模糊的情况下跟踪性能变差，在进行目标回归时，稳健性较差，容易跟丢目标。另外，基于孪生网络的目标跟踪算法使用互相关性特征图来度量模板帧特征和检测帧局部特征的相似性，从而确定跟踪目标的位置，理想的互相关得分图的尖峰位置即为被跟踪目标的实际位置。通过离线训练学习到好的特征表征进而产生一个好的互相关得分图，使跟踪算法获得更好的跟踪效果，这也是 SiamFC 算法真正有效的原因，而一些基于孪生网络的跟踪算法背离了这个初衷，离线训练学习到一个扭曲的特征图，因此限制了其跟踪性能的提高。

2　算法描述

本文算法以 SiamRPN++算法为基础，引入一种在线更新机制。该在线更新机制包括具有判别性的在线目标分类模块和有效的自适应模板更新模块，提出在线目标分类及自适应模板更新的孪生网络跟踪算法。整体框架如图 1 所示，主要包括特征提取模块、SiamRPN 模块、分类互相关特征图监督模块、在线目标分类模块和自适应模板更新模块。

2.1　特征提取模块和 SiamRPN 模块

本文将 SiamRPN++算法作为基准算法，特征提取模块仍然沿用 SiamRPN++使用的、修改后的 ResNet50 网络，SiamRPN 模块的使用也与 SiamRPN++算法保持一致。基于孪生网络的目标跟踪算法使用互相关操作将目标跟踪问题表述为模板匹配问题，通过学习一个嵌入式空间 $\varphi(\cdot)$（如图 1 中的特征提取模块所示）来计算待搜索区域中能够最佳匹配目标模板的位置，如式(1)所示。

$$f_{\mathrm{M}}(\boldsymbol{x},\boldsymbol{z})=\varphi(\boldsymbol{x})*\varphi(\boldsymbol{z})+b*\boldsymbol{I} \tag{1}$$

其中，分支 $\varphi(\boldsymbol{z})$ 为学习目标模板帧 $\boldsymbol{z}$ 的特征表示，分支 $\varphi(\boldsymbol{x})$ 为学习检测帧 $\boldsymbol{x}$ 的特征表示，并且这 2 个分支 $\varphi(\bullet)$ 的网络参数权重是共享的；b 为表征相似性度量值的偏置量，*为互相关操作，M 表示 mathing 阶段。

在式(1)基础上，SiamRPN++算法使用区域候选网络头（如图 1 中 RPN_head 所示）中的 $h^{\mathrm{cls}}[\cdot]$ 和 $h^{\mathrm{reg}}[\cdot]$ 分别独立地预测目标位置和回归预测边界框，如式(2)所示。

$$\begin{aligned} f_{\mathrm{M}}^{\mathrm{cls}}(\boldsymbol{x},\boldsymbol{z})&=h^{\mathrm{cls}}\left[\varphi_{\mathrm{cls}}(\boldsymbol{x})*\varphi_{\mathrm{cls}}(\boldsymbol{z})\right]\\ f_{\mathrm{M}}^{\mathrm{reg}}(\boldsymbol{x},\boldsymbol{z})&=h^{\mathrm{reg}}\left[\varphi_{\mathrm{reg}}(\boldsymbol{x})*\varphi_{\mathrm{reg}}(\boldsymbol{z})\right] \end{aligned} \tag{2}$$

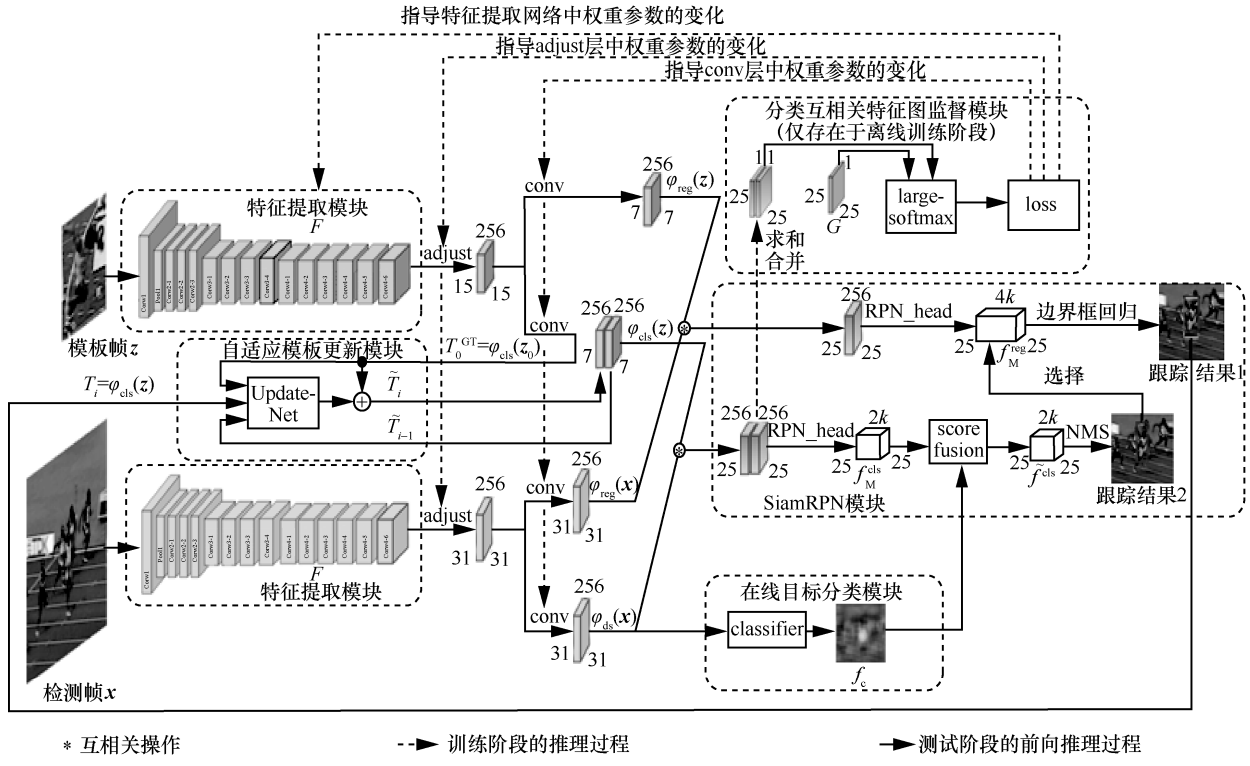


图 1　在线目标分类及自适应模板更新的孪生网络跟踪算法整体框架

其中，$\varphi_{\text{cls}}(\cdot)$ 和 $\varphi_{\text{reg}}(\cdot)$ 等同于式(1)中的 $\varphi(\cdot)$，分别用于学习目标模板帧 $\boldsymbol{z}$ 和检测帧 $\boldsymbol{x}$ 的特征表示；$f_{\text{M}}^{\text{cls}}(\boldsymbol{x},\boldsymbol{z})$ 和 $f_{\text{M}}^{\text{reg}}(\boldsymbol{x},\boldsymbol{z})$ 均为 4 维向量，$f_{\text{M}}^{\text{cls}}(\boldsymbol{x},\boldsymbol{z})$ 存储了各个预定义锚点框的目标/背景得分信息，$f_{\text{M}}^{\text{reg}}(\boldsymbol{x},\boldsymbol{z})$ 存储了相对于预定义锚点框中心点位置的坐标偏移量以及预定义锚点框与真实目标框的宽高比例信息。本文采用和 SiamRPN[7]、SiamRPN++[10]一致的候选边界框筛选策略，得到更加可靠的目标/背景得分信息 $\tilde{f}_{\text{M}}^{\text{cls}}(\boldsymbol{x},\boldsymbol{z})$，根据 $\tilde{f}_{\text{M}}^{\text{cls}}(\boldsymbol{x},\boldsymbol{z})$ 使用非极大值抑制（NMS, non maximum suppression）找到得分最高的预定义锚点框 $\hat{B}_{\text{anchor}}=(\hat{x}^{\text{an}},\hat{y}^{\text{an}},\hat{w}^{\text{an}},\hat{h}^{\text{an}})$（如图 1 的跟踪结果 1 中边界框所示），从 $f_{\text{M}}^{\text{reg}}(\boldsymbol{x},\boldsymbol{z})$ 中选择得到对应锚点框中心点的坐标偏移量 $(\text{d}x^{\text{reg}},\text{d}y^{\text{reg}})$ 以及该锚点框与真实目标框的宽高比例信息 $(\text{d}w^{\text{reg}},\text{d}h^{\text{reg}})$，在此基础上，对得分最高的预定义锚点框进行边界框坐标回归，如式(3)所示，进而得到最终的目标预测边界框 $B_{\text{predict}}=(x^{\text{pre}},y^{\text{pre}},w^{\text{pre}},h^{\text{pre}})$（如图 1 的跟踪结果 2 中边界框所示）。

$$\begin{aligned}
x^{\text{pre}}&=\hat{x}^{\text{an}}+\text{d}x^{\text{reg}}\hat{w}^{\text{an}}\\
y^{\text{pre}}&=\hat{y}^{\text{an}}+\text{d}y^{\text{reg}}\hat{h}^{\text{an}}\\
w^{\text{pre}}&=\hat{w}^{\text{an}}\text{e}^{\text{d}w^{\text{reg}}}\\
h^{\text{pre}}&=\hat{h}^{\text{an}}\text{e}^{\text{d}h^{\text{reg}}}
\end{aligned}\tag{3}$$

2.2 分类互相关特征图监督模块

基于孪生网络的目标跟踪算法使用互相关特征图来度量模板帧特征和检测帧局部特征的相似性，从而确定跟踪目标的位置，理想的互相关特征图的尖峰位置即为被跟踪目标的实际位置，这也是 SiamFC 真正有效的原因。SiamRPN++中的区域提议网络可以看作一个修正网络，因此，如果通过网络可以学习到一个好的互相关特征图，那么经过 RPN 模块修正就会得到一个更好的响应得分图。互相关特征图与 RPN 特征图如图 2 所示。由于 SiamRPN++采用多层融合的策略，对经过 3 个 RPN 模块的输出值附加相应的权重值，从图 2 可以发现，SiamRPN++中互相关特征图（如图 2 中互相关特征图所示）与经过 RPN 模块修正之后的得分图（如图 2 中 RPN 特征图所示）并不是简单的正相关关系，这与 3 个 RPN 模块对应的权重值有关。在训练过程中，这 3 个权重值也需要通过训练学习得到，并且这 3 个权重值的学习变化会使整个跟踪框架中的参数学习问题变得复杂。因此，本文舍弃了多层融合的策略，直接选用来自特征提取模块的单层输出特征，受 ta-SiamRPN++的启发，layer4 的输出值对跟踪效果影响较大[12]，因此本文只选用 layer4。3.3.1 节实验证明，使用单层输出特征取得了比 SiamRPN++使用多层输出特征更好的跟踪效果。为了得到理想的互相关特征图，本文采用对互相关特征图进行监督的策略，从而有利于克服相似干扰。

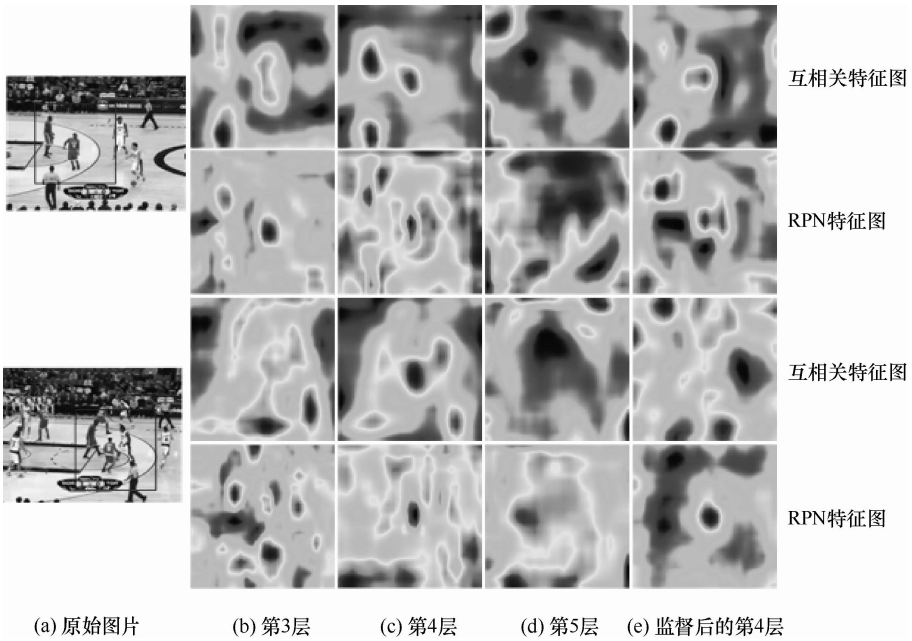


图 2 互相关特征图与 RPN 特征图对比

2.2.1 监督标签的设定

设监督标签与 SiamRPN 模块设定的真实标签保持一致，定义为

$$G(i,j)=\begin{cases}1, \exists \text{IoU} \geqslant 0.6\\ 0, \forall \text{IoU}<0.6\end{cases} \tag{4}$$

$G(i,j)$=1 代表该位置至少存在一个锚点框与真实边界框的交并比（IoU, intersection over union）值大于 0.6，即判别该位置为正；$G(i,j)$=0 代表该位置所有锚点框与真实锚点框的 IoU 值均小于 0.6，即判别该位置为负。IoU=$(\boldsymbol{B}_{\text{anchor}} \cap \boldsymbol{B}_{\text{groundtruth}})/(\boldsymbol{B}_{\text{anchor}} \cup \boldsymbol{B}_{\text{groundtruth}})$，其中，$\boldsymbol{B}_{\text{anchor}}$ 为锚点的边界框，$\boldsymbol{B}_{\text{groundtruth}}$ 为目标的真实边界框。

2.2.2 L-softmax 损失函数

交叉熵损失和归一化指数函数（softmax）是卷积神经网络（CNN, convolutional neural network）中最常用的计算机视觉数学工具之一，而大裕度归一化指数损失函数（L-softmax, large-margin softmax loss）[13]是一种改进的 softmax 分类方法，可以提高类间可分离性和类内紧凑性。此外，L-softmax 不仅可以调整所需的裕度，而且可以避免过度拟合。因此使用 L-softmax 损失函数代替传统的交叉熵损失函数。

定义第 i 个输入特征 x_i 对应标签值 y_i，传统的交叉熵损失为

$$L=\frac{1}{N}\sum_i^N L_i=\frac{1}{N}\sum_i^N -\log\left(\frac{\mathrm{e}^{f_{y_i}}}{\sum_j^J \mathrm{e}^{f_j}}\right) \tag{5}$$

其中，N 为样本集合的大小；f_j 为类别得分 $\boldsymbol{f}$ 中的第 j 个元素值，$j\in\{0,\cdots,J\}$，J 为类别的数量，本文用于二分类判别任务，J=1，$j\in\{0,1\}$；标签值 $y_i\in\{0,1\}$。在目标分类任务中，$\boldsymbol{f}$ 通常为全连接层 $\boldsymbol{W}$ 的输出值，所以 $f_j=\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{x}_i$，其中，$\boldsymbol{W}_j$ 为 $\boldsymbol{W}$ 的第 j 列，f_{y_i} 为第 i 个输入特征 $\boldsymbol{x}_i$ 对应标签值 y_i 处的类别得分值。由于 f_j 为 $\boldsymbol{W}_j$ 和 $\boldsymbol{x}_i$ 的内积，因此 $f_j=\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)$，其中，$\theta_j\in[0,\pi]$ 为 $\boldsymbol{W}_j$ 与 $\boldsymbol{x}_i$ 之间的矢量夹角，由此可得

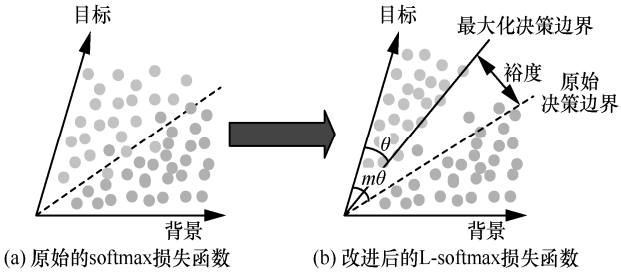
$$L_i=-\log\left(\frac{\mathrm{e}^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i})}}{\sum_j^J \mathrm{e}^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right) \tag{6}$$

跟踪问题实际解决的是跟踪目标的判别问题（目标为正样本，非目标为负样本），因此可以将该问题归结为二分类问题，假设样本 $\boldsymbol{x}_i$ 为正样本，原始的 softmax 函数中需满足 $\boldsymbol{W}_+^{\mathrm{T}}\boldsymbol{x}_i>\boldsymbol{W}_-^{\mathrm{T}}\boldsymbol{x}_i$（$\|\boldsymbol{W}_+\|\|\boldsymbol{x}_i\|\cos(\theta_+)>\|\boldsymbol{W}_-\|\|\boldsymbol{x}_i\|\cos(\theta_-)$，其中 $\boldsymbol{W}_+$ 和 $\boldsymbol{W}_-$ 分别为将样本 $\boldsymbol{x}_i$ 训练为正、负样本学习得到的权重）。

如图 3 所示，L-softmax 为了使正负样本之间存在一个决策裕度，即 $\|\boldsymbol{W}_+\|\|\boldsymbol{x}_i\|\cos(\theta_+)\geqslant\|\boldsymbol{W}_+\|\|\boldsymbol{x}_i\|\cos(m\theta_+)>\|\boldsymbol{W}_-\|\|\boldsymbol{x}_i\|\cos(\theta_-)$，其中，$\theta_+\in[0,\pi/m]$；$m$ 为与决策裕度密切相关的整数。所以必然满足

$$\|\boldsymbol{W}_+\|\|\boldsymbol{x}_i\|\cos(m\theta_+)>\|\boldsymbol{W}_-\|\|\boldsymbol{x}_i\|\cos(\theta_-) \tag{7}$$

式(7)中的分类标准是对样本 $\boldsymbol{x}_i$ 进行正确分类予以更严格的要求，从而为正样本与负样本之间产生更严格的决策边界。



(a) 原始的softmax损失函数　　(b) 改进后的L-softmax损失函数

图 3　2 种决策边界对比说明

引入决策裕度后，L-softmax 损失函数定义为

$$L_i=-\log\left(\frac{\mathrm{e}^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\psi(\theta_{y_i})}}{\mathrm{e}^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\psi(\theta_{y_i})}+\sum_{j\neq y_i}^J \mathrm{e}^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_j)}}\right) \tag{8}$$

其中，$\psi(\theta_{y_i})$ 需满足

$$\psi\left(\theta_{y_i}\right)=\begin{cases}\cos\left(m\theta_{y_i}\right), 0\leqslant\theta_{y_i}\leqslant\dfrac{\pi}{m}\\ D\left(\theta_{y_i}\right), \dfrac{\pi}{m}<\theta_{y_i}\leqslant\pi\end{cases} \tag{9}$$

其中，m 越大，决策裕度越大，目标函数的学习越难；$D\left(\theta_{y_i}\right)$ 应该单调递减并且 $D\left(\dfrac{\pi}{m}\right)=\cos\left(\dfrac{\pi}{m}\right)$。为了简化前向和反向传播的计算，文献[13]将 $\psi(\theta_{y_i})$ 定义为

$$\psi\left(\theta_{y_i}\right)=(-1)^k\cos\left(m\theta_{y_i}\right)-2k \tag{10}$$

其中，$\theta_{y_i}\in\left[k\pi/m,(k+1)\pi/m\right]$，正整数 $k\in[0,m-1]$，$\cos\left(m\theta_{y_i}\right)$ 定义为

$$\cos\left(m\theta_{y_i}\right)=$$
$$\mathrm{C}_m^0\cos^m\left(\theta_{y_i}\right)-\mathrm{C}_m^2\cos^{m-2}\left(\theta_{y_i}\right)\left(1-\cos^2\left(\theta_{y_i}\right)\right)+$$
$$\mathrm{C}_m^4\cos^{m-4}\left(\theta_{y_i}\right)\left(1-\cos^2\left(\theta_{y_i}\right)\right)^2+\cdots+$$
$$(-1)^n\mathrm{C}_m^{2n}\cos^{m-2n}\left(\theta_{y_i}\right)\left(1-\cos^2\left(\theta_{y_i}\right)\right)^n+\cdots \tag{11}$$

其中，$\cos\left(\theta_j\right)=\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{x}_i/\left\|\boldsymbol{W}_j\right\|\left\|\boldsymbol{x}_i\right\|$，正整数 n 满足 $2n\leqslant m$。

可以看出，L-softmax 在原来的基础上附加满足更严格的约束条件式(7)，对输出预测值 f_{y_i} 进行优化。在训练过程中，L-softmax 存在难以收敛的问题[13]，采用一种学习策略使式(12)成立。

$$\hat{f}_{y_i}=\frac{\lambda\left\|\boldsymbol{W}_{y_i}\right\|\left\|\boldsymbol{x}_i\right\|\cos\left(\theta_{y_i}\right)+\left\|\boldsymbol{W}_{y_i}\right\|\left\|\boldsymbol{x}_i\right\|\psi\left(\theta_{y_i}\right)}{1+\lambda} \tag{12}$$

其中，$f_{y_i}=\left\|\boldsymbol{W}_{y_i}\right\|\left\|\boldsymbol{x}_i\right\|\cos(\theta_{y_i})$ 为原始的输出预测值，$\tilde{f}_{y_i}=\left\|\boldsymbol{W}_{y_i}\right\|\left\|\boldsymbol{x}_i\right\|\psi(\theta_{y_i})$ 为附加严格约束条件式(7)下的输出预测值。由于直接采用 $\hat{f}_{y_i}=\tilde{f}_{y_i}=\left\|\boldsymbol{W}_{y_i}\right\|\left\|\boldsymbol{x}_i\right\|\psi(\theta_{y_i})$ 会存在最终损失值难以收敛的情况，因此使用 λ 参数来调节 f_{y_i} 与 $\tilde{f}_{y_i}$ 之间的平衡比重，在进行梯度下降的初始阶段，设置较大的 λ 值，使不附加严格约束条件的 f_{y_i} 主导损失值的优化，在每次的迭代过程中逐步减小 λ 值，逐渐减小 f_{y_i} 所占比重，从而增加 $\tilde{f}_{y_i}$ 所占比重，理想情况下，λ 值可以逐步减小到 0，最终使附加严格约束条件的 $\tilde{f}_{y_i}$ 主导损失值的优化。在实际的应用过程中，λ 减小到很小的值即可。

由于实际的跟踪问题采用与目标检测任务不同的框架，因此需要对 L-softmax 进行如下调整。将检测分支得到的特征图 $\varphi_{\mathrm{cls}}(\boldsymbol{x})$ 作为式(8)中的 $\boldsymbol{x}_i$，将模板分支得到的特征图 $\varphi_{\mathrm{cls}}(\boldsymbol{z})$ 作为式(8)中的 $\boldsymbol{W}_j$，则对于 L-softmax，$\boldsymbol{W}_j$、$\boldsymbol{x}_i$ 都已经是固定的参数，而 $\boldsymbol{W}_j$、$\boldsymbol{x}_i$ 也是特征提取网络 F、调整（adjust）层、卷积（conv）层学习后的结果，所以最终是通过前面各个卷积层的学习，使 $\boldsymbol{W}_j$、$\boldsymbol{x}_i$ 满足一定的目标或者条件，即式(5)最小原则。

从图 2(e)可以发现，对互相关特征图进行监督后，互相关特征图中的目标区域具有较高的响应，并且在此基础上，使用 RPN 模块进行修正，得到了一个更好的响应得分图；滤除了目标周围的相似干扰；在 VOT2018 数据集上取得了比 SiamRPN++ 更好的效果。

2.3 在线目标分类模块

在线目标分类模块主要包括 3 个子模块，如图 4 所示。

1) 压缩子模块，用于减少来自特征提取模块的特征通道数，使用1×1的卷积层加以实现，从而使其更适用于分类任务，也减少了相应的计算量。

2) 注意力子模块，用于解决原始特征在空间位置和各个通道之间的数据失衡问题，以提取特定于当前目标的特征。经过离线训练得到卷积特征 $\varphi_{\mathrm{cls}}(x)$ 并且在实际的跟踪过程中固定卷积层 φ_{cls} 的权重参数，提取得到的卷积特征 $\varphi_{\mathrm{cls}}(x)$ 并不针对某个特定的被跟踪对象，而是提取目标的通用特征。直接使用原始特征，相对于正样本（即目标区域）而言，负样本（即图像中的背景区域）所占比重大于正样本所占比重，导致所有负样本置信度得分的拟合将主导在线学习；另外，只有很少的卷积核在构造每个特征模式或对象类别时发挥重要作用[14]。

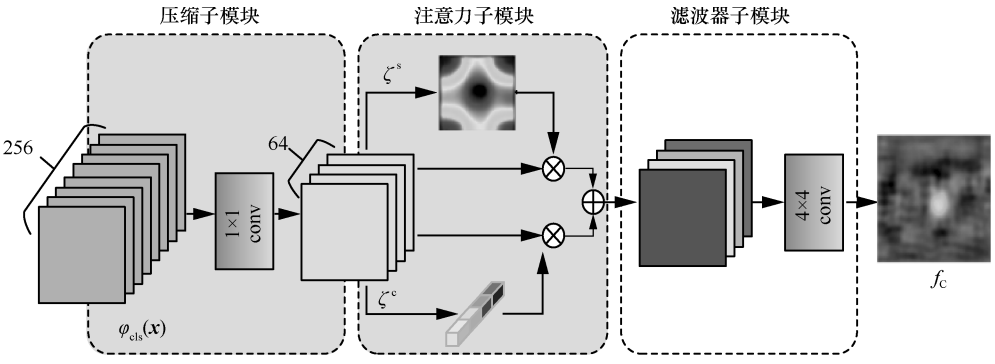


图 4 在线分类模块

原始特征在空间位置和各个通道之间的数据失衡都会降低模型的判别能力，基于以上问题，本文引入双重注意力机制（空间注意力机制和通道注意力机制，如图 4 中 ζ^{s}、ζ^{c} 所示）[15]，ζ^{s} 代表对每个二维空间位置平均池化后，由 softmax 操作形成的二维空间注意力特征图，用于特征图位置权重的获取；ζ^{c} 代表对每个通道进行平均池化之后经由 2 个全连接层（包含激活函数）形成的通道注意力特征图，用于特征图通道权重的获取，最终提取到特定于当前目标的特征，从而将目标和搜索区域中的其他干扰物区分开。

3) 滤波器子模块，用于在线学习实际跟踪过程中的滤波器参数更新，使用卷积核大小为 4×4 的卷积层加以实现，以抑制在线跟踪过程中的背景噪声。Da-SiamRPN[8]中指出，即使提取到能对干扰物感知（特定于当前目标）的特征，基于孪生网络的跟踪算法在跟踪过程中也容易被相似物体干扰。产生这种现象的一个更深层次的原因在于，没有执行在线权重更新来抑制在线跟踪过程中存在的背景噪声。因此，本文引入在线更新的滤波器子模块，以抑制在线跟踪过程中的背景噪声。

图 4 中的压缩子模块和注意力子模块主要用于提取对当前被跟踪目标的特定特征，因此只需要在给定图像序列的第一帧中进行参数更新，在后续的跟踪过程中该参数保持不变以确保跟踪的稳定性。利用提取得到的特定于当前目标的特征来优化后续帧中的滤波器子模块，以抑制跟踪过程中的背景噪声。3.3.2 节实验验证了在线分类模块的有效性。

在线分类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。

$$L_{\mathrm{C}}(\boldsymbol{w}^{\mathrm{C}})=\sum_{j=1}^{M}\gamma_j^{C}r_{\mathrm{C}}(\boldsymbol{f}_{\mathrm{C}}(\boldsymbol{x}_j^{\mathrm{C}};\boldsymbol{w}^{\mathrm{C}}),\boldsymbol{y}_j^{\mathrm{C}})+\sum_{k}^{K}\lambda_k^{\mathrm{C}}\left\|\boldsymbol{w}_k^{\mathrm{C}}\right\|^2 \quad (13)$$

其中，$\boldsymbol{x}_j^{\mathrm{C}}(j=1,2,\cdots,M)$ 为在线分类模块的输入特征，在本文中等同于图 1 中的 $\varphi_{\mathrm{cls}}(x)$，$M$ 为训练样本池的容量大小；$\boldsymbol{f}_{\mathrm{C}}$ 为在线分类模块输出的置信度得分图（如图 4 中的 $\boldsymbol{f}_{\mathrm{C}}$ 所示）；$r_{\mathrm{C}}(\boldsymbol{f}_{\mathrm{C}},\boldsymbol{y}_j^{\mathrm{C}})$ 为在每个空间位置的残差，$\boldsymbol{y}_j^{\mathrm{C}}\in\boldsymbol{R}^{W\times H}$ 为对应位置的真实标签值，W、H 为置信度得分图 $\boldsymbol{f}_{\mathrm{C}}$ 的宽和高；γ_j^{C} 为每个输入样本 x_j^{C} 对应的权重值，用于控制每个训练样本的影响程度；$\boldsymbol{w}_k^{\mathrm{C}}(k=1,2,\cdots,K)$ 为在线分类模块中卷积层权重，K 为卷积层的层数，本文中 $K=4$；λ_k^{C} 为对应 $\boldsymbol{w}_k^{\mathrm{C}}$ 的正则化系数。

式(13)中，$r_{\mathrm{C}}(\boldsymbol{f}_{\mathrm{C}},\boldsymbol{y}_j^{\mathrm{C}})$ 参考判别式跟踪算法 ATOM（accurate tracking by overlap maximization）[16]，在初始化阶段（第一帧），将 r_{C} 设置为 L2 损失，对卷积层提取得到的特征 $\varphi_{\mathrm{cls}}(x)$ 进行监督，从而生成特定于当前目标的特征，即

$$r_{\mathrm{C}}(\boldsymbol{f}_{\mathrm{C}},\boldsymbol{y}_j^{\mathrm{C}})=\left\|\boldsymbol{f}_{\mathrm{C}}-\boldsymbol{y}_j^{\mathrm{C}}\right\|^2 \quad (14)$$

式(14)中，$\boldsymbol{y}_j^{\mathrm{C}}$ 是以跟踪算法实际预测得到的目标位置为中心，得到的具有高斯分布的标签值，可以随着跟踪的进行迭代优化滤波器子模块的参数。

针对式(13)的在线学习优化问题，本文沿用 ATOM[16]中的牛顿−高斯下降法代替传统的随机梯度下降（SGD, stochastic gradient descent）作为优化策略，将式(13)重新定义为残差向量的平方范数形式

$$L_{\mathrm{C}}\left(\boldsymbol{w}^{\mathrm{C}}\right)=\left\|r_{\mathrm{C}}\left(\boldsymbol{w}^{\mathrm{C}}\right)\right\|^2$$

$$r_j^{\mathrm{C}}\left(\boldsymbol{w}^{\mathrm{C}}\right)=\sqrt{\gamma_j}\left(\boldsymbol{f}_C(\boldsymbol{x}_j^{\mathrm{C}};\boldsymbol{w}^{\mathrm{C}})-\boldsymbol{y}_j^{\mathrm{C}}\right)$$

其中，$j\in\{1,\cdots,M\}$，$r_{M+k}^{\mathrm{C}}(\boldsymbol{w}^{\mathrm{C}})=\sqrt{\lambda_k^{\mathrm{C}}}\boldsymbol{w}_k^{\mathrm{C}}$，$k=1,2,3,4$。从而将式(13)的优化问题转变为正定的二次规划问题，使其能够在反向传播期间自适应地更新搜索方向 p 和学习率 $\boldsymbol{\alpha}$。

获得 $\boldsymbol{f}_{\mathrm{C}}$ 后，使用三次插值将其调整到与 SiamRPN 模块中的分类得分 $\boldsymbol{f}_{\mathrm{M}}^{\mathrm{cls}}$ 相同的空间大小，然后，通过加权求和将它们融合在一起，得出在线目标分类得分，可以表示为

$$\hat{\boldsymbol{f}}^{\mathrm{cls}}\left(\boldsymbol{x}^{\mathrm{C}};\boldsymbol{w}^{\mathrm{C}}\right)=\beta_{\mathrm{C}}\boldsymbol{f}_{\mathrm{C}}(\boldsymbol{x}^{\mathrm{C}};\boldsymbol{w}^{\mathrm{C}})+\left(1-\beta_{\mathrm{C}}\right)\boldsymbol{f}_{\mathrm{M}}^{\mathrm{cls}}\left(\boldsymbol{x},\boldsymbol{z}\right) \quad (15)$$

其中，β_{C} 为 2 种分类分数的加权系数值。

2.4　自适应模板更新模块

2.4.1　经典的模板更新策略

一些跟踪方法（如 Da-SiamRPN[8]、SiamMargin[17]）使用一种简单的移动平均策略基于给定的跟踪样本更新目标外观模型，目标模板作为滑动平均值进行更新，权重随着时间的增长呈指数衰减。选择合适的指数权重，可以得出用于更新模板的后续递推式为

$$\tilde{T}_i = (1-\eta)\tilde{T}_{i-1} + \eta T_i \tag{16}$$

其中，i 为第 i 帧图像；T_i 为使用第 i 帧计算得到的新模板帧；$\tilde{T}_i$ 为累积模板；η 为更新率，通常设置为一个固定的较小值（如 η=0.01），假设对象的外观在连续帧中平稳且持续地变化。在基于孪生网络的跟踪算法中，T 是由特征提取网络从特定帧中得到的目标外观模板。尽管原始的 SiamFC 跟踪算法[6]和一系列基于孪生网络的跟踪算法[7,9,11]不执行任何目标模板更新，但较新的孪生网络跟踪器[8,17]已采用式(16)来更新目标模板信息。

虽然模板平均方法为整合新信息提供了一种简单的方法，在大多数跟踪情况下，这种更新机制是不够的，存在以下几个缺点。1) 目标对象可能会因变形、快速运动或遮挡而出现外观变化，从而使更新的条件不同，但它为每个图像序列应用了恒定的更新速率。即使在同一视频中，目标模板上所需的更新也可能在不同时间动态变化。2) 固定的更新策略还导致对象模板更集中于最近的帧，而遗忘了被跟踪目标的历史外观信息。3) 沿目标模板的所有空间维度（包括通道维度）的更新是恒定的。被跟踪目标面临部分遮挡情况下，仅需要更新模板中的一部分，这种更新策略并不有效。4) 跟踪算法无法在目标漂移后重新跟踪目标。部分原因是它无法访问目标的原始外观模板 T_0，而外观模板 T_0 是唯一给定目标信息真实可靠的模板。目标模板更新后的特征仅限于先前帧目标外观模板和当前帧目标外观模板的简单线性组合，其严重限制了更新机制的灵活性，这在目标进行复杂外观变化时很重要，因此考虑更复杂的组合功能有望改善跟踪结果。

2.4.2 自适应模板更新策略

为了解决上述移动平均策略出现的问题，本文通过学习通用的函数 ϕ 来更新目标模板。

$$\tilde{T}_i = \phi\left(T_0^{\mathrm{GT}}, \tilde{T}_{i-1}, T_i\right) + T_0^{\mathrm{GT}} \tag{17}$$

其中，T_0^{GT} 为初始第一帧给定的目标模板信息，$\tilde{T}_{i-1}$ 为上一帧累积得到的目标模板信息，T_i 为根据当前帧预测得到的目标位置提取的目标模板信息。该函数 ϕ 使用卷积神经网络加以实现，具有强大的特征表达能力和从大量数据中学习的能力，该神经网络称为 UpdateNet[18]。

图 5 展示了在基于孪生网络的跟踪算法上使用 UpdateNet 来自适应更新目标模板信息的整体框架。本文使用图 1 中的 φ_{cls} 提取得到目标区域的深层特征信息。首先，根据第一帧给定的目标真实边界框信息提取得到第一帧目标模板特征 T_0^{GT}。为了获得当前帧的模板特征 T_i，使用之前所有帧的累积模板特征 $\tilde{T}_{i-1}$（$\tilde{T}_{i-1}$ 为上一帧中 UpdateNet 的输出值）来预测第 i 帧中目标位置（如图 5 中虚线箭头所示），并且提取得到目标区域的特征信息 T_i（如图 5 中最下部实线箭头所示）。将第一帧目标模板特征 T_0^{GT}、当前帧的模板特征 T_i、上一帧的累积模板特征 $\tilde{T}_{i-1}$ 级联并送入 UpdateNet。对于第一帧，将 $\tilde{T}_{i-1}$ 和 T_i 均设置为 T_0^{GT}。UpdateNet 唯一使用的真实信息是第一帧给定的目标边界框信息，其他所有 UpdateNet 的输入全部基于跟踪算法预测得到的目标边界框信息。可以说，T_0^{GT} 是指导更新 UpdateNet 最可靠的信息来源，因此，采用残差学习策略，通过从 T_0^{GT} 向 UpdateNet 的输出添加跳连接的方式使 UpdateNet 学习如何修正真实目标模板特征 T_0^{GT}，并将其应用于当前帧的跟踪。具体的 UpdateNet 的训练细节可以参考文献[18]和 3.2.2 节中关于 UpdateNet 的具体参数设置。

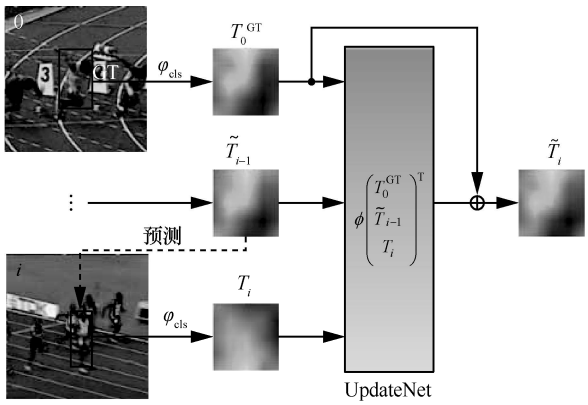


图 5 UpdateNet 的整体框架

UpdateNet 通过整合当前帧给出的信息来更新上一帧累积得到的目标模板 $\tilde{T}_{i-1}$。因此，基于当前帧目标模板和累积目标模板之间的差异，UpdateNet 能够自适应当前帧的特定更新需求。此外，UpdateNet 还考虑了初始目标模板 T_0^{GT}，从而提高了抵抗目标模板漂移的稳健性。

3 实验

本节采用 VOT2018[19]、VOT2019[17]作为实验数据集。VOT2018 包含 60 个具有精细人工标注的

目标跟踪图像序列，含有摄像机运动、光照变化、运动变化、尺寸变化、遮挡 5 种跟踪难点。VOT2019 是通过替换 VOT2018 中跟踪难度较小的 20%目标跟踪图像生成得到的，跟踪难度更高。

3.1　实验平台

本节实验均在 GPU 为 Nvidia GTX 1080ti 的台式机上进行，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。

3.2　实验参数设置

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取更大的性能增益。因此对不同的数据集应设置不同的算法参数，为了提高算法应用适应性，本节给出了具体的超参数搜索算法。

3.2.1　超参数设置

针对数据集 VOT2018 和 VOT2019，文献[10]中对 penalty_k（记为 κ）、window_influence（记为 α_{wi}）、scale_lr（记为 α_{LR}）设置了 4 组不同的超参数。

文献[10]采用网格搜索的超参数搜索方法。本文在超参数搜索的过程中发现，这种方法会增加搜索算法的时间复杂度（其时间复杂度为 $O\left(n^3\right)$），因而采取一种更加简单的超参数搜索方式，即控制变量法，具体过程如下：固定 3 个参数中的 2 个，确定剩余的一个参数的搜索区间，在相应的数据集上进行评估，找到性能表现最好的一个参数，再依次按照同样的方法寻优另外 2 个参数，最终确定在当前数据集上表现最好的一组参数。这种寻优方式的时间复杂度为 $O(n)$，相比网格搜索法，其时间复杂度大大降低，并且取得了和网格搜索法一致的最优参数组合结果。本文对于数据集 VOT2018、VOT2019 设置寻优区间如下：κ 为 $\left[0.01,0.62\right)$，α_{wi} 为 $\left[0.01,0.62\right)$，α_{LR} 为 $\left[0.01,0.62\right)$，寻优步长为 0.01。最终找到的效果最好的参数设置如下：对于 VOT2018 数据集，设置 κ=0.05，α_{wi}=0.38，α_{LR}=0.44；对于 VOT2019 数据集，设置 κ=0.44，α_{wi}=0.26，α_{LR}=0.44。

3.2.2　其他参数设置

互相关特征图监督模块的参数设置如下：m 为 4；λ 的初始值为 100，衰减系数为 0.99，即 $\lambda_{N+1}=0.99\times\lambda_N$，其中 N 为迭代次数。

在线目标分类模块中的参数设置如下：优化训练样本池的大小 $M=250$，训练样本池容量达到 250 后，用最新得到的帧替换最旧的历史帧进而添加到训练样本池中，其中训练样本权重 γ_j 的更新学习率为 0.01，当在邻近目标周围检测到干扰后学习率为 0.02；在线目标分类模块中的滤波器子模块每 10 帧更新一次；为了有效地融合分类得分，令 $\beta_{\mathrm{C}}=0.8$ [15]。

自适应模板更新模块的参数设置主要涉及 UpdateNet 离线训练的参数设置。参考文献[18]使用 LaSOT 标准数据集[20]中的 20 个训练视频图像序列，采用三阶段训练的方式，在第一阶段，在同一视频图像序列中，根据真实坐标边界框裁剪得到 T_0^{GT} 和 T_{i+1}^{GT}，运行本文提出的跟踪算法（加入分类互相关特征图监督模块和在线目标分类模块，不加 UpdateNet，下同）得到每一帧的坐标边界框，并根据其裁剪得到 T_i，使 $\tilde{T}_i$=T_i；在第二和第三阶段，使用上一阶段训练好的 UpdateNet 权重，在同一视频图像序列中，运行本文跟踪算法得到每一帧的预测边界框，并根据其裁剪得到 T_i，将 T_0^{GT}、$\tilde{T}_{i-1}$、T_i 一起送入 UpdateNet 中，得到 $\tilde{T}_i$，其中 $\tilde{T}_0$=T_0^{GT}。UpdateNet 由两层卷积神经网络组成，包括一个 1×1×1536×192 的卷积层，经过 ReLU 激活以及一个 1×1×192×512 的卷积层；在第一阶段，权重使用 Xavier 初始化，在每个 epoch，学习率从 10^{-6} 呈对数下降到 10^{-7}；此后，使用上一阶段训练得到的最好模型参数进行初始化。在训练过程中发现，在训练的第二和第三阶段，学习率会不同程度地影响实际的训练效果，因而尝试使用不同学习率的对数衰减区间，依次在 VOT2018 数据集上进行测试，进而寻找到最优的学习率衰减区间。在第二阶段，区间依次设定为[10^{-5}, 10^{-6}]、[10^{-6}, 10^{-7}]、[10^{-7}, 10^{-8}]、[10^{-8}, 10^{-9}]、[10^{-9}, 10^{-10}]、[10^{-10}, 10^{-11}]，通过测试发现，学习率的对数衰减区间设置为[10^{-9}, 10^{-10}]效果最好。在第三阶段，区间依次设定为[10^{-7}, 10^{-8}]、[10^{-8}, 10^{-9}]、[10^{-9}, 10^{-10}]、[10^{-10}, 10^{-11}]、[10^{-11}, 10^{-12}]、[10^{-12}, 10^{-13}]，通过测试发现，学习率的对数衰减区间设置为[10^{-11}, 10^{-12}]效果最好；每个训练阶段使用批次大小为 64 的样本训练 50 个 epoch 的模型，使用动量为 0.9、权重衰减为 0.000 5 的随机梯度下降法进行训练。其他参数与文献[10]中的参数设置相同。

3.3　对比实验

本节在 VOT2018 标准数据集上进行对比实验，

评估互相关特征图监督模块、在线目标分类模块、自适应模板更新模块的作用。采用期望重叠率（EAO, expected average overlap）、准确性 A、稳健性 R、跟丢次数（LN, lost number）、跟踪速度 V_{FPS} 这 5 个评价指标对改进的算法进行评估。

3.3.1 使用互相关特征图监督模块

将 SiamRPN++作为基准算法，在此基础上，只使用特征提取网络中 layer4 的输出特征，并且加入分类监督模块（CS module, classification supervision module）对互相关特征图进行监督。只使用单层特征，在 VOT2018 数据集上取得了比 SiamRPN++更好的跟踪结果，结果如表 1 所示。

表 1 在 VOT2018 数据集上实验结果对比

方案	A	R	EAO	LN/次	V_{FPS} /(帧·s^{-1})
SiamRPN++	0.601	0.234	0.415	50	35
+ CS module	0.583	0.197	0.432	42	62

从表 1 可以发现，通过对互相关特征图进行监督后在 VOT2018 数据集上带来 4.1%的 EAO 提升。主要原因是算法跟踪稳健性提升，与 SiamRPN++算法相比，本文算法跟丢次数减少了 8 次。值得注意的是，这里仅仅使用了特征提取网络中的 layer4 的特征，却取得了比 SiamRPN++中 layer3、layer4、layer5 三层特征融合策略更好的结果，并且算法跟踪速度明显提升。

3.3.2 使用在线目标分类模块

在 SiamRPN++算法的基础上，加入在线分类模块（OC module, online classification module），包括通道压缩子模块、注意力子模块、在线滤波器子模块，实验结果如表 2 所示。在 VOT2018 数据集上，EAO 提升到 0.417，大幅减少了跟丢次数，从 SiamRPN++的 50 次减少到 32 次；跟踪精度也明显提高，从 0.601 提升到 0.611，提高了 1%。在 3.3.1 节实验的基础上，加入在线分类模块在 VOT2018 数据集上 EAO 提升到 0.463，比基准算法 SiamRPN++提升了 11.8%；跟丢次数也进一步减少，从 50 次减少到 30 次，取得了和分类监督模块近似的效果。

进一步设置对比实验，使用 VOT2018 数据集作为测试集，验证在线分类模块中各子模块（压缩子模块、注意力子模块、滤波器子模块）的重要性。在 3.3.1 节的最佳设置下，依次去除压缩子模块、注意力子模块、滤波器子模块，观察对应子模块的重要性，结果如表 3 所示。从表 3 可以看出，去除子模块后评价指标 EAO 明显下降，跟丢次数增多，说明 3 个子模块均对提高算法稳健性、提升 EAO 有所贡献。其中，滤波器子模块的贡献最大，在去除滤波器子模块后，EAO 从 0.463 下降到 0.406，性能下降最大，证明了本文算法中在线更新滤波器子模块的重要性，其能有效降低跟踪过程中的噪声干扰，从而提高算法稳健性，并提高跟踪算法整体性能。

表 2 在 VOT2018 数据集上实验结果对比

方案	A	R	EAO	LN/次	V_{FPS} /(帧·s^{-1})
SiamRPN++	0.601	0.234	0.415	50	35
+OC module	0.611	0.150	0.417	32	28
+ OC module + CS module	0.591	0.140	0.463	30	41

表 3 在 VOT2018 数据集上实验结果对比

方案	online update			A	R	LN/次	EAO
	cmp	attn	flt				
SiamRPN++	×	×	×	0.601	0.234	50	0.415
+ CS module +OC module	✓	✓	✓	0.591	0.140	30	0.463
+CS module +OC module	×	✓	✓	0.589	0.150	32	0.437
+CS module +OC module	✓	×	✓	0.589	0.150	32	0.433
+CS module +OC module	✓	✓	×	0.593	0.220	47	0.406

3.3.3 使用自适应模板更新模块

在 SiamRPN++算法上加入分类监督模块、在线目标分类模块的基础上，进一步加入自适应模板更新模块（TU module, adaptive template update module），在 VOT2018 数据集上进行对比实验，结果如表 4 所示。从表 4 可以看出，EAO 提升到 0.471，跟丢次数减少到 26 次，算法稳健性进一步提升，取得了更好的跟踪效果。

VOT2019 与 VOT2018 得到的结论相同，因此不详细论述。

3.4 实验结果与分析

3.4.1 VOT2018 实验

尽管 SiamRPN++算法体现了深度神经网络强大的特征表征能力，但当前某些基于孪生网络的跟踪算法仍然会在面临相似物干扰、完全遮挡

和严重形变（如 VOT2018 中的 hands、liquor、gymnastics3）时表现不佳。本文方法由于引入了在线更新机制，因而在处理上述问题时表现更加稳健，获得了比 SiamRPN++算法更好的跟踪结果。

表 4　在 VOT2018 数据集上实验结果对比

pipeline	*A*	*R*	EAO	LN/次	V_{FPS} /(帧·s^{-1})
SiamRPN++ +CS module	0.601	0.234	0.415	50	35
OC module +TU module	0.588	0.122	0.471	26	34

VOT2018 标准数据集包含许多具有挑战性的因素，因此可以被视为在准确性和稳健性方面较全面的测试平台。为了保证实验结果的客观性，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跟踪算法 SiamBAN[21]、DiMP50[22]、SiamFC++[23]、SiamRCNN[24]、ATOM[14]、SiamMargin[17]、ta-SiamRPN++[12]、SiamMask[11]、SiamDW[9]、SiamRPN++[10]、DaSiamRPN[8]、SiamRPN[7]、UpdateNet[18]、SiamFC[6]，采用期望重叠率、准确性、稳健性、跟丢次数、跟踪速度这 5 个评价指标对 15 种性能优异的跟踪算法进行了性能比较，如表 5 所示。

表 5　在 VOT2018 数据集上实验结果对比

Tracker name	*A*	*R*	EAO	LN/次	V_{FPS} / (帧·s^{-1})
SiamFC++	0.587	0.150	0.467	32	90
SiamBAN	0.590	0.178	0.447	38	40
DiMP50	0.597	0.153	0.440	33	43
SiamMargin	0.583	0.197	0.432	42	62
SiamRPN++	0.601	0.234	0.415	50	35
SiamR-CNN	0. 612	0.22	0.405	47	15
ATOM	0.590	0.204	0.401	44	30
UpdateNet	0.587	0.276	0.393	59	92
DaSiamRPN	0.586	0.276	0.383	59	160
SiamMask	0.609	0.281	0.381	60	35
Ta-SiamRPN++	0.593	0.272	0.360	58	36
SiamDW	0.538	0.398	0.270	85	150
SiamFC	0.503	0.585	0.187	125	101
本文算法	0.588	0.122	0.471	26	34

从表 5 可以看出，相对于该数据集上进行评测的最新跟踪算法，本文提出的跟踪算法具有良好的性能，以较高的准确性（0.588）和良好的稳健性（0.122），获得了最高的 EAO（0.471），保证了算法的稳健性，这主要是因为本文算法中引入了在线更新机制。与基准算法 SiamRPN++算法相比，本文算法虽然在准确性上不如 SiamRPN++算法，但算法跟丢次数从 SiamRPN++算法的 50 次大幅度减少至 26 次，使跟踪稳健性大幅提高，最终 EAO 比 SiamRPN++算法提升了 13.5%。同时，本文在保证良好跟踪准确性的前提下，延续了基于孪生网络类跟踪算法的高效率，运行速度为 34 帧/秒。

3.4.2　VOT2019 实验

同样，本文算法在 VOT2019 标准数据集上进行测试与评估。与 VOT2018 相比，VOT2019 跟踪难度更高。对于 VOT2019 标准数据集的 60 组跟踪图像序列，本节引入 VOT2019 的实时组中表现较好的跟踪算法 SiamMargin、DiMP、SiamBAN、SiamDW_ST[17]、SiamMask、SiamRPN++、ATOM，采用期望重叠率、准确率、稳健性、跟丢次数这 4 个评价指标对 12 种性能优异的跟踪算法进行了性能比较，如表 6 所示。

表 6　在 VOT2019 数据集上实验结果对比

Tracker name	*A*	*R*	EAO	LN/次	V_{FPS} / (帧·s^{-1})
SiamMargin	0.579	0.321	0.366	65	46
SiamBAN	0.602	0.396	0.327	79	40
DiMP	0.582	0.371	0.321	74	40
SiamDW_ST	0.600	0.467	0.299	93	—
SiamMask	0.594	0.461	0.287	92	35
SiamRPN++	0.599	0.482	0.285	96	35
ATOM	0.596	0.557	0.240	111	—
本文算法	0.579	0.296	0.337	59	33

从表 6 可以看出，SiamMargin[17]通过使用对互相关特征图监督的策略和移动平均的模板更新策略实现了较少的跟丢次数。本文算法跟丢次数最少（为 59 次），因此本文算法具备较好的稳健性；准确性与 SiamMargin 相同，与基准算法 SiamRPN++相比有所下降；EAO 从 0.285 提升至 0.337，性能提升了 18.2%。

4 结束语

目前，基于孪生网络的目标跟踪算法只使用离线训练好的网络进行目标的辨识和定位，在处理相似干扰、目标形变时缺乏足够的判别力，往往跟踪的稳健性较差，容易跟丢目标。为解决该问题，本文引入互相关特征图监督模块、在线目标分类模块、自适应模板更新模块。在互相关特征图监督模块中，通过在离线训练阶段使用 L-softmax 损失函数对互相关特征图附加更严格的约束条件，从而学习到更易区分目标和背景的互相关特征图，使网络学习到的特征更具判别力，有利于克服相似干扰。在线目标分类模块中，压缩子模块用于压缩通道信息、减少计算量；双重注意力（空间注意力和通道注意力）子模块用于提取特定于当前被跟踪目标的特征；在线更新滤波器执行判别式学习，辅助修正离线训练网络提取的特征，从而增强了孪生网络处理干扰物的判别能力。在自适应模板更新模块中，使用 UpdateNet 整合第一帧目标模板、累积目标模板和当前帧目标模板的信息，自适应地更新可靠的目标模板信息，以应对目标发生严重形变的问题，并且具备抵抗目标模板漂移的稳健性。在满足实时性速度要求的前提下，利用标准数据集 VOT2018 和 VOT2019 进行测试，相比基准算法 SiamRPN++，本文算法分别带来 13.5%和 18.2%的性能（EAO）提升，证明了本文算法的有效性。

参考文献：

[1] BOLME D S, BEVERIDGE J R, DRAPER B A, et al. Visual object tracking using adaptive correlation filters[C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2010: 2544-2550.

[2] 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.

[3] DANELLJAN M, HÄGER G, KHAN F S, et al. Learning spatially regularized correlation filters for visual tracking[C]//2015 IEEE International Conference on Computer Vision. Piscataway: IEEE Press, 2015: 4310-4318.

[4] 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. Piscataway: IEEE Press, 2017: 6931-6939.

[5] TAO R, GAVVES E, SMEULDERS A W M. Siamese instance search for tracking[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2016: 1420-1429.

[6] BERTINETTO L, VALMADRE J, HENRIQUES J F, et al. Fully-convolutional Siamese networks for object tracking[M]. Berlin: Springer, 2016.

[7] 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. Piscataway: IEEE Press, 2018: 8971-8980.

[8] ZHU Z, WANG Q, LI B, et al. Distractor-aware Siamese networks for visual object tracking[M]. Berlin: Springer, 2018.

[9] 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. Piscataway: IEEE Press, 2019: 4586-4595.

[10] 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. Piscataway: IEEE Press, 2019: 4277-4286.

[11] 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. Piscataway: IEEE Press, 2019: 1328-1338.

[12] 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.
CHEN Z W, ZHANG Z X, SONG J, et al. Tracking algorithm for Siamese network based on target-aware feature selection[J]. Acta Optica Sinica, 2020, 40(9): 0915003

[13] LIU W Y, WEN Y D, YU Z D, et al. Large-margin softmax loss for convolutional neural networks[J]. arXiv Preprint, arXiv: 1612.02295, 2016.

[14] LI X, MA C, WU B Y, et al. Target-aware deep tracking[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2019: 1369-1378.

[15] ZHOU J H, WANG P, SUN H Y. Discriminative and robust online learning for Siamese visual tracking[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 13017-13024.

[16] DANELLJAN M, BHAT G, KHAN F S, et al. ATOM: accurate tracking by overlap maximization[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2019: 4655-4664.

[17] MATEJ K, JIRI M, ALES L, et al. The seventh visual object tracking VOT2019 challenge results[C]//2019 IEEE/CVF International Conference on Computer Vision Workshop. Piscataway: IEEE Press, 2019: 2206-2241.

[18] ZHANG L C, GONZALEZ-GARCIA A, WEIJER J V D, et al. Learning the model update for Siamese trackers[C]//2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE Press, 2019: 4009-4018.

[19] 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. Berlin: Springer, 2018: 3-53.

[20] FAN H, LIN L T, YANG F, et al. LaSOT: a high-quality benchmark for large-scale single object tracking[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2019: 5369-5378.

[21] CHEN Z D, ZHONG B N, LI G R, et al. Siamese box adaptive network for visual tracking[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2020: 6667-6676.

[22] BHAT G, DANELLJAN M, VAN GOOL L, et al. Learning discriminative model prediction for tracking[C]//2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE Press, 2019: 6181-6190.

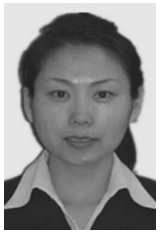
[23] XU Y D, WANG Z Y, LI Z X, et al. SiamFC++: towards robust and accurate visual tracking with target estimation guidelines[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12549-12556.

[24] VOIGTLAENDER P, LUITEN J, TORR P H S, et al. Siam R-CNN: visual tracking by Re-detection[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE Press, 2020: 6577-6587.

[作者简介]

陈志旺（1978– ），男，河北武清人，博士，燕山大学副教授、硕士生导师，主要研究方向为多旋翼飞行控制、目标跟踪等。

张忠新（1996– ），男，山东聊城人，燕山大学硕士生，主要研究方向为目标跟踪。



宋娟（1978– ），女，黑龙江佳木斯人，国网黑龙江省电力有限公司高级工程师，主要研究方向为发电厂智能控制。

雷海鹏（1997– ），男，河北张家口人，燕山大学硕士生，主要研究方向为目标跟踪。

彭勇（1963– ），男，河北唐山人，博士，燕山大学教授、博士生导师，主要研究方向为特种机器人与人工智能。