

基于改进 SiamFC 的实时目标跟踪算法

张红颖*, 贺鹏艺, 王汇三

中国民航大学电子信息与自动化学院, 天津 300300

摘要 针对全卷积孪生神经网络 SiamFC 在目标跟踪速度以及网络判别能力上有待提升的问题, 提出了一种基于改进 SiamFC 的实时目标跟踪算法。将原网络结构中的第二层卷积层替换为深度可分离卷积, 通过减少参数计算量, 提高了跟踪速度; 为了提高网络判别能力, 第三层卷积层使用混合深度卷积, 通过不同尺寸的卷积核提取特征, 实现多特征融合, 提取到鲁棒性更强的特征; 采用预处理后的 ILSVRC2015 数据集, 使用随机梯度下降法对网络进行训练, 并在 OTB2015、VOT2016、ILSVRC2015 数据集上对算法性能进行测试。实验结果表明, 该算法和 SiamFC 算法相比, 在跟踪成功率、跟踪精度以及跟踪速度上都有一定的提升, 并能够满足实时跟踪要求。

关键词 机器视觉; 目标跟踪; 孪生网络; 深度可分离卷积; 混合深度卷积; 多特征融合

中图分类号 TP391

文献标志码 A

doi: 10.3788/LOP202158.0615003

A Real-Time Target-Tracking Algorithm Based on Improved SiamFC

Zhang Hongying*, He Pengyi, Wang Huisan

College of Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Abstract A real-time target tracking algorithm based on improved SiamFC (a fully convolutional Siamese network) is proposed herein to improve the target tracking speed and the network discrimination ability of traditional SiamFC. First, after the first conventional convolution layer, a depth-wise separable convolution is set to improve the tracking speed by reducing the amount of parameter calculation. Second, the third convolution layer is set as a mixed depth-wise convolution (MixConv) to improve the recognition ability of the network. We extracted features from the convolution kernel of different sizes and concatenated them in the channel to achieve a multifeature fusion and extract more robust features. Finally, the preprocessed ILSVRC2015 data set was used to train the network using the random gradient descent method, and the performance of the algorithm was tested on the OTB2015, VOT2016, and ILSVRC2015 data sets. Experimental results show that compared with the SiamFC algorithm, our algorithm shows a certain improvement in the tracking success rate, tracking accuracy, and tracking speed, and can meet real-time tracking requirements.

Key words machine vision; object tracking; Siamese network; depthwise separable convolutions; mixed depthwise convolutions; multi-feature fusion

OCIS codes 150.0155; 150.1135

1 引言

目标跟踪是计算机视觉领域里的一个研究热点, 在诸如公共场所安防监控、交通监督、军事应用、医疗器械等方面均有广泛的应用。目标跟踪的应用

面临着环境复杂、长时间跟踪等一系列挑战^[1]。传统目标跟踪算法由于存在特征提取困难、计算量过大、实时性差等缺点, 跟踪性能难以保证^[2-4]。随着大数据时代的到来, 深度学习技术为目标跟踪领域带来新的特征提取手段, 推动了目标跟踪的发展。

收稿日期: 2020-08-11; 修回日期: 2020-08-17; 录用日期: 2020-08-27

基金项目: 国家重点研发计划(2018YFB1601200)、中央高校基本科研业务费项目中国民航大学专项(3122018C004)

* E-mail: carole_zhang0716@163.com

近年来,随着深度学习的蓬勃发展,各种神经网络框架不断地得到创新和发展,自 Alexnet 问世以来,通过卷积神经网络与传统的目标跟踪方法相结合,大大提高了目标跟踪的性能^[5]。Wang 等^[6]提出的 DLT(deep learning tracker)算法首次将深度学习应用于目标跟踪领域,通过采用离线预训练和在线微调的思路,有效解决了训练样本不足的情况,但由于提取特征的能力不足,其跟踪效果仍低于一些传统的跟踪算法。Bertinetto 等^[7]提出的 SiamFC 跟踪算法使用了孪生神经网络结构,网络的两个分支使用相同的特征提取网络共享权重,并使用相似度学习解决跟踪问题。该算法的跟踪速度达到实时性要求,故有望应用于实际场景;但在物体出现遮挡、环境复杂的情况下,该算法因网络无法准确地判别目标,导致其跟踪速度慢于一些相关滤波类跟踪算法。为了提升孪生网络的判别能力,Wang 等^[8]提出的 RASNet 算法使用三个注意力机制对 SiamFC 特征的空间和通道进行加权,对特征提取和判别性分析的耦合进行分解,但跟踪速度的提升效果不佳。Li 等^[9]提出的 SiamRPN 算法在 SiamFC 的基础上加入区域候选网络(RPN)用于产生候选区域,卷积网络层数变得更深,提取到的特征信息变得更丰富、更鲁棒,省掉多尺度测试,大大提高了跟踪速度,但模型的数量和计算量也随之增多,并因为数据集大小而受到限制。综合各算法的优势与不足,在不过多增加网络层数的基础上提高孪生网络的判别能力和跟踪的实时性是有待改进的方向。

本文基于 SiamFC 算法提出一种基于孪生网络的实时目标跟踪算法。该算法使用深度可分离卷积,通过减少算法的参数计算量,提高了算法的速度;并通过设置拥有不同尺寸卷积核的混合深度卷积来实现多特征融合,获取表达性更强的特征,提高网络判别能力。

2 SiamFC 算法介绍

SiamFC 跟踪算法使用孪生神经网络结构在较大的搜索区域上进行样本图片搜索。其网络结构主要由 5 个卷积层构成的特征提取网络和用于互相关操作的卷积层组成,模板分支和待搜索分支分别经过相同的特征提取网络。SiamFC 的网络结构是一个全卷积网络,不包含连接层,优点是待搜索图像无需与样本图像尺寸相同。网络结构^[7]如图 1 所示。

基于孪生网络的 SiamFC 算法将跟踪目标过程

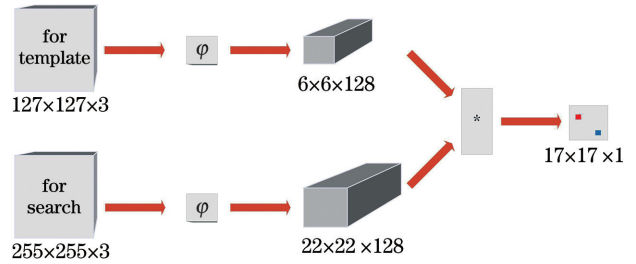


图 1 SiamFC 网络结构

Fig. 1 Structure of SiamFC

视为相似度学习,通过判断第一帧模板和后续帧待搜索图像的相似性确定目标的位置,将 6×6 的特征图看作卷积核,在 22×22 的特征图上进行卷积操作,得到响应图,计算相似度。相似度函数为

$$f(x, y) = \varphi(x) * \varphi(y) + b_1, \quad (1)$$

式中: x 代表模板图像; y 代表待搜索图像; φ 表示提取网络; b_1 代表在响应图上每个位置的取值。

SiamFC 在跟踪阶段的目标模型不更新,模板分支始终是第一帧图像,后续帧图像都在采样以后和第一帧图像进行相似度计算,便于目标被遮挡时能够迅速找回,但当图像中出现多个同类型目标交替遮挡时,该算法可能会跟丢目标对象。同时,由于在线不更新,网络不存在模板累积误差,但图像上下文之间的联系未能得到充分利用,无法随时适应目标和背景信息的变化,因此提高网络的判别能力、设置合适的跟踪策略是必要的改进手段。

SiamFC 使用卷积层替代传统 Alexnet 的全连接层,减少了参数,有效地提高了实时性,并拥有较好的模型泛化能力。但 SiamFC 网络结构仍采用常规卷积,并且包含较大尺寸的卷积核,网络参数量规模仍然较大,其跟踪速度不如一些使用深度网络的相关滤波类跟踪算法,因此需从卷积结构上进行改进,以减少参数。

3 本文算法

3.1 网络结构

为减少网络在卷积操作时的参数量,提高速度,使用深度可分离卷积替代原网络的第二层常规卷积。深度可分离卷积可以理解为将常规卷积分解为一个 depthwise (DW) 卷积和一个 1×1 的逐点 pointwise (PW) 卷积^[10]。深度可分离卷积的两参数计算量与常规卷积的参数计算量比值为

$$d = \frac{X_{DW} + X_{PW}}{X_{Conv}} = \frac{h \times w \times c \times n \times n + h \times w \times c \times k}{h \times w \times c \times k \times n \times n}, \quad (2)$$

式中: X_{DW} 、 X_{PW} 、 X_{Conv} 表示各类型卷积的参数计算量; h 、 w 、 c 分别代表输入的高度、宽度和通道数; $n \times n$ 代表卷积核的尺寸; k 为卷积核的数目即输出通道数。原网络中第二层常规卷积的卷积核尺寸为 5×5 , 卷积核数目为 256, 将原网络替换为深度可分离卷积后参数量约减少为原来的 $1/25$ 。

针对如何提高网络判别能力, 在原网络的第三层卷积层处, 对通道进行分组并使用混合深度卷积提取特征。通过混合深度卷积的不同尺寸卷积核提取的特征是不同的, 对每组通道使用不同尺寸的卷积核进行 DW 卷积相当于进行了特征融合^[11]。将混合深度卷积通道数等分为 2 组, 设计原则为第 1 组使用的卷积核尺寸为 3×3 , 第 2 组卷积核尺寸为 5×5 , 为了充分利用不同通道在相同空间位置上的特征信息, 在混合深度卷积后添加 PW 卷积。常规卷积和混合深度卷积的参数计算量比值为

$$d_1 = \frac{X_{Mix}}{X_{Conv}} = \frac{\frac{h \times w \times (c/2) \times (n_1 \times n_1 + k/2)}{h \times w \times c \times k \times n \times n} + \frac{h \times w \times (c/2) \times (n_2 \times n_2 + k/2)}{h \times w \times c \times k \times n \times n}}{h \times w \times c \times k \times n \times n}, \quad (3)$$

式中: X_{Mix} 表示混合深度卷积的参数计算量; $n_1 \times n_1$ 、 $n_2 \times n_2$ 代表混合深度卷积两组卷积核的尺寸。原网络中第三层常规卷积的卷积核尺寸为 3×3 , 卷积核数目为 384, 替换为 3×3 、 5×5 混合深度卷积后参数量约减少为原来的 $3/50$ 。

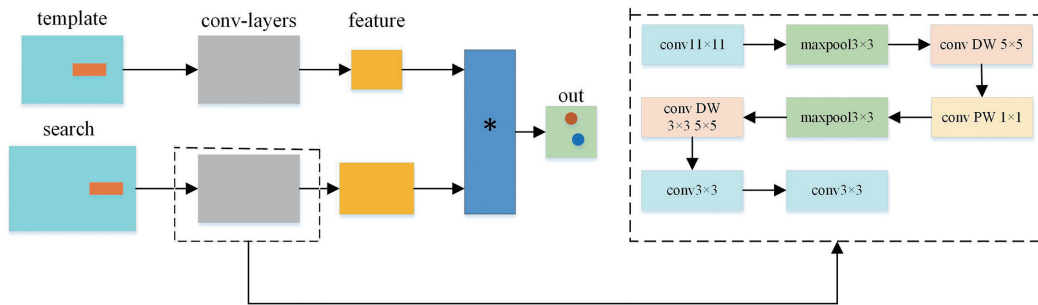


图 3 网络结构图

Fig. 3 Network structure

3.2 数据预处理

使用 ILSVRC2015 VID^[12] 数据集来训练网络。ILSVRC2015 数据集大小为 86 GByte, 其中图像有 30 个基本类别。将数据集下载并解压后, 包含 3800 个训练集视频序列, 555 个验证集视频序列, 937 个测试集视频序列。因样本分支和待搜索分支

深度可分离卷积和使用不同尺寸的卷积核的混合深度卷积结构如图 2 所示。图 2 中椭圆区域代表进行卷积操作, 箭头代表通道。图 2 表示深度可分离卷积先对输入张量的每个通道进行 $K \times K$ 的 DW 卷积, 而后对得到的所有特征图进行 1×1 的 PW 卷积, 最终得到输出张量。混合深度卷积是先对通道进行分组, 分别使用 3×3 、 5×5 、 $K \times K$ 的 DW 卷积核对各组进行卷积, 而后对所有特征图在通道上合并。

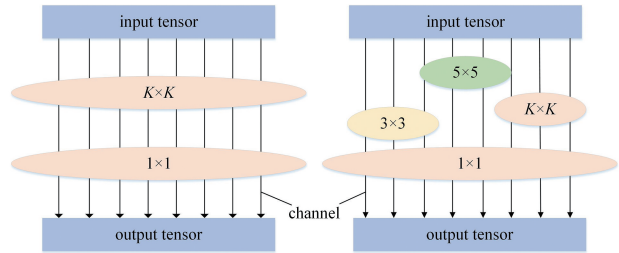


图 2 深度可分离卷积和混合深度卷积结构

Fig. 2 Structures of depthwise separable convolution and mixed depthwise convolution

本文改进后的卷积网络结构如图 3 所示。其中, “conv”代表常规卷积, “conv DW”代表 depthwise 卷积, “conv PW”代表 pointwise 卷积, “maxpool”代表池化层。卷积层默认使用 ReLU 函数作为激活函数, 第一层常规卷积和深度可分离卷积层后接最大池化层。深度可分离卷积的填充方式选择为“SAME”, 深度乘数设置为 1, 首先经由 5×5 的 DW 卷积在二维空间内提取特征图, 再通过 PW 卷积在通道上对特征图进行加权组合, 得到新的特征图。

的输入图片尺寸大小需分别为 127 和 255, 故先对原始图片进行预处理, 即通过数据集中给出的标注信息获得物体的位置, 在模板图像中截取一个尺寸为 z 的图像框。尺寸 z 的确定方式为

$$z = (k + 2p) \times (g + 2p), \quad (4)$$

其中 k 、 g 对应数据集中给出标注信息中的尺寸, p

为 k 和 g 之和的一半。这一步的目的是将上下文信息包含进来。随后选择一个比例常数 s , 对图片进行缩放变换, 使得图像尺寸变为 127, 即

$$s(k+2p) \times s(g+2p) = 127. \quad (5)$$

对于待搜索图像, 则直接在整张图片中以第一帧的目标边界框裁剪出 255×255 大小的图片。在对模板图像和待搜索图像进行预处理时出现图像框超出图片范围的情况下, 需先对原图像进行填充操作后再进行缩放和裁剪。

由于 ILSVRC2015 VID 数据集中数据太多, 包含一百多万张图像, 因此采取 GPU 多线程 (multi-threaded) 加速的方式进行图像预处理, 处理后以质量 90 储存图像, 并将数据保存为二进制序列化格式文件用于下一步训练。

3.3 训练过程

为节省时间, 本研究使用 SiamFC 的三尺度网络模型作为预训练网络, 使用预训练网络中的参数作为训练参数的初始值, 训练方法采用随机梯度下降法。训练的 batch size 设为 8, 数据集通过网络训练的次数设为 50 次, 每次训练的样本数设为 53200, 则总训练步数为

$$s_t = \frac{e \times n}{N_{bs}}, \quad (6)$$

式中: s_t 为总训练步数, 设定为 332500 步; e 代表训练总次数; n 代表每次训练的样本数; N_{bs} 代表 batch size 的大小。确定完总训练步数后, 设定每训练十步输出一次训练损失。

在训练中首先对正负样本进行分类, 从目标中心出发, 以半径 r 画圆, 落在圆区域内的为正样本, 落在圆区域外的为负样本。正样本取值为 1, 负样本取值为 0, 如

$$\begin{cases} y = 1, & d < r \\ y = 0, & d > r \end{cases} \quad (7)$$

定义训练中对每一组训练样本使用的损失函数为

$$l = \sum \omega \log [1 + \exp(-yf)], \quad (8)$$

其中 l 代表每组训练样本的损失, ω 为样本权重, y 为样本取值, f 为输出响应图的置信结果。

定义完训练策略和损失函数后进行训练, 由于训练样本数目过大, 设定每训练 6500 步保存一次, 后续不断更新。训练完成后, 使用 tensorboard 将训练的效果进行可视化, 判断训练效果的好坏, 进而调试参数。经调试, 确定训练网络的权值衰减值为 0.0005, 动量项设定为 0.9, 学习率设定为 0.0001,

同时使用随机梯度下降算法 (SGD) 对每个训练样本进行参数更新。

3.4 跟踪

SiamFC 算法的跟踪速度部分得益于其跟踪策略, 该算法在跟踪过程中不更新模板和网络权值, 因此在背景复杂、目标形变较大等条件下其目标跟踪效果会受到限制。这一问题可以通过改用在线跟踪策略或增强对首帧图像的特征提取来解决, 但在线跟踪策略会牺牲部分速度, 降低实时性, 因此本研究通过改进网络结构来增强特征的表达力。为了更好地与 SiamFC 比较, 本文算法采取相同的跟踪策略, 即在跟踪阶段不更新, 始终采用第一帧图像作为模板进行跟踪。本文跟踪算法通过上述步骤, 在跟踪阶段使用训练好的网络进行特征提取, 通过互相关操作得到目标的响应图, 其中响应值最大处即为目标所在位置。

4 实验验证与结果分析

本文算法使用的计算机硬件配置为 Intel Core i7-9750H CPU@2.60GHz, GPU 版本为 NVIDIA GeForce GTX 1660Ti。计算机软件配置为 tensorflow-gpu1.9.0, python3.6.5, matlab2016a, CUDA9.2。

为了评估本文算法的性能, 使用 OTB2015^[13] 数据集进行测试对比, 选择 SiamFC 原文中使用的一次性评估 (OPE)、时间鲁棒性评估 (SRE)、空间鲁棒性评估 (TRE) 作为评价指标, 查看跟踪效果并与 SiamFC 算法进行比较。再选取 VOT2016^[14] 数据集测试本文算法的预期平均重叠率 (EAO)、速度 (FPS 和 EFO)、鲁棒性 (Robustness) 以及准确度 (Accuracy), 并与几种主流算法进行对比。为验证本文算法在实际应用中的表现, 使用 ILSVRC2015 中的部分火车测试集进行测试, 定义评价指标: 跟踪成功率 Overlap 为估计值与真值的交并比, 跟踪精度 Accuracy 为跟踪中心误差小于阈值的比例, 跟踪速度 Speed 为视频帧数与跟踪时长之比。

4.1 OTB 数据集测试

在 OTB2015 数据集上测试本文算法的跟踪性能, 设置本文算法对应的 OTB 数据集跟踪器, 调用训练好的网络, 测试本文算法与 SiamFC^[7] 算法以及其他 4 种近年来表现出色的算法^[15-18] 的性能。结果如图 4 和图 5 所示。通过 OTB 数据集测试结果可知, 本文算法成功率第二, 精确度第三。本文算法相对于 SiamFC 算法, 在 OPE 上成功率和精度分别

提高了 0.018、0.034,在 TRE 上成功率和精度分别提高了 0.029、0.045,在 SRE 上成功率和精度分别提高了 0.021、0.053。由此可以得知本文对卷积结构的改进具有一定的改善效果。图 6 中展示了本文算法在 OTB 部分数据集上的跟踪效果,其中分别截取了视频序列第 1 帧、第 50 帧、第 100 帧和第

150 帧以及对应的响应图。图 7 中对比了 6 种算法在目标快速实时移动时的跟踪效果。通过对图 6 和图 7 的分析,可知当目标快速移动时,SiamFC 和本文算法均能够持续且较准确地跟踪目标,优于其他 4 种算法。并且通过对比发现,本文算法相较于 SiamFC 表现出更好的跟踪效果。

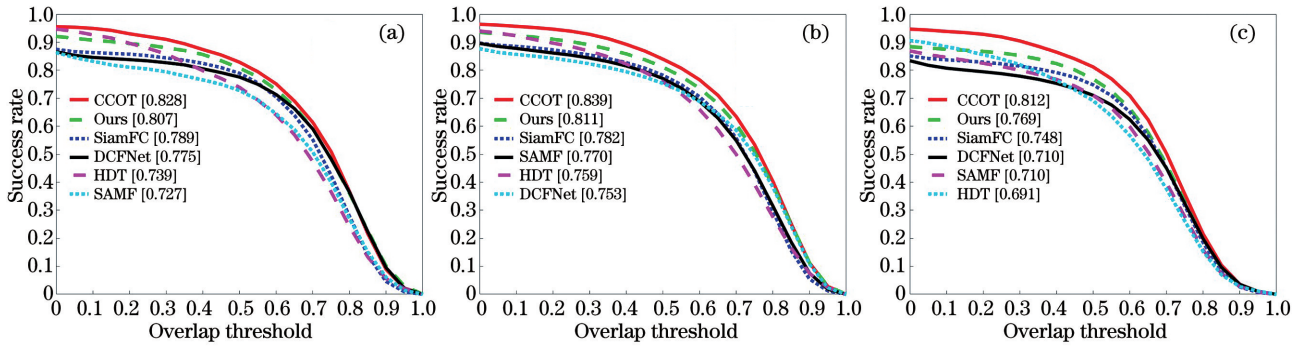


图 4 6 种跟踪算法的 OPE、TRE、SRE 成功图。(a) OPE;(b) TRE;(c) SRE

Fig. 4 Success plots of OPE, TRE, SRE on six tracking algorithms. (a) OPE; (b) TRE; (c) SRE

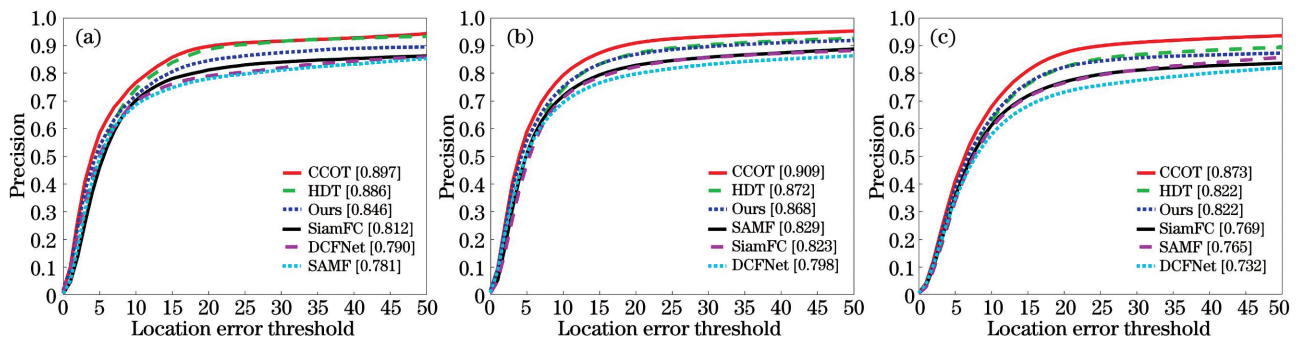


图 5 6 种跟踪算法的 OPE、TRE、SRE 精度图。(a) OPE;(b) TRE;(c) SRE

Fig. 5 Precision plots of OPE, TRE, SRE on six tracking algorithms. (a) OPE; (b) TRE; (c) SRE

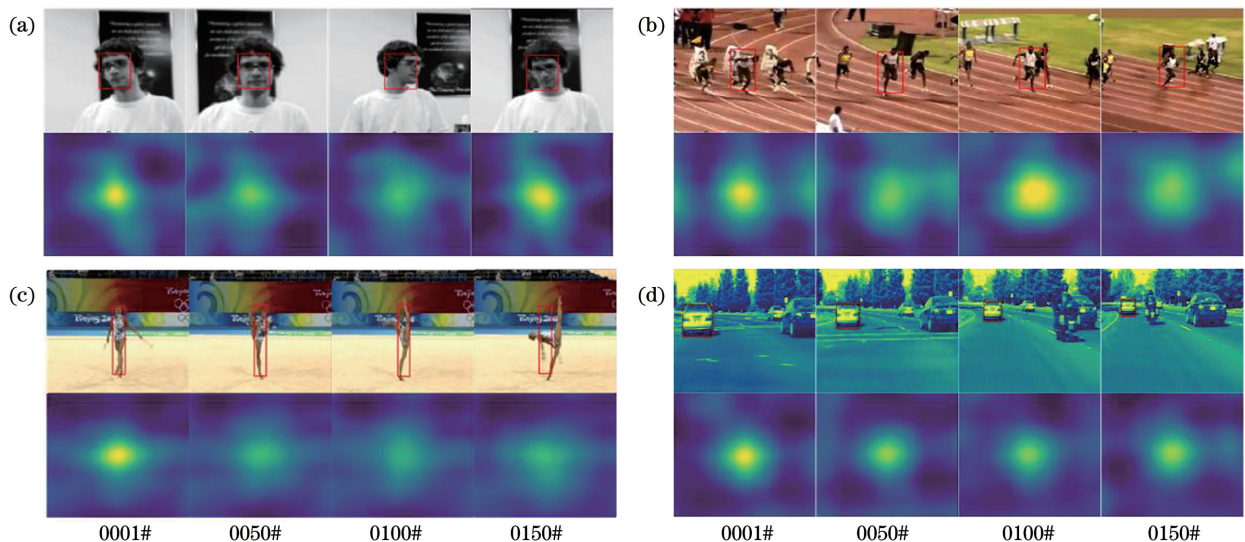


图 6 本文算法在 OTB2015 的部分跟踪结果。(a) David;(b) Bolt;(c) Gym;(d) Car

Fig. 6 Tracking results of our algorithm on OTB2015. (a) David; (b) Bolt; (c) Gym; (d) Car

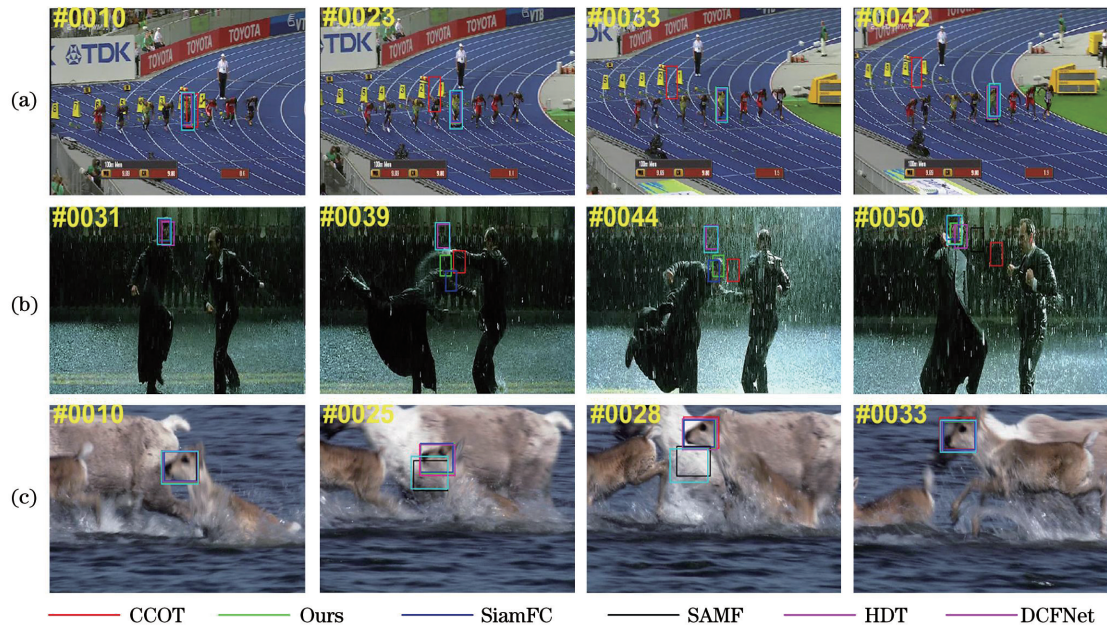


图 7 本文算法与其他 5 种算法在目标快速移动视频序列中的跟踪结果对比。(a) Video 1;(b) video 2;(c) video 3
Fig. 7 Tracking results of our algorithm compared with the other five algorithms in the fast moving video sequences.

(a) Video 1; (b) video 2; (c) video 3

4.2 VOT2016 数据集测试

本研究以官方 VOT2016 数据集上的测试数据为基准,同样使用 VOT2016 数据集对本文算法和 KCF^[19]、SRDCF^[20]、SiamFC^[7]、Staple^[21] 进行对比,调用数据集进行跟踪,得到平均重叠率、跟踪精度以及跟踪速度。该数据集评价指标中 EAO、FPS、EFO 和准确度数值越高,鲁棒性数值越小,由此说明算法性能越优。测试结果如表 1 所示,本文算法虽然在速度指标上无法超越 KCF 和 Staple,但相对于 SiamFC 算法有一定的提高,能够满足实时性要求。同时本文算法在 EAO 上比 SiamFC 高约 0.013,在准确度上比 SiamFC 高约 0.019;本文算法的 EAO 仅次于 Staple,准确度在 5 种算法中最高。在鲁棒性上本文算法相对于 SiamFC 也得到了提升。

表 1 VOT2016 数据集下的评估结果

Table 1 Evaluation results based on VOT2016

Algorithm	EAO	Accuracy	EFO	Robustness	FPS
KCF	0.197	0.512	16.21	0.562	54
SRDCF	0.217	0.524	1.87	0.415	3
SiamFC	0.235	0.532	9.21	0.461	26
Staple	0.294	0.544	10.86	0.384	43
Ours	0.248	0.551	9.47	0.452	32

4.3 ILSVRC2015 火车测试集测试

在 ILSVRC2015 测试集中以序号类别为基准

逐个搜索出火车类别的视频序列,将图像格式转换为 .jpg 格式,并将 .xml 格式的标注数据转换成 .txt 格式的标注数据,将筛选出来的视频序列使用 SiamFC 算法和本文算法进行跟踪比较。在选出的火车类别视频序列中包含火车旋转、火车长时间跟踪、火车快速通过以及火车尺度大小等实际问题^[22]。本研究针对每一个小类别图像只选择其中一个视频序列,共选择了 13 段火车类视频序列进行测试。火车类别视频序列测试后的性能记录如表 2 所示。

为了更加直观地比较本文算法与 SiamFC 算法的性能,以视频序列号为横坐标,评价指标为纵坐标,绘制折线图,如图 8 所示,其中实线代表 SiamFC 的性能,虚线代表本文算法的性能。由图可知,本文算法的性能较 SiamFC 有一定的提高,但在 Overlap 上的提升效果不佳。通过对不同视频序列进行分析,可以看出本文算法在火车快速通过时的性能提升较为明显,但对于火车旋转、尺度大小变化等不敏感。

4.4 网络参数量分析

为了求证改进后的模型参数量如理论分析所示大幅减少,首先根据(2)式和(3)式以及网络各层结构推导出预期的参数量 Params1,再使用 tensorflow 的参数量统计模块统计出实际的参数量 Params2^[23],并与 SiamFC 网络参数量进行对比,如表 3 所示。由表 3 可知,改进后模型的预期参数量

表 2 ILSVRC2015 数据集评估结果

Table 2 Evaluation results based on ILSVRC2015 dataset

Vedio sequence	Overlap / %		Accuracy / %		Speed / (frame · s ⁻¹)	
	SiamFC	Ours	SiamFC	Ours	SiamFC	Ours
00009000	32.2	44.7	55.7	63.2	32.6	35.4
00021000	45.0	47.2	51.8	53.4	44.1	46.2
00138001	65.9	71.3	73.1	84.5	38.9	41.7
00138013	53.4	53.8	68.1	73.0	41.2	43.2
00139004	29.2	44.8	62.0	81.9	10.4	10.7
00323000	58.3	59.2	67.8	68.1	50.7	55.3
00579000	72.1	73.4	58.2	59.2	39.1	39.7
00580000	49.6	54.1	77.6	89.7	48.7	51.4
00631000	52.1	52.3	79.4	80.6	44.2	49.8
00657000	44.5	49.6	57.1	62.0	30.3	35.5
00895000	82.3	85.0	90.7	96.8	48.5	49.1
00938000	78.1	78.5	92.1	92.7	22.7	24.3
00965000	44.7	54.9	66.3	69.4	45.9	52.0

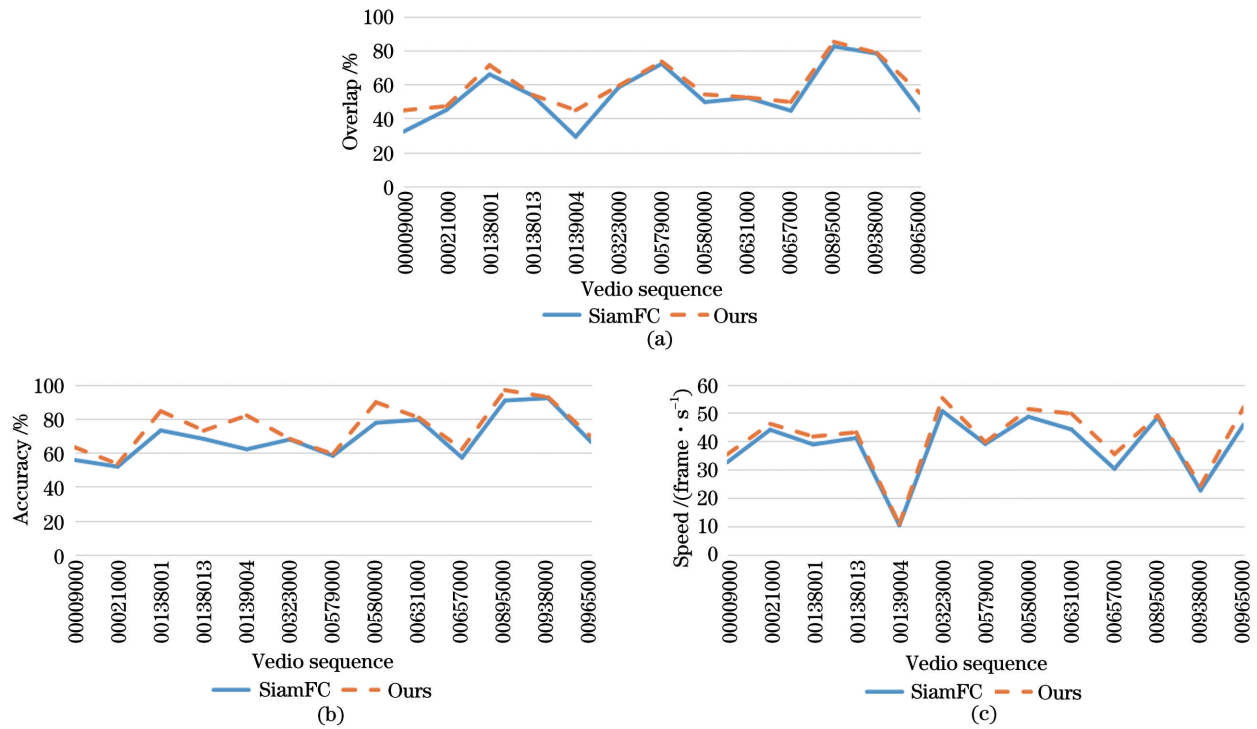


图 8 SiamFC 和本算法在 ILSVRC2015 火车测试集的性能折线。(a)平均重叠率;(b)跟踪精度;(c)跟踪速度

Fig. 8 Performance polylines of SiamFC and our algorithm on the ILSVRC2015-train test set. (a) Overlap; (b) accuracy; (c) speed

表 3 网络参数量

Table 3 Network parameters unit: MByte

Module name	SiamFC	Params1	Params2
Module size	8.31	7.18	7.52

为 7.18 MByte,实际参数量为 7.52 MByte,相对于原 SiamFC 的网络模型,参数量已在一定程度上减少,但并未达到理论效果。通过分析发现,深度可分离卷积和混合深度卷积虽然可以大大减少参数,但同时增加了卷积层层数,故可能会因此带来额外的

参数计算量。

5 结 论

本文算法基于 SiamFC 算法的特征提取网络 Alexnet 结构改进卷积层结构,引入深度可分离卷积和混合深度卷积,提高了跟踪速度,实现了在卷积层上的特征融合,并在 OTB2015、VOT2016、ILSVRC2015 数据集上对 SiamFC 算法和本文算法进行了测试和比较。在 OTB2015 和 VOT2016 公开数据集上的实验结果证明本文算法在跟踪成功率、跟踪精度以及跟踪速度上都有一定的提升;在 ILSVRC2015 的火车类别测试集上的实验结果说明本文算法相较于 SiamFC 在目标旋转、快速移动以及长时间移动的实际问题上表现更好且跟踪速度能够满足实时性要求。但是,本文算法仍然存在一定局限性,对卷积层结构的改进并未达到预期的提升效果,并且仍未解决视频序列中出现目标交替时跟丢目标的问题,期望在下一步工作中得以改善。

参 考 文 献

- [1] Liu M J, Cao Y Z, Zhu S Y, et al. Feature fusion video target tracking method based on convolutional neural network [J]. *Laser & Optoelectronics Progress*, 2020, 57(4): 041502.
刘美菊, 曹永战, 朱树云, 等. 基于卷积神经网络的特征融合视频目标跟踪方法[J]. *激光与光电子学进展*, 2020, 57(4): 041502.
- [2] Zhan R H, Wan J W. Iterated unscented Kalman filter for passive target tracking [J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2007, 43(3): 1155-1163.
- [3] 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.
- [4] Nejhum S M S, Ho J, Yang M H. Online visual tracking with histograms and articulating blocks[J]. *Computer Vision and Image Understanding*, 2010, 114(8): 901-914.
- [5] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks [J]. *Communications of the ACM*, 2017, 60(6): 84-90.
- [6] Wang N Y, Yeung D Y. Learning a deep compact image representation for visual tracking [C] // *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, December 5-10, 2013, Lake Tahoe, Nevada, USA. New York: Neural Information Processing Systems Foundation, Inc., 2013: 809-817.
- [7] Bertinetto L, Valmadre J, Henriques J F, et al. Fully-convolutional Siamese networks for object tracking [C] // Hua G, Jégou H. *European Conference on Computer Vision (ECCV 2016)*. Cham: Springer, 2016, 9914: 850-865.
- [8] 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, USA. New York: IEEE Press, 2018: 4854-4863.
- [9] 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, USA. New York: IEEE Press, 2018: 8971-8980.
- [10] Chollet F. Xception: deep learning with depthwise separable convolutions[C]//*2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 1800-1807.
- [11] Tan M, Le Q V. Mixconv: mixed depthwise convolutional kernels[EB/OL]. (2019-12-01) [2020-08-01]. <https://arxiv.org/abs/1907.09595?context=cs.LG.09595>, 2019.
- [12] Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge [J]. *International Journal of Computer Vision*, 2015, 115(3): 211-252.
- [13] Wu Y, Lim J, Yang M H. Object tracking benchmark [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(9): 1834-1848.
- [14] Kristan M, Matas J, Leonardis A, et al. The visual object tracking VOT2015 challenge results[C]//*2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 564-586.
- [15] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking [C] // *European Conference on Computer Vision*. Cham: Springer, 2016: 472-488.
- [16] Wang Q, Gao J, Xing J L, et al. DCFNet:

- discriminant correlation filters network for visual tracking [EB/OL]. (2017-04-13) [2020-08-01]. <https://arxiv.org/abs/1704.04057>.
- [17] Li Y, Zhu J. A scale adaptive kernel correlation filter tracker with feature integration [C] // European Conference on Computer Vision. Cham: Springer, 2014: 254-265.
- [18] Qi Y K, Zhang S P, Qin L, et al. Hedged deep tracking [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 4303-4311.
- [19] 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.
- [20] 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 (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 4310-4318.
- [21] Bertinetto L, Valmadre J, Golodetz S, et al. Staple: complementary learners for real-time tracking [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 1401-1409.
- [22] Zhao C M, Chen Z B, Zhang J L. Research on target tracking based on convolutional networks [J]. Opto-Electronic Engineering, 2020, 47(1): 0180668. 赵春梅, 陈忠碧, 张建林. 基于卷积网络的目标跟踪应用研究 [J]. 光电工程, 2020, 47(1): 0180668.
- [23] Tan G H, Hou J, Han Y P, et al. Low-parameter real-time image segmentation algorithm based on convolutional neural network [J]. Laser & Optoelectronics Progress, 2019, 56(9): 091003. 谭光鸿, 侯进, 韩雁鹏, 等. 基于卷积神经网络的低参数量实时图像分割算法 [J]. 激光与光电子学进展, 2019, 56(9): 091003.