中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

基于孪生网络的实时视觉目标跟踪研究

作者姓名：王强

指导教师：胡卫明 研究员 中国科学院自动化研究所

学位类别：

学科专业：模式识别与智能系统

培养单位：中国科学院自动化研究所

2020 年 6 月

Learning Siamese Networks for Real-time Visual Object Tracking

A dissertation submitted to the
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Engineering
in Pattern Recognition and Intelligent Systems
By

Institute of Automation, Chinese Academy of Sciences

June, 2020

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：王强

日　　期：2020.6.2

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：王强　　导师签名：

日　　期：2020.6.2　　日　　期：2020.6.2

摘 要

视觉目标跟踪是计算机视觉领域的一项基本研究课题，被广泛应用于智能监控、车辆导航、人机交互和虚拟现实等领域。通用物体的视觉目标跟踪被用来建立任意类型目标物体在视频帧之间的关联关系，进而确定目标运动轨迹，实现对于物体的运动感知，它是智能感知中的重要组成部分。针对通用物体的目标跟踪模型需要适应物体的光照、形变、遮挡等复杂变化，这对跟踪算法具有极大挑战。同时，以无人机、自动驾驶为典型代表的无人自主智能平台对视觉目标跟踪算法实时性的要求较高，因此做到精度与速度的兼顾与平衡也是视觉跟踪算法得以在现实场景中应用的关键之一。本文围绕视觉目标跟踪算法面向现实场景应用的上述需求，以高效的孪生网络目标跟踪架构为研究对象，提出了多种有效的特征表示学习方法，实现了端到端的跨层级特征融合表示以及类脑注意力机制建模。同时我们将图像分割思想引入到目标跟踪对象的状态表述中，拓展了目标跟踪的表述形式，并首次构建了视觉目标跟踪与视频目标分割的一体化处理框架，建立了目标跟踪新范式。本文的主要贡献概括如下：

- **提出了基于端到端学习的判别相关滤波器高效跟踪算法。**本文通过对判别相关滤波操作的反向传播过程进行推导，创新性地实现了深度特征自动提取与相关滤波判别模型的联合优化。该方法有效提升了深度特征表示的学习能力，增加了特征表示网络设计的自由度，同时显著降低了算法的计算存储消耗。在端到端学习过程中，通过基于尺度-位移空间的联合学习，算法引入了尺度空间样本，进而可以提供更准确的目标尺度估计。在此基础上，本文又探索了基于深度特征的语义嵌入模型，并提出使用编解码自监督学习孪生网络实现对目标及其周围环境结构信息的有效感知，提升了特征表示的泛化性能与细粒度表示能力。最后，本文通过分别构建具有上下文感知能力的判别相关滤波器和自监督学习语义嵌入模型，实现了具有互补性的跨层级特征融合表示与学习，显著提升了算法的跟踪性能。

- **提出了基于残差注意力机制的孪生网络高效目标跟踪算法。**本文首先重新形式化了判别式目标跟踪算法框架，将整体跟踪网络解耦为目标特征的表示网络以及用于判别分析的判别网络。然后，本文提出了带有加权的交叉相关操作

算子，可以对目标不同空间位置的相关操作赋以自适应调整的权重。通过联合学习判别相关损失以及目标区域的判别系数，算法实现了较强的表观形态适应性。本文通过注意力机制实现判别权重表述，并提出将注意力机制分解为用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行权重调整的通道注意力机制。该算法通过多种注意力机制的引入，减少了训练过程的过拟合。同时本文算法通过轻量化的网络设计，保证了良好的跟踪速度。

- **提出了基于孪生网络的视觉目标跟踪与分割一体化高效处理框架。**本文深入分析了当前视觉目标跟踪的输出表述形式，借鉴图像分割表述思想，首次创新性地提出针对于目标跟踪的多任务输出表示方法。本文通过引入独立的分割分支到全卷积孪生网络框架，使得孪生网络架构可以同时估计目标的矩形框位置以及输出精细的目标分割表述。对于分割分支的架构设计，本文采用向量化的分割表述方式获取目标全局信息，并提出自顶向下的堆叠精细化模块来增强分割细节。该框架的离线训练过程可通过多分支任务联合学习进行优化。在线跟踪过程中，算法只需要输入初始帧标注的矩形框初始化，即可同时完成视觉目标跟踪任务与视频目标分割任务。整个框架在完成多个任务的基础上，具有较高的分割效率，运行速度接近 55 帧每秒。最后，本文将上述一体化处理框架扩展到多目标跟踪场景，实现了无输入标签监督的多目标视频实例分割。

基于上述方法和创新，本文所提出的跟踪算法在多个公开数据集与挑战赛上都取得了当时最好或者领先的精度指标。同时本文对于跟踪算法的计算效率进行重点关注，本文算法均取得了实时的运算速度。最后，本文的方法和创新对于其它相关计算机视觉任务和应用，比如行为理解等，也有一定的借鉴意义。

关键词：视觉目标跟踪，孪生网络，相关滤波，端到端学习，注意力机制，实例分割

Abstract

Visual object tracking is a basic research problem in computer vision, and is widely used in intelligent monitoring, vehicle navigation, human-computer interaction, virtual reality, and other fields. The generic object tracking is used to establish the association between objects in video frames, and then determine the target trajectory for realizing the motion perception of the object. It is an important part of intelligent perception. The main challenging issues for successful tracking lie in various appearance changes caused by drastic illumination changes, non-rigid deformation, and heavy occlusion, *etc.* At the same time, the unmanned autonomous intelligent platform typified by drones and autonomous driving has high requirements for the real-time performance of visual object tracking. Therefore, a good compromise between accuracy and speed is also one of the keys for the application of visual tracking methods in real-world scenes. Around the visual object tracking algorithm and its application, this thesis proposes multiple effective feature learning methods for siamese network-based tracking architecture. Besides, we introduce the idea of image segmentation to expand the expression form of target tracking with the dense description of the tracking result, and for the first time build an integrated framework for visual object tracking and video object segmentation, leading to a new paradigm for visual tracking. The main contributions of this thesis are summarized as follows.

- **We propose an end-to-end learnable correlation filter tracking algorithm.** Through the derivation of the back-propagation process of discriminative correlation operations, this thesis unifies the feature representation learning and correlation filter-based appearance modeling within an end-to-end learnable framework. It effectively improves the learning ability of deep feature representation, increases the freedom of feature network design, and significantly reduces both the computational cost and memory demand of the algorithm. In the end-to-end learning process, through joint scale-position learning, the algorithm introduces scale samples, which can provide more accurate target scale estimation. Then this thesis explores the semantic embedding model

based on deep features, and proposes an encoder-decoder network for structure-aware self-supervised learning, which improves the generalization performance and fine-grained expression ability of the model. Finally, two complementary cross-layer features are used to jointly learn the context-aware correlation filters and semantic embedding, which significantly increases the tracking accuracy.

- **We propose a residual attentional siamese network for high performance visual object tracking.** This thesis first reformulates the discriminative visual tracking framework, and decouples the overall tracking network into a feature representation network and a discriminant network for discriminant analysis. Then, a weighted cross-correlation operator is proposed to perform adaptive weight adjustment on different spatial positions of the target. The model achieves strong adaptability of apparent morphology through jointly learning to discriminate related losses and discriminant coefficients of target areas. This thesis implements the discriminant weight expression through the attention mechanism, and proposes to decompose the attention mechanism into a priori attention mechanism for statistical overall sample distribution, a residual attention mechanism with individual adaptability, and a channel attention for adjusting the weights of different semantic layers of the network. Through the introduction of attention mechanism, the algorithm not only mitigates the overfitting problem in deep network training, but also enhances its discriminative capacity thanks to the separation of representation learning and discriminator learning. Besides, benefiting from the lightweight network design, the speed of the proposed tracker is far beyond real-time.

- **We propose a unified framework for visual object tracking and video object segmentation in siamese networks.** This thesis deeply analyzes the current output expression form of visual object tracking, and propose an accurate output description method for visual tracking. By introducing an independent segmentation branch into the full convolutional siamese network framework, it can simultaneously predict the rectangular bounding box of target location and the object's dense representation with binary segmentation. In order to further refine the segmentation representation, this thesis proposes a top-down refinement module to enhance segmentation details. Once the offline training is completed, the algorithm solely relies on a single bounding box

for initialization, and can simultaneously implement real-time visual object tracking and segmentation tasks. Despite the collaborative handling of multiple tasks, it has high processing efficiency around 55 frames per second. Finally, this thesis extends the above framework to multi-object tracking scenarios and achieve unsupervised video instance segmentation.

These innovations contribute to leading evaluation results on some publicly available tracking benchmarks, and most are the best at the time. The proposed lightweight network design enables the algorithm to achieve leading accuracy while maintaining a real-time speed that is beneficial to practical applications. In addition, some other computer vision applications (e.g., action recognition) can take advantage of our proposed methods.

Keywords: Visual Object Tracking, Siamese Network, Correlation Filters, End-to-End Learning, Attention Mechanism, Instance Segmentation

目 录

图形列表

表格列表

符号列表

字符

符号	描述	单位
$\mathbb{R}$	实数	-
$\mathbf{I}$	图像帧	-
$\mathbf{x}$	目标图像区域	-
$\mathbf{z}$	搜索图像区域	-
$\mathbf{w}$	判别相关滤波器参数	-
λ	判别相关滤波器正则系数	-
θ	卷积神经网络参数	-
$\mathbb{1}$	全 1 阵	-

算子

符号	描述
$\star$	循环相关操作
$\odot$	Hadamard 乘积
$\cdot^*$	复数矩阵共轭
$\hat{\cdot}$	离散傅里叶变换
$\mathcal{F}$	离散傅里叶变换函数
$\mathcal{F}^{-1}$	离散傅里叶变换的逆变换函数
∂	偏导数

第1章 绪论

视觉目标跟踪是计算机视觉感知中的重要组成部分，具有广泛的应用前景。视觉目标跟踪算法融合了图像信号处理、机器学习、数值计算以及人工智能等相关领域的关键技术。其研究内容是以视频序列图像帧为研究对象，关注于视频场景中所存在的目标对象，通过在连续视频帧之间创建基于位置、速度、颜色、纹理、关键点、形状等特征的关联匹配，建立目标对象在帧间的关联关系。视觉目标跟踪问题经过数十年的发展，逐步形成了以基础应用需求不同而衍生的多种特定类别目标跟踪研究。例如，以具有任意类别的目标物体作为跟踪对象的模型非固定式在线目标跟踪研究；以安防监控场景中的人或车辆为研究对象的模型固定式在线或离线多目标跟踪研究；针对智能手机中视频图像编辑应用所需的精确分割作为输出的视频目标跟踪与分割研究；针对行为分析场景中以人的关键点运动为中心的人脸或人体关键点跟踪研究；针对工业机器人生产场景展开基于深度（depth）图像的RGB-D跟踪研究；针对自动驾驶场景中的激光雷达点云视频序列的3D点云目标对象跟踪研究等等。

本文主要针对模型非固定式在线视觉目标跟踪开展研究。由于目标对象具有较为广泛的类别分布，因此对于通用目标对象的特征表示提取具有重要的研究意义。同时，这类问题仅在第一帧提供对目标物体的位置描述，而单一图像帧信息对于在立体空间中存在的目标描述通常都不够充分，因而算法可利用的先验信息较为匮乏。此外，为了适应目标场景发生的光照变化、视角变化以及目标物体形态的变化，需要算法在线完成对目标的表观建模与更新。然而在线跟踪的过程中，目标对象的矩形框描述包含一定的背景信息，同时算法存在不可避免的跟踪误差，使得跟踪模型的在线更新过程经常出现误差累积直至漂移（drift）现象发生。因此对于目标对象的精确描述以及利用物体边缘信息进行及时修正有助于长时间跟踪的鲁棒性提升。基于上述分析，本文以高效的孪生网络目标跟踪架构为研究对象，提出了多种有效的特征表示学习方法，实现了端到端的跨层级特征融合表示以及类脑注意力机制建模。同时我们将图像分割思想引入到目标跟踪对象的状态表述中，拓展了目标跟踪的表述形式，并首次构建了视觉目标跟踪与视频目标分割的一体化处理框架，建立了目标跟踪新范式。

在本章的后续内容中，首先介绍了视觉目标跟踪问题的研究背景与意义，然后梳理了模型非固定式在线视觉目标跟踪算法的研究现状，随后阐述了本文的研究内容与贡献，最后列出本文的组织结构与安排。

1.1 研究背景与意义

近年来，数字媒体的快速兴起以及智能手机的广泛普及，加速了人们对于视频图像内容的获取与传播。同时由于计算能力与存储技术的大幅革新，廉价且高质量的视频采集设备也遍布于我们生活的各个角落。高性能图形处理器（GPU）与张量处理器（TPU）等并行计算架构的出现，进一步加速了边缘端以及云端的处理效率。这一系列基础架构设施的升级，使得与计算机视觉相关的研究也得到了极大地发展，同时对于海量视频内容的检测、识别与跟踪等基础功能应用为人们的生活提供了极大的便捷。视觉目标跟踪作为计算机视觉技术的重要环节，广泛应用于视频监控、交通管理、手机应用、自动驾驶、虚拟现实以及人机交互等相关领域中，蕴藏了巨大的应用前景和商业价值。图1.1展示了视觉目标跟踪的典型应用场景，下面我们将分别介绍这些应用场景：

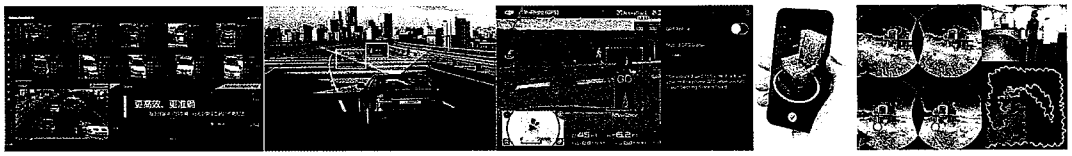


图 1.1 视觉目标跟踪在监控、自动驾驶、无人机、智能手机、虚拟现实领域应用示例。

- 安防监控领域长期以来对视频跟踪有着极高的需求，国内多个城市先后投入大量资金开展“智慧城市”建设，其数字化核心之一的安防监控工程的建设正在蓬勃发展。城市中海量的视频端口对人民生活的每时每刻进行精确地记录，有效守护了城市的公共安全。面对海量的视频信息，只有充分利用计算机视觉技术分析监控中的人与车辆的行为轨迹，才能在关键时刻进行及时检索查询。例如，杭州市推行的“雪亮工程”项目在重大活动的安全保障中发挥了重要作用。在智慧交通场景中，视觉目标跟踪服务于交通流量控制、车辆异常行为检测、行人行为判定、智能车辆识别等多个方面。

- 在新兴的自动驾驶领域，随着电动汽车的逐步普及，人们对于高级别（L3到L5）的自动驾驶的需求不断攀升。在2019年，国内电动车厂商蔚来汽车在其生产的电动车中发布了自动辅助驾驶功能。百度以及图森等科技创新企业都推

出了相应的自动驾驶系统，大量具有自动驾驶功能的工程车辆已经进行了实际开放道路测试。麦肯锡预测国内自动驾驶服务收入在 2030 年将会达到万亿级别。自动驾驶车辆对场景的识别、跟踪和预测对最终的行为决策有着至关重要的影响。通过视觉目标跟踪捕获车辆周围环境中的行人、车辆以及其他障碍物的运动轨迹状态是自动驾驶中不可或缺的基本功能。

- 在无人机以及智能机器人领域，高可靠性的控制系统已日趋完善。给予这些硬件平台相匹配的智能系统才能最大程度发挥硬件平台的先进性。大疆创新在 2015 年发布的精灵 3 无人机上就实现了自主跟随功能，无人机通过视觉目标跟踪技术实时捕获目标的位置，进而实现自动跟拍功能。在仓储机器人领域同样有着大量自动跟随或自主路径规划的需求，这些功能的实现都依赖于对场景中的人或特定目标物体的实时跟踪分析。随着机器人的智能化水平不断提升，越来越多智能产品的应用将大幅度减少了人员重复性工作消耗。

- 在视频编辑领域，随着 5G 技术的不断普及，视频直播、数字流媒体等领域的快速发展，视频内容产业出现了爆发性增长。智能手机中更加个性化的视频编辑需求也不断增加，视频场景中关于人脸以及肢体的运动跟踪捕捉为后续的特效渲染提供了极为重要的信息。此外，在虚拟现实应用 AR/VR 场景中，需要对场景结构进行运动分析，实现跟踪定位与场景重建。在虚拟场景中实时实现对人体姿态以及空间平面进行高精度的跟踪估计显得尤为重要。

除了上述典型应用需求外，视觉目标跟踪领域关于表观、时序以及立体空间的研究对计算机视觉的发展起到积极作用。计算机视觉领域的顶级会议 CVPR、ICCV、ECCV 以及顶级刊物 TPAMI 和 IJCV 都将视觉目标跟踪定义为重要研究方向，每年有超过五十篇相关文章刊登在这些会议期刊中。由于视觉目标跟踪具有广阔的应用价值以及理论价值，众多研究机构以及科技创新公司都投入到这一领域的研究中。近三十年来，视觉目标跟踪领域取得了很多重要突破，产生了众多的研究成果。但是，视觉目标跟踪领域依然存在着很多理论和技术问题有待解决，特别是跟踪过程中嘈杂背景干扰、运动模糊、光照变化、目标非刚体形变旋转以及障碍物遮挡等在开放环境中遇到的复杂挑战。所以如何能够自适应、实时和鲁棒地跟踪目标一直是广大研究人员亟需解决的问题，依然具有很高的研究价值和研究空间。

1.2 国内外研究现状

近年来，视觉目标跟踪领域呈现快速发展的趋势，研究人员提出了许多创新的跟踪方法与框架，但该问题的研究具有较为清晰的组织架构。视觉目标跟踪算法通常利用初始帧标签样本来构建系统模型，主要由基于运动模型的候选样本（proposal）生成算法、区域表示的特征提取算法、进行最终跟踪决策的表观模型以及模型更新方法这四个部分组成（如图1.2）。同时，由于视觉目标跟踪具有广泛的现实应用场景，大量的实时响应需求使得跟踪速度也作为视觉跟踪算法的重要评价指标。本章在对算法进行分类梳理的同时会着重对比不同算法的运算效率，这些经验分析引导了本文的相关算法设计。

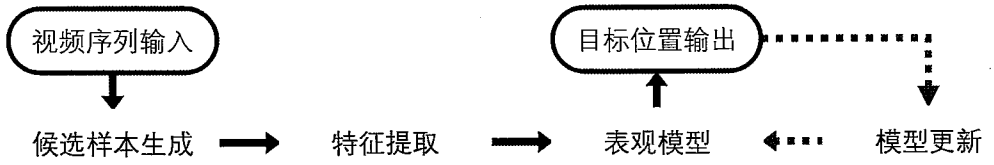


图 1.2 视觉目标跟踪算法基本流程示意图。其中，虚线部分表示可选操作。

我们首先将视觉目标跟踪流程进行形式化描述。视觉目标跟踪问题需要在给定视频序列 $\mathcal{I}=\left\{\mathbf{I}^t\right\}_{t=1}^{T}$ 以及初始帧目标位置 $\mathbf{b}^1$ 后，通过跟踪算法估计目标对象在后续帧中的空间位置 $\mathcal{B}=\left\{\mathbf{b}^t\right\}_{t=2}^{T}$。视觉目标跟踪算法通常在视频序列的第一帧根据图像 $\mathbf{I}^1$ 和目标位置标签 $\mathbf{b}^1$ 启发式地构建初始训练样本集合 $\mathcal{S}_{train}^1=\left\{\left(\mathbf{x}_i^1,y_i^1\right)\right\}_{i=1}^{m}$，通过表观模型训练得到判别分类器或目标表观分布描述 $f^1(\mathbf{x})$。在后续图像帧 $\mathbf{I}^t$ 通过运动模型 $\mathcal{M}$ 生成候选样本集合 $\mathcal{P}^t=\left\{\mathbf{p}_j^t\right\}_{j=1}^{n}$，并利用表观模型最大化后验概率或判别得分以估计当前帧的目标位置 $\mathbf{b}_t=\underset{p\in\mathcal{P}^t}{\operatorname{argmax}} f^{t-1}(\mathbf{p})$。按照预测结果更新样本集合 $\mathcal{S}_{train}=\mathcal{S}_{train}^{t-1}\cup\mathcal{S}_{train}^{t}$，并在线更新表观模型为 $f^t(\mathbf{x})$，依次迭代直到视频序列结束。视频目标跟踪的各个环节的选择都会显著影响算法的整体精度与效率，下面我们将依次分析这些核心环节。

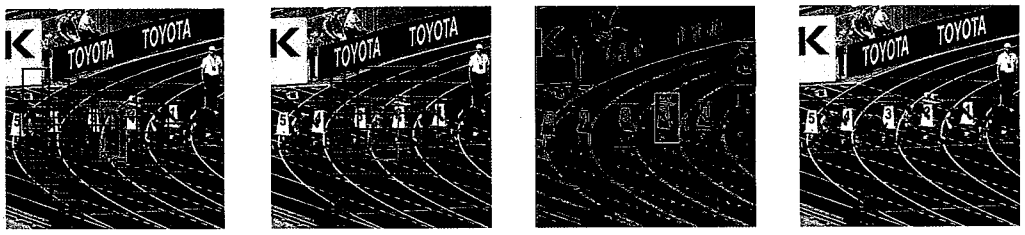
1.2.1 视觉目标跟踪的候选样本生成

视觉目标跟踪任务中表观模型训练样本集 $\mathcal{S}_{train}$ 的生成以及后续的候选样本集合 $\mathcal{P}^t$ 生成都涉及到对原始图像空间 $\mathcal{I}$ 的大量图像块采样。目标跟踪中通常采用粒子滤波（particle filters）模型 [20]、滑动窗口模型 [7, 21] 以及基于物体边缘信息的提议生成模型 [22] 进行候选样本生成。

粒子滤波模型是一种基于蒙特卡罗采样方法的递归滤波器模型，通过分析模型所生成的一组具有权重的随机样本集合的后验概率来估计目标在当前帧的状态，该方法具有较为严谨的数据模型理论保障。在具体实践中，该类方法 [23–25] 通常假定目标在相邻帧的中心点位置以及长宽比的变化符合正态分布，并对发生变化的状态进行采样估计。因而这类方法可以有效地预测目标的尺度以及长宽比的变化，具有较好的形变适应性。文献 [23] 将粒子滤波采样与灰度特征相结合，利用增量主成分分析算法进行表观模型构建，取得良好的目标跟踪效率。其后的大量传统目标跟踪算法 [24, 26] 均采用这种模型进行采样。但是粒子滤波模型的采样结果在空间位置上具有较强的随机性，因而不利于深度学习特征的高效提取，影响了这类算法 [10, 25] 的运行速度。直到 2018 年中，文献 [27] 提出共享特征计算方法，利用目标检测领域中提出的感兴趣区域对齐操作（ROI Align）[28] 将粒子滤波采样图像的特征进行高效提取，使得基于粒子滤波的深度学习模型优化到可以实时运行。

研究人员为了改善后续特征提取操作的效率，提出使用滑动窗口模型进行规则化密集采样。滑动窗口模型也被称作局部均匀采样模型，该模型假设目标在当前帧相对于上一帧的每个均匀移动的空间位置具有相同的先验概率。这种方法有效地建模了目标的平移变换，同时其均匀采样的特性非常利于结合卷积神经网络进行特征提取。算法通过这种采样方式可以有效地实现候选目标间的特征共享，节约了计算资源。文献 [21] 提出采用循环移位的采样方式进行训练样本收集与测试，通过快速离散傅里叶变换高效训练滤波器以及实现候选目标测试，取得了超过 600 帧每秒（frames per second，简称 fps）的运算速度，这也为相关滤波器算法 [12, 29–31] 的普及做出了重要支撑。同时，近期取得较快跟踪速度的孪生网络跟踪框架 [7] 也基于滑动窗口模型进行图像块匹配。基于滑动窗口的采样方法存在的问题是缺乏对尺度空间以及目标长宽比变化的建模，这类算法通常采用在多个尺度分别构建滑动窗口来解决对尺度空间建模缺陷 [7, 32]，但该方法仍然难以适应目标的长宽比变化以及旋转变化。

上述两种采样方式主要依赖于对目标位置先验分布的建模 $\mathcal{P}^t = \mathcal{M}(\mathbf{b}_{t-1})$，均没有利用图像的表观信息，需要采集大量的样本来实现对目标的精确位置估计，同时也难以扩大搜索范围。文献 [22] 提出使用图像的边缘轮廓特征生成少量的高质量候选窗口，利用图像边缘信息节省对简单负样本的判断消耗，因而可



(a) 粒子滤波模型　(b) 滑动窗口模型　(c) 提议生成模型　(d) 单一采样模型

图 1.3 在视频序列 *Bolt* 上四种不同采样方式的跟踪结果对比展示。

以将搜索范围显著扩大至全图。这种采样器同时使用当前帧图像以及目标在上一帧的位置进行建模 $\mathcal{P}^t = \mathcal{M}(\mathbf{I}_t, \mathbf{b}_{t-1})$，较粒子滤波模型和滑动窗口模型相比更为复杂，因而并没有得到较大范围的推广使用。但这种通过图像信息过滤简单负样本的建模方式启发了后续的基于级联采样的跟踪算法 [33, 34]。

除了以上三种常用的采样方式外，由于深度学习建模能力的不断提升，文献 [35] 提出在当前帧只采样单一的搜索区域，通过神经网络直接预测目标的位置信息，首次实现了基于深度学习的目标跟踪的实时运行。该方法可以视为利用深度神经网络集成样本选择、特征提取以及表观建模三个步骤。因而其模型训练较难实现，算法性能表现并不理想。

以上四种方法主要从采样规则的角度对采样方法进行分类，我们在图1.3中给出了一组对照实例。上述方法的空间位置模型都是采用了静止运动模型，一些工作 [36, 37] 采用匀加速度运动模型对物体的运动进行建模也带来了性能提升。综上所述，对于目标的采样需要平衡考虑：随机的采样方式有助于模型更好的适应尺度变化，但会给算法速度优化带来巨大挑战；过于密集的采样通常会带来更好的跟踪精度，但是会显著降低速度，影响实用性；过于稀疏的采样尽管可以有效提升运行速度，但会使得后续模型学习难度极大增加。本论文所有提出的方法均基于滑动窗口模型进行采样，该模型在精度和速度两个方面表现较为均衡。在第 §4章中，我们提出使用分割输出改善滑动窗口模型对于尺度旋转估计的不足。

1.2.2 视觉目标跟踪的特征表示与提取

在视觉目标跟踪任务中，良好的表观模型可以将目标与背景进行有效区分。Wang 等 [38] 通过分析视觉目标跟踪中的不同部件，实验证明了特征提取模块对于跟踪精度的影响最为重要。因而，目标对象的特征提取可以视作视觉目标跟踪

中的核心问题，算法需要构建对于光照、视角、形变等都较为鲁棒的特征表示。在本节中，我们将视觉目标跟踪算法关于特征表示的发展分为三个阶段：（1）首先是采用经过视觉领域专家手工设计的（hand-crafted）特征表示阶段，描述子从颜色、纹理、形态、关键点等不同视角对物体的表观进行描述。（2）然后，随着深度学习的快速普及，研究人员 [39] 发现卷积神经网络（Convolutional Neural Network，简称 CNN）特征可以有效区分不同类别的目标对象。跟踪算法直接将针对图像分类训练的深度学习特征替换原有的手工设计特征，即可取得极大的性能提升。这一阶段的主要表现是特征提取器与跟踪任务相关数据以及表观模型的训练相分离，没有针对跟踪问题进行定制化设计。（3）随后，研究学者将目标跟踪问题的特征提取以及表观模型贯通起来，实现端到端的特征表示训练，使得基于深度学习的特征表示对于跟踪场景得到进一步优化。目前该方法是现阶段跟踪研究的主流选择。本节将对上述阶段划分的典型工作进行具体介绍。

颜色描述特征：视觉目标跟踪算法是以视频中的图像帧为研究对象，而在计算机中图像帧通常是在 RGB 颜色空间进行呈现。颜色特征的使用具有较强的可解释性以及较为明确的区分度。该特征可以有效应对目标发生的图像平面内旋转、非刚体形变，具有较好的形变适应性。通常采用的颜色描述空间包括 RGB 颜色空间、YUV 颜色空间、HSV 颜色空间、LAB 颜色空间、灰度空间以及颜色名称（color name）[40] 空间。文献 [23] 采用低分辨率灰度图像特征对样本进行特征表示提取，将低分辨的图像特征展成一维数组后进行增量主成分分析。文献 [41] 使用灰度特征向量结合稀疏表示对目标进行建模。相关滤波领域的开创性工作 MOSSE[21] 同样基于灰度特征表示，使用离散傅里叶变换求解回归损失，运行速度达到近 700fps。基于灰度图像的跟踪算法一般具有较快的运行速度，但是通常难以满足对于光照变化的鲁棒性。在文献 [42, 43] 中，研究人员提出使用基于 HSV 空间的颜色直方图作为特征描述，结合贝叶斯概率模型建模目标前景与背景的颜色分布用来判别目标。Danelljan 等人 [44] 将基于灰度图像的单通道相关滤波扩展到多通道相关滤波，采用颜色名称特征 [40] 对目标图像进行描述，显著提升了跟踪鲁棒性。

纹理描述特征：纹理描述特征通常反映图像像素块的统计信息，对于光照变化引起的灰度值变化具有稳健性。文献 [45] 提出了一种基于协方差的目标描述方法，该方法成功地跟踪了非刚体形变目标。局部二值模式（Local Binary Patterns,

简称 LBP）[46] 将图像像素与其附近的像素进行比较编码，具有较强的光照不变性。Haar-like 特征 [47] 对图像的子区域之间的差值进行统计分类，可以有效应对目标的尺度和旋转变化。文献 [48] 提出使用 Haar-like 特征结合在线半监督增强（boosting）算法来更新判别模型。此外，文献 [49] 提出结构化输出的支持向量机作为表观模型，并使用 Haar-like 特征表示。该算法取得了 OTB-2013 数据集 [3] 发表时的最好性能，证明了该特征的有效性。

目标边缘特征：边缘特征相对于纹理特征具有较好的位置敏感特性，有利于跟踪过程中精确的目标位置估计。方向梯度直方图特征（Histogram of Oriented Gradient，简称 HOG）[50] 在密集图像网格上高效统计每个网格单元内的方向梯度，具有较好的尺度不变性。文献 [29] 提出使用基于 HOG 的多维特征来扩充原有的基于灰度特征的相关滤波算法 [51]，在 OTB-2013 数据集中提升了超过 20% 的跟踪精度。该方法的优异性能引起研究学者的广泛关注，后续大量的基于相关滤波器的跟踪算法 [12, 32, 52–55] 延续使用 HOG 特征。

光流特征：光流特征显示了图像中最小粒度对象的运动，该类方法对图像中的像素点运动位移进行估计。光流估计通常分为稀疏光流估计和稠密光流估计。Shi 和 Tomasi[56] 提出使用角点的检测进行稀疏光流跟踪。而稠密光流计算复杂度较高，通常难以实时运行。在深度学习发展过程中，Dosovitskiy 等人 [57] 提出使用深度网络来精确估计稠密光流。Zhu 等 [58] 将上述方法提取的稠密光流用作目标跟踪中的候选窗口运动估计。

预训练深度特征：随着深度卷积神经网络在图像识别任务上的成功应用 [39, 59–61]，基于深度学习的算法在计算机视觉的各个领域 [28, 62–64] 逐步兴起。深度学习特征与传统手工设计特征最大的不同来自于其从大规模数据中学习的数据驱动特性，因而深度学习特征具有良好的鲁棒性与判别性。在将深度学习引入到视觉目标跟踪的早期研究中，算法通常将图像分类或自监督学习所使用的特征直接迁移到视觉目标跟踪任务中。早在 2013 年，Wang 等在文献 [65] 中提出在大量自然图像数据中使用无监督方法学习具有鲁棒性的堆叠编解码网络特征，在线跟踪过程中将该特征迁移到特征提取任务中。文献 [66] 提出使用全卷积神经网络跟踪架构，将基于深层语义特征的检测器与基于浅层特征的判别模型相结合，在此基础上提出特征选择模块用来去除具有噪声的特征层。Ma 等 [30] 提出将预训练的深度学习特征 VGGNet[59] 与相关滤波器相结合，利用不同层级特

征进行逐步精确定位。文献 [67] 采用预训练的深度学习特征与具有空间正则的相关滤波器结合，取得了 VOT-2015[4] 挑战赛的冠军。后续的基于相关滤波的工作 [68, 69] 大都参考了使用预训练的深度学习特征与相关滤波器结合的策略。而直接采用预训练好的神经网络进行特征提取的弊端是网络结构通常较为复杂，直接移植到目标跟踪任务上会严重制约目标跟踪的实用性。Danelljan 等 [12] 提出通过分解卷积操作、统计训练样本分布以及保守的更新策略来优化判别相关滤波器学习，取得了 20 倍的速度提升。但当算法使用深度特征表示时，仍然无法达到实时跟踪速度。

基于视觉目标跟踪任务端到端学习的深度特征：除了上述直接将预训练好的深度学习特征迁移到视觉目标跟踪中的算法，针对目标跟踪设计损失函数进行端到端训练的跟踪算法 [7, 10, 25, 35, 70] 也逐渐占据主导地位。文献 [25] 提出针对多个视频序列学习独立领域知识的判别分支，同时共享底层特征表示的方式进行多领域（multi-domain）学习。该方法在线过程中固定共享参数的特征提取模块，针对特定目标训练领域自适应的跟踪分支，实现了高精度的目标跟踪。Han 等 [27] 通过加速特征提取过程，以及提出更具判别力的损失函数，将算法优化到可以实时跟踪。文献 [11, 71] 提出将相关滤波器当作网络中的一种可微分操作，将相关滤波器与神经网络连接进行端到端学习，但其网络设计缺乏针对目标跟踪的优化。文献 [7, 10, 35] 提出使用孪生网络对目标图像和搜索区域图像进行建模，通过直接回归目标位置或相似性度量的方式进行目标跟踪学习。Li 等 [70] 在全卷积孪生网络跟踪器的基础上提出利用孪生网络同时回归目标的坐标位置以及相似性度量，取得了良好的形变适应性。文献 [72] 通过离线过程中优化困难样本选择，将端到端训练的特征表示判别性能显著提升，取得了 VOT-2018[2] 实时跟踪挑战赛的冠军。文献 [73] 进一步对网络的主干结构进行调整，通过数据增强的方式解决了孪生网络学习中的位置偏见问题，消除了孪生网络跟踪算法的网络结构限制，极大地提升了孪生网络特征表示的模型容量以及网络架构灵活性。文献 [74] 通过分析孪生网络的感受野（receptive field）以及网络拓扑结构，设计了针对孪生网络的内部裁剪残差模块来优化网络学习。端到端的特征表示学习依赖于训练数据的支持，同时训练样本选择方式、网络结构设计以及损失函数设计都占据非常重要的地位。

多特征融合表示：单一的特征通常只在某一方面对图像进行描述，利用多

种特征融合来对图像进行全面描述的方式具有更强的鲁棒性。文献 [75] 提出使用 HOG 特征与颜色名称特征相结合，构成具有互补性的相关滤波跟踪器。文献 [76] 同时采用 HOG 特征、颜色名称特征和卷积神经网络特征进行互补增强，取得了显著的精度提升。近年来，在 VOT[2, 77] 竞赛中取得冠军的算法通常采用多个神经网络特征进行集成（ensemble）学习。

通过对视觉目标跟踪领域的特征表示方法进行梳理，我们可以发现研究人员在一个时期内会集中进行特征表示的相关改进，然后达到基本统一。现阶段的高性能跟踪算法主要使用基于深度学习建模的特征表示。算法通常采用较大的搜索区域，利用全卷积神经网络高效进行特征提取。

1.2.3 视觉目标跟踪的表观模型构建

不同于特征表示发展的快速统一，在视觉目标跟踪的各个时期都存在大量不同的表观模型方法，这也是视觉目标跟踪领域中最重要的创新所在。利用多种机器学习方法构建针对跟踪任务的表观与时序模型，从多个视角对跟踪问题进行形式化构建。但对于总体而言，视觉目标跟踪的表观模型大致可以分为产生式学习模型（Generative Learning based Models，简称 GLM）和判别式学习模型（Discriminative Learning based Models，简称 DLM）。基于产生式学习的模型主要从目标的样本分布估计出发，通过分析不同候选样本的后验概率进行比较，得出物体归属类别。而判别式学习模型主要对目标对象和背景的区别进行建模，直接得出每个候选样本的类别估计，用来确定目标位置。本节通过结合具体的算法实例来直观了解视觉目标跟踪在这两类模型中的发展历程。

1.2.3.1 基于产生式学习的表观模型

基于产生式学习的表观模型主要使用正样本学习模型参数，判别能力相对较弱。产生式学习方法主要统计目标表观分布，并在每一帧中通过最小化重构误差来搜索目标。在视觉目标跟踪早期研究中，Lucas 和 Kanade[78] 提出使用基于光照强度的模板匹配算法进行目标跟踪。为了改善该算法对光照以及形态变化的适应性，Matthews 等人 [79] 通过结合首帧信息以及上一帧的预测结果来更新模板特征，减少了跟踪漂移（drift）问题。

子空间学习：基于子空间的跟踪方法 [23, 80, 81] 被提出用来更好地适应目标表观变换。文献 [81] 提出了一个高效的 LK 跟踪算法，通过将目标对象映射到

低维子空间特征表示来跟踪不同光照条件下的目标对象。为了提升视觉目标跟踪的鲁棒性，文献 [80] 采用了一个较为鲁棒的误差范数，并提出使用预训练好的不同视角场景下的特征向量投影表示来进行跟踪。Ross 等 [23] 提出通过增量学习低维子空间表述来应对目标的表观变化。

稀疏表示：稀疏表示是常见的生成模型，同时稀疏表示理论也广泛应用于计算机视觉、信号处理、图像处理等领域。该方法的学习目标是找到一组足够稀疏同时具有最小化重构损失的向量空间作为目标的表示。稀疏表示方法通常需要先学习一个表示字典，在此基础上使用较为稀疏的系数与表示字典线性组合得到目标表示。通过交替优化字典学习与稀疏系数学习进行求解。Mei 和 Ling[41] 采用由目标模板和手工设计的简单模板组成的整体模板字典，通过解决多个 ℓ_1 最小化问题来确定目标位置。为了提升稀疏表示对于遮挡场景的处理能力，Zhang 等 [82] 提出基于粒子滤波框架的结构稀疏跟踪算法 SST。该算法利用局部的图像块与全局目标图像的内在关系，联合学习稀疏表示。全局稀疏表示利用目标的整体表示，并使用 ℓ_1 范数进行优化。局部稀疏表示利用局部图像结构之间的内在关系，共享学习词典。为了提升稀疏表示求解效率，文献 [83] 引入了一个最小误差边界策略用来减少需要求解的优化问题数量。Bao 等人 [84] 提出使用加速迫近梯度法（Accelerated Proximal Gradient，简称 APG）来有效地求解 ℓ_1 范数最小化问题。

1.2.3.2　基于判别式学习的表观模型

基于判别式学习的表观模型通常通过回归模型或者分类器进行学习，最大化前景目标与背景图像的距离，然后对候选区域进行判别决策。由于目标跟踪问题只在第一帧中给定目标的标注信息，这使得目标跟踪中的训练样本获取较为困难，而训练样本的质量直接影响判别模型的判别能力。此外，判别学习算法本身的建模能力对鲁棒的目标跟踪具有至关重要的影响，常用的基于判别式学习的跟踪方法包括使用支持向量机 [85] 进行分类学习的跟踪算法 [49, 86, 87]、基于多实例学习的跟踪算法 [26, 48]、基于判别子空间学习的跟踪算法 [88]、基于判别相关滤波学习的跟踪算法 [21, 29, 31, 89] 以及近期提出的基于深度学习的跟踪算法 [7, 10, 25, 35, 70]。本节根据跟踪算法的不同发展时期，主要介绍基于传统机器学习的目标跟踪研究、基于相关滤波器的目标跟踪研究以及基于深度学习的目标跟踪研究。跟踪算法的精度与速度一直以来都是算法在实际应用中所

需要重点考察的两个方面，本小节中主要对算法效率与判别能力进行分析。

基于传统机器学习的目标跟踪研究：传统机器学习方法主要采用基于 SVM、决策树、Logistic 回归等方法进行监督学习。早期目标跟踪算法将这些基础机器算法迁移到目标跟踪任务中。Avidan[86] 首先提出将支持向量机理论引入到目标跟踪中进行判别学习。Hare 等人 [49] 将其扩展为基于结构化输出的支持向量机模型进行在线学习。Zhang 等 [87] 提出基于不同历史信息维护多个 SVM 判别器来进行集成跟踪，通过熵最小化来选择专家跟踪器进行决策，该方法显著提升了跟踪器的时序鲁棒性。Gao 等 [24] 通过分析高斯回归过程，引入隐变量来协助跟踪过程决策。Wang 等 [38] 通过实验分析比较了 SVM 与 Logistic 回归等简单线性模型，验证线性回归模型的有效性。Zhang 等 [88] 提出使用基于判别性学习的图模型将目标特征表示投影到低维表示空间，通过图嵌入理论最大化正负样本集合距离，最后通过在低维表示空间中度量候选样本与模板样本的距离，确定目标位置。判别学习方法通过第一帧标签构建训练样本进行训练，并利用后续帧跟踪结果进行模型更新，其难点在于学习数据中含有大量噪声。Grabner 等人 [48] 使用半监督学习的在线强化（online boost）算法有效解决了由累计误差引起的漂移问题。Babenko 等人 [26] 提出使用在线多实例学习机制，通过对有噪声的样本集进行整体优化来消除不精确估计带来的学习偏差。Kalal 等人 [90] 将长时（long-term）目标跟踪问题分解为跟踪、学习与检测三个部分，提出新颖的 P-N 学习方法来应对目标漏检与目标误识，该方法达到了实时跟踪性能并被广泛应用于工程项目中。

基于相关滤波器的目标跟踪研究：基于相关滤波器的跟踪算法本质上采用岭回归学习，由于其使用循环移位的特殊采样方式，可以通过快速离散傅里叶变换在频域中高效求解，具有较快的运行速度，成为了视觉目标跟踪自 2011 年以来的研究热点。Bolme 等 [21] 首次提出使用最小化输出误差平方损失来学习滤波器，通过将时域信号转换到傅里叶频域快速求解，其运行速度达到 669fps。随后 Henriques 等 [51] 通过循环移位矩阵理论来解释相关滤波器，利用循环移位矩阵形式推导相关滤波学习过程，为相关滤波在视觉目标跟踪中的理论学习奠定基础。然后，Henriques 等 [29] 进一步将相关滤波中的线性回归分析扩展到核（kernel）回归分析，提出使用多维的 HOG 特征表示进行特征描述，同时保留了传统相关滤波的闭式解，取得较好的跟踪效率。

相关滤波器具有良好的平面移动建模能力，研究人员 [32, 89, 91] 将相关滤波的平面分析扩展到尺度空间分析。Zhang 等 [91] 通过时空上下文分析对相关滤波器进行建模，在频域推导了相关滤波器关于尺度形变的估计方法。Li 等 [32] 通过对多个尺度样本离散采样构建图像金字塔并分别测试来评估目标的尺度变化。Danelljan 等 [89] 在平面相关分析的基础上级联一维尺度相关滤波器用来估计目标尺度变化，通过两阶段的方式进行目标的精确跟踪。Zhang 等 [92] 在尺度位移空间的相关滤波学习基础上，通过引入图像块的对数极坐标系变换，增加模型对于目标旋转变化的建模，显著提升了模型对于旋转变化的适应性。文献 [93] 提出利用粒子滤波框架增加相关滤波器的采样多样性，有效适应了目标的尺度和长宽比变化，但其跟踪速度受到较大影响。文献 [68] 将支持向量机的损失函数与相关滤波学习相结合，同时提出了基于相关响应结果的状态估计方法，有效消除了干扰对象与遮挡物的影响。文献 [94] 将相关滤波中固定的高斯响应调整为适应目标状态变化的动态响应，通过同时优化目标响应以及滤波器参数减弱了算法的跟踪漂移问题。

此外，为了消除相关滤波学习中的边缘效应（boundary effects），Danelljan 等 [54] 对于滤波器增加空间正则约束，抑制滤波器在边缘处的幅值。Galoogahi 等 [53, 95] 提出使用和目标大小一致的二值（binary）模板来约束滤波器，通过增广拉格朗日方法进行优化求解。Lukezic 等 [55] 为了进一步适应不同形态的跟踪目标，提出使用颜色直方图计算前景概率分布，通过二值化的前景分割模板来正则滤波器学习，使得算法具有良好的目标形态适应性。

Danelljan 等 [12, 31] 在离散相关滤波器学习的基础上，提出连续相关滤波操作，通过多尺度特征的联合相关滤波分析得到目标的亚像素位移估计，有效提升了跟踪精度。在相关滤波器的特征使用方面，Danelljan 等 [44] 将常用的一维灰度特征表示扩展为多维颜色名称表示，增加了相关滤波器的鲁棒性。Ma 等 [30] 提出使用预训练的卷积神经网络进行特征提取，通过融合多层特征的相关响应对目标进行定位。由于神经网络特征的使用，其判别性得到显著提升。为了充分挖掘卷积神经网络特征表示的潜力，Danelljan 等 [96] 研究分析了多种数据增强方式对于基于深度特征的相关滤波器的贡献，通过混合多种数据增强方式来显著提升跟踪器在线学习判别性能。研究人员 [11, 71, 97] 通过将相关滤波进行端到端求解，在离线学习过程中对特征网络进行优化。此外，Chen 等 [98] 提出利

用卷积层在线学习来替代具有边缘效应的判别相关滤波器学习。Song 等 [9] 在此基础上提出空间残差和时序残差学习来修正滤波器参数。

由于相关滤波器本身具有较高的运行效率，研究人员通常会使用多个相关滤波器或将其与其他具有互补特性的跟踪器进行集成。文献 [52] 通过结合不同更新速率的相关滤波跟踪器实现目标的长时间跟踪。文献 [99] 提出使用关键点跟踪辅助基于全局特征的相关滤波器进行目标找回。文献 [100] 提出并行使用相关滤波跟踪器与高精度验证器进行长时间跟踪，通过验证器来识别目标状态并进行遮挡后的目标识别。文献 [101, 102] 提出使用多层卷积网络替代线性相关滤波分析，并结合神经网络来优化目标的矩形框重叠率，有效适应了目标的尺度变化，在多个数据集中取得当时最优精度。至此，相关滤波跟踪与深度学习跟踪发展基本统一，基于循环移位的线性回归模型逐步被多层卷积网络学习所取代。

基于深度学习的目标跟踪研究：利用深度学习作为表观模型的目标跟踪算法主要的特点是具有较大的模型容量，可以离线通过在大规模数据样本上的学习获取针对于视觉目标跟踪的领域知识。Wang 等 [65] 提出在离线过程中采集大量自然图片训练堆叠的自编码去噪网络，在线过程中利用网络的中层特征表示进行目标跟踪。Wang 等 [66] 通过分析全卷积神经网络的特征响应，提出使用神经网络高层语义特征进行鲁棒的目标类别识别同时使用底层网络进行精细的判别分析定位，通过稀疏特征分析去除无关或具有噪声的网络通道，以节约计算消耗。Nam 等 [25] 提出了一种基于卷积神经网络的多分支结构，离线过程中针对不同视频调整网络最后的全连接层，同时共享底层特征提取网络。该方法在线跟踪过程中通过粒子滤波进行采样，并在线训练领域自适应的跟踪分支。这种跟踪算法被称为多领域学习网络（Multi-Domain Network，简称 MDNet），相比于基于传统机器学习的跟踪方法以及相关滤波器方法在 OTB-2015[1] 数据集中取得了超过 10% 的精度提升，这也被认为是深度学习在目标跟踪领域快速推广的奠基之作。基于这种多领域学习方法，Nam 团队 [103] 提出使用树（tree）状结构维护多个卷积网络分支的推断与更新，取得了 VOT-2015[4] 和 VOT-2016[13] 视觉目标跟踪挑战赛的冠军。Fan 等 [104] 将多领域学习网络的卷积特征提取器替换为使用递归神经网络构建的特征网络，增强对目标的结构特征提取。Teng 等 [105] 提出在多领域学习的基础上增加时空上下文约束，以提升算法的时序鲁棒性。此外，Choi 等人 [106] 提出使用元学习（meta-learning）机制来提升网络的

更新速度。Song 等 [107] 提出在判别学习的基础上，增加对抗学习（adversarial learning）机制，通过网络自适应学习最具判别性的区域，使其具有较强的遮挡鲁棒性。上述方法通过在特征表示、判别分支更新以及图像结构学习等方面对 MDNet 进行了有效提升。但是由于上述算法仍然采用粒子滤波进行独立图像采样并分别计算图像块的特征表示，以及需要进行在线训练深度神经网络，因而该类方法的速度通常保持在 1fps 左右，这严重影响了深度学习跟踪算法的工程应用。直到 ECCV 2018，Jung 等人 [27] 通过使用共享特征计算，结合精确区域特征提取模块才使得算法的速度可以达到实时运行。

为了改善在线更新带来的计算负担，文献 [108] 提出使用神经网络直接预测判别网络的权重。Held 等 [35] 将目标跟踪问题转化为矩形框回归学习，直接通过孪生网络学习目标的相对位移，这种方法去除了大量的候选窗口采样过程，实现了首个基于深度学习的超实时目标跟踪算法。借鉴人脸识别领域方法，Tao 等 [10] 将视觉目标跟踪问题转化为识别验证问题，利用孪生网络训练特征提取模块，通过对比候选区域特征与目标模板特征进行目标识别。Bertinetto 等 [7] 进一步使用全卷积的网络结构优化上述孪生网络，对搜索区域进行高效特征共享，达到了超实时的跟踪速度（86fps）。同时，由于在大规模数据集中进行离线训练，该方法具有良好的泛化性能，取得了 VOT-2017[8] 实时目标跟踪挑战赛的冠军。基于这种全卷积孪生网络的结构框架，He 等 [109] 提出使用多层次特征进行相关分析，通过对网络特征的不同通道进行加权来调整网络的类别属性。Zhang 等 [110] 提出结构化的孪生网络特征表示方法，通过条件随机场对图像块的不同空间位置的特征进行关联，增加了算法对于目标旋转的适应性。Sun 等 [111] 提出使用对抗学习方法增加孪生网络学习中的样本数量，通过在线困难样本挖掘提升网络的判别能力。Dong 等 [112] 通过对孪生网络的损失函数进行分析，提出基于三元组（triplet）的损失函数用来增强网络的判别力。此外，为了增加网络的时序适应能力，Yang 等 [113] 提出使用基于长短时记忆的网络结构来动态调整孪生网络的匹配特征。

此外，除了按照产生式学习和判别式学习划分表观模型外，表观模型建模区域的粒度选择也可作为目标跟踪表观模型的重要划分指标：（1）通常目标跟踪将整个目标样本进行建模跟踪，该方法称为整体模型（holistic model），其典型代表为基于相关滤波的目标跟踪算法 [21, 29, 31, 51, 54, 89] 以及基于孪生网络的跟踪

算法 [7, 10, 70]。（2）为了应对目标的大幅度形变与旋转，研究人员提出将目标的不同部件进行分别建模跟踪再整合得到跟踪结果的方法，该方法被称为基于部件的模型（part-based model）。文献 [114] 提出通过多个相关滤波器分别跟踪目标局部区域，并采用序列蒙特卡洛算法估计各区域的跟踪可靠度，利用较为可靠的区域的跟踪结果估算目标的整体运动。该类方法通常需要维护多个跟踪器，运行效率较低。（3）此外，将跟踪目标的描述进一步细化到图像像素或关键点级别进行跟踪的方法被称为基于像素的模型（pixel-based model），这类方法在早期的跟踪研究中 [78] 被使用，近期基于像素的跟踪模型 [75, 99] 通常结合整体模型一起跟踪目标对象。

1.2.4 视觉目标跟踪的更新策略

视觉目标跟踪算法对视频序列进行建模，而目标的表观会随着场景光照以及摄像机视角发生剧烈变化。为了考虑目标以及周围场景的表观变化，Ross 等 [23] 提出高效的增量子空间学习进行更新。基于判别式学习的方法同样需要在线收集样本，进行模型更新，而由于样本集的不断扩充，在线跟踪的内存消耗以及速度会受到极大影响。Hare 等 [49] 提出使用预算机制来防止在跟踪过程中发生支持向量数量的无限增长。

对于效率较高的判别相关滤波器学习，研究人员 [21, 29, 30] 通常采用一种固定比例系数线性加权的方式进行更新，由于只需要维护一个历史滤波器和进行当前帧滤波器估计，该方法的内存消耗以及计算开销都较小。但是，这种更新算法对于视频序列的更新权重较为固定，无法阻止目标遮挡带来的模板污染。此外，由于跟踪过程中不可避免存在一定的估计误差，在线更新收集到的样本噪声随着时间显著增加，产生跟踪漂移（drift）问题。Ma 等 [52] 提出结合使用一个快速更新的短时跟踪模板以及一个缓慢更新的长时跟踪模板的长时跟踪策略。Danelljan 等 [115] 提出联合优化更新权重以及滤波器的策略，对于不同帧实时调整权重来有效避免遮挡等情况对于更新学习的影响。为了减少更新样本的冗余，文献 [12] 提出使用混合高斯模型来学习训练样本分布，只采样少数具有代表性的目标模板进行更新训练，取得了显著的速度提升。

随着视觉目标跟踪研究逐步进入深度学习时代，深度神经网络的在线训练对目标跟踪的实际应用带来了极大的挑战。Han 等 [25] 提出的 MDNet 跟踪算法不仅离线过程中需要大量的样本来训练鲁棒的共享特征参数，其在线过程中也

需要采样数千个样本进行随机梯度下降来训练领域自适应分支。尽管后续更新过程中只采用少量的样本进行微调，但神经网络的参数量过大导致优化过程较慢，使其跟踪速度约为 1fps。这严重阻碍了深度学习跟踪算法的实际应用。研究人员 [7, 10] 将在线训练神经网络进行判别分类问题转向将跟踪定义为验证问题，通过孪生网络进行度量学习，查询比对候选样本与目标样本的相似度进行目标跟踪。离线过程中，通过深度神经网络对于大量目标样本对的光照、视角变化进行学习，在线过程中完全不更新模板，使得这类方法取得了较高的跟踪效率。Yang 等 [113] 提出使用长短时记忆网络来建模孪生神经网络更新过程，通过网络自主学习更新策略，显著提升了孪生网络的时序鲁棒性。

1.2.5　视觉目标跟踪的输出模式

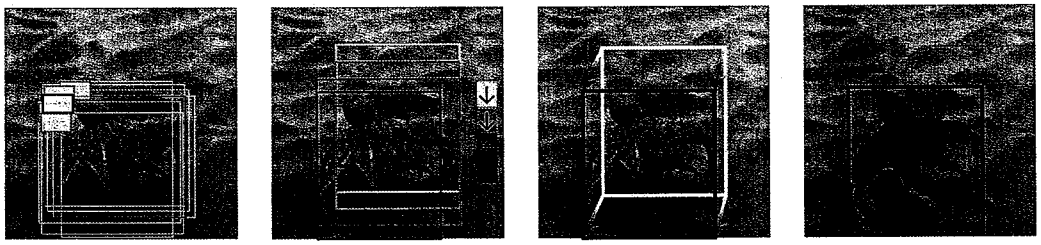
除了上述根据目标跟踪中的不同环节进行划分的介绍外，本节着重关注目标跟踪的输出模式。研究人员需要结合视觉目标跟踪任务特点，设计适当的输出模式用来完成目标跟踪任务。其中最典型的方法是基于前景分类或相似性度量回归的跟踪算法，该类算法通过最大化候选样本得分寻找目标对象。而随着深度学习的快速发展，网络模型的建模能力得到了大幅度提升，模型可以更好地适应目标的形态变化。同时数据标注的规模以及标注质量在不断提升 [6, 19, 116, 117]，我们可以训练网络产生更加精细多样的目标表示。因而近年来跟踪算法框架扩展出利用行为决策分类的跟踪框架、利用矩形框回归学习的跟踪框架以及通过前景目标分割进行定位的跟踪框架。

基于分数预测的输出模式：通常视觉目标跟踪算法将目标跟踪问题形式化为前景与背景的分类问题或目标概率预测问题，通过表观模型对候选样本进行得分估计，在候选样本集中选择最大得分的候选样本当作当前帧的目标估计输出。文献 [23] 使用增量主成分分析对目标子空间进行建模，采用粒子滤波进行目标采样，通过在子空间投影中度量候选样本与目标模板的距离进行目标选取。基于相关滤波的跟踪算法 [21, 29] 通过循环移位的滑动窗口比较不同位置的响应，通过最大响应位置来定位目标在当前帧的中心点位置变化。文献 [7, 10] 分别通过粒子滤波和滑动窗口对候选区域进行采样，通过孪生网络预测目标和模板之间的相似性来搜索目标位置。通常基于滑动窗口的采样方式只建模了物体的平移变换，如果需要精确地估计目标的尺度变化，需要增加对于尺度空间的采样。文献 [32] 提出使用多尺度金字塔测试来改进相关滤波的尺度估计。文献 [7]

同样采用多尺度测试来提升孪生网络跟踪算法的尺度适应能力。文献 [118] 提出增加旋转测试样本来增强孪生网络对于目标旋转的建模，显著提升了孪生网络的跟踪精度。文献 [43] 基于颜色直方图构建贝叶斯模型用来估计每个图像像素属于目标前景的概率，通过积分图快速估计每个滑动窗口的目标前景概率总和。尽管该算法在目标像素上进行概率求解，但最终的跟踪决策是依赖于滑动窗口内的累计相似性得分进行判定。基于分数预测的输出模式具有较强的可解释性，同时也易于实现，长期以来是视觉目标跟踪中最常见的输出模式。其精度性能与采样数量密切相关，高精度的目标状态估计需要大量的采样粒子 [25]。由于难以同时对于位移空间、尺度空间、目标长宽比等进行高密度的采样，该类算法在目标发生快速尺度变化的时候容易发生错误的目标位置估计。

基于动作决策的输出模式：视觉目标跟踪也可以构建为前后帧位置关系的动作决策问题。文献 [119] 提出在上一帧目标矩形框位置的基础上预测目标向上、向下、向左、向右、放大或者缩小等几种变化行为，逐步迭代找到目标在当前帧的位置。同时该方法将动作决策与增强学习（reinforcement learning）相结合，离线过程中使用少量标注即可进行网络训练。该方法在动作决策建模过程中只能使用离散的固定操作进行迭代，因而其决策空间对于目标的尺度变化只能得到离散的估计，对于精细的亚像素级别目标运动以及长宽比变化难以建模。

基于矩形框回归的输出模式：不同于上述两种输出模式，对于视觉目标跟踪的定位问题也可以采用直接回归矩形框坐标的方法进行建模。算法利用表观模型直接预测目标在当前帧的真实位置 $\mathbf{b}^t$ 与候选窗口 $\mathbf{p}$ 的坐标位置差异，充分利用表观模型的回归能力。文献 [25] 最早通过粒子滤波的采样方式结合领域自适应的神经网络进行判别分类，随后将物体检测算法 R-CNN[120] 中的线性回归模型应用到目标跟踪的矩形框回归过程中，对少数具有较高得分的候选样本进行坐标回归，通过这些回归估计的均值来确定目标的最终位置，算法因而取得了非常突出的跟踪精度。文献 [35] 进一步简化跟踪问题，通过直接将目标图像和搜索图像构成的样本对输入到神经网络来学习目标的位置变化，实现了首个基于深度学习的实时目标跟踪算法。但是该方法在训练过程中需要对所有的位移变化进行穷举，训练过程较为困难，过于简洁的实现方式让网络参数难以收敛。文献 [70] 将目标检测中的区域提议网络 [62]（region proposal network）引入到目标跟踪中，充分结合全卷积孪生网络优势，利用分治法实现矩形框位置回归。具



(a) 基于分数预测的输出　(b) 基于动作决策的输出　(c) 基于矩形框回归的输出　(d) 基于分割预测的输出

图 1.4 在视频序列 *octopus* 上四种不同输出模式跟踪算法的跟踪结果对比展示。

体而言，算法首先通过全卷积孪生网络预测目标在不同锚点（anchor）位置的概率，然后基于锚点进行矩形框回归。该方法有效降低了回归学习的难度，同时引入了判别学习来增强网络对于目标表观的建模。在此基础上，文献 [121] 借鉴物体检测中的单阶段（one-stage）方法去除了基于锚点的回归学习，通过优化矩形框回归的损失函数来增强回归预测精度。基于矩形框的回归预测理论上可以实现对于目标矩形框状态的无偏估计，该方法可以有效降低候选样本的采样数量。

基于分割预测的输出模式：目标跟踪任务的本质是对目标在不同帧之间的对应关系进行建模，而矩形框的表述只是算法对所要跟踪的目标在当前帧位置的一种简单近似描述。矩形框描述的优点在于其具有较为简单的表述方式，该描述对于用户输入、传输以及数值计算都较为友好。但在跟踪过程中的矩形框描述通常会引入许多无关的背景，VOT-2015 数据集 [4] 提出采用旋转矩形框对目标进行表述与标注。研究人员 [122, 123] 提出将视觉跟踪问题与视频分割问题相结合，利用超像素对图像块建模关联，最终输出精确的分割表述。文献 [124] 使用多个基于全卷积孪生网络 [7] 的跟踪器来跟踪目标部件，并整合不同部件的跟踪结果分别分割得到最终的分割结果，在视频目标分割数据集 DAVIS-2016[14] 中取得了良好的分割表现。

在图1.4中，我们直观地对比了上述四种输出模式的跟踪算法并总结各算法的特点。基于分数预测的跟踪算法需要进行较为密集的候选样本采样；基于动作决策的跟踪算法需要进行多次迭代，速度难以提升；基于矩形框回归的跟踪算法可以直接得到目标位置；基于分割预测的跟踪算法可以得到目标精确的前景描述。在本文第 §4章提出使用孪生神经网络得到跟踪目标的分割描述，该描述精确地刻画了目标在当前帧的前景状态。

1.2.6 视觉目标跟踪数据集与评测指标

视觉目标跟踪算法在不断演进的同时，越来越多围绕视觉目标跟踪任务的数据集被提出用来评测视觉目标跟踪算法。视觉目标跟踪的研究长期以来都受到 CAVIAR[1]、ALOV300[125]、MOTC[126] 以及 VideoNet[2] 等项目的持续推动。目前针对视觉目标跟踪任务常用的数据集有 Object Tracking Benchmark （简称 OTB）[1, 3] 数据集、Visual Object Tracking（简称 VOT）[2, 4, 8, 13, 77, 127, 128] 挑战赛数据集。除了这两个广泛使用的数据集外，针对于不同视觉目标跟踪中所需要考虑的特性，研究人员提出专注于无人机视角的 Unmanned Aerial Vehicle 123（简称 UAV123）[129] 数据集、针对于颜色特征跟踪的 Temple Color 128（简称 TC128）数据集、针对于行人以及刚体运动的 NUS-PRO 数据集 [130]、针对于超高速相机的 Need for Speed (简称 NFS）数据集 [131] 和针对于长时跟踪场景的 OxUvA 长时跟踪数据集 [132] 等。

近年来，基于深度学习的视觉跟踪算法 [7, 70] 快速发展，其训练集通常需要从视频检测领域的 ILSVRC 视频检测数据集 [6] 以及 YouTube-BoundingBoxes 数据集 [117] 中进行迁移。研究人员也逐步增强了针对视觉跟踪领域的大规模数据集的开发，文献 [17] 将 YouTube-BoundingBoxes 数据集中的部分数据进行重新标注，获得超过 30 万段稠密标注的视频序列。文献 [133] 提出大规模单目标跟踪数据集（Large-scale Single Object Tracking，简称 LaSOT) 来满足深度学习跟踪算法的特征训练。Huang 等 [16] 提出具有高度类别多样性的大规模通用视觉目标跟踪数据集 GOT-10k，通过显式分离训练集与验证集中的样本类别来测试网络对于目标类别的泛化性能。在这些大规模训练数据的推动下，基于深度学习的视觉目标跟踪算法成为当前视觉目标跟踪的主流方法。

本节中主要对后续章节实验中使用到的 OTB 数据集和 VOT 数据集以及其各自相应的评测指标进行详细介绍。

OTB 视觉跟踪数据集：为了消除文献报告的定量结果的不一致，便于进行公平地性能评估，Wu 等 [3] 在 2013 年收集并标注了常用的跟踪序列汇总提出 OTB-2013 数据集，该数据集由 50 段视频序列组成。随后在 2015 年，Wu 等 [1] 进一步将 OTB-2013 的视频目标数量扩展到 100 个，构成 OTB-2015 数据集。为

[1]http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1

[2]http://videonet.team

了更好地分析视觉目标跟踪算法适用的场景，Wu 等定义了 11 种视觉属性并对视频序列进行了分类标注。这些属性包括光照变化、低分辨率、尺度变化、背景干扰、目标遮挡、目标出画面、非刚性形变、平面内旋转、平面外旋转、运动模糊以及快速运动。每种属性对应视觉跟踪中的一个特定的挑战性因素。

为了定量评价跟踪算法性能，OTB 数据集提出距离精度曲线（Precision Plot）以及重叠率成功曲线（Success Plot）。在早期的目标跟踪评价指标研究中，中心点位置误差被广泛使用，通过计算跟踪器预测的中心点位置与对应帧的真实目标位置中心之间的平均欧式距离来估计算法性能。但当跟踪器丢失对目标的跟踪时，通常输出位置可能是随机分布的。因此，平均误差值无法正确地评估跟踪性能。跟踪器预测的目标中心位置与真实目标中心位置在给定阈值距离内的帧数百分比是测量跟踪性能的更好度量，OTB 数据集使用在阈值为 20 像素时的平均距离精度作为度量指标。

然而距离的度量与图像以及目标物体的尺度大小相关。OTB 数据集为了消除这一影响，采用平均重叠率指标作为主要度量指标。预测矩形框 $\mathbf{p}^t$ 与真实目标矩形框 $\mathbf{b}^t$ 之间的重叠率定义为:

$$\Phi^t = \frac{|\mathbf{p}^t \cap \mathbf{b}^t|}{|\mathbf{p}^t \cup \mathbf{b}^t|}, \tag{1.1}$$

其中 $\cap$ 和 $\cup$ 分别代表交集和并集操作，$|\cdot|$ 表示区域内面积。通过计算不同重叠率阈值下的图像帧数比例来绘制重叠率成功曲线。通过在该曲线下将不同阈值的成功率采样取平均可以计算成功图的曲线下面积（Area Under Curve，简称 AUC），该指标是 OTB 数据集进行排名的主要依据。

VOT 视觉跟踪挑战赛：在 2013 年，以建立标准化评估视觉跟踪算法性能为目标的 VOT 组委会成立[3]。VOT 的主要目标是建立标准化的数据集、评估指标和工具包，通过组织跟踪挑战赛来讨论与算法评估相关的问题。自 2013 年以来，已经成功组织了六次挑战赛，分别为 VOT-2013[127]、VOT-2014[128]、VOT-2015[4]、VOT-2016[13]、VOT-2017[8]、VOT-2018[2] 以及 VOT-2019[77]。该竞赛作为视觉目标跟踪领域最为重要的挑战赛，吸引学术界和工业界大量的研究人员参与。该挑战赛主要关注单摄像头、单目标、无特定模型、基于因果的短时跟踪。无特定模型意味着所提供的唯一训练信息是第一帧中的初始矩形框，不允

[3]http://votchallenge.net

许使用特定类别的检测器进行检测。基于因果关系要求跟踪器不使用任何未来帧或重新初始化之前的帧来推断当前帧中的目标位置。短时跟踪意味着跟踪器不能在目标丢失后进行重新检测，因此目标丢失之后会重新设置并初始化跟踪器。本文所使用的 VOT-2015、VOT-2016、VOT-2017 以及 VOT-2108 短时目标跟踪挑战赛数据都是由 60 段具有挑战性的视频构成。视频序列中的目标由旋转的矩形包围框进行标注，所有序列中的每个图像帧都由目标遮挡、光照变化、运动变化、大小变化和摄像机运动以及无属性标签进行标注。

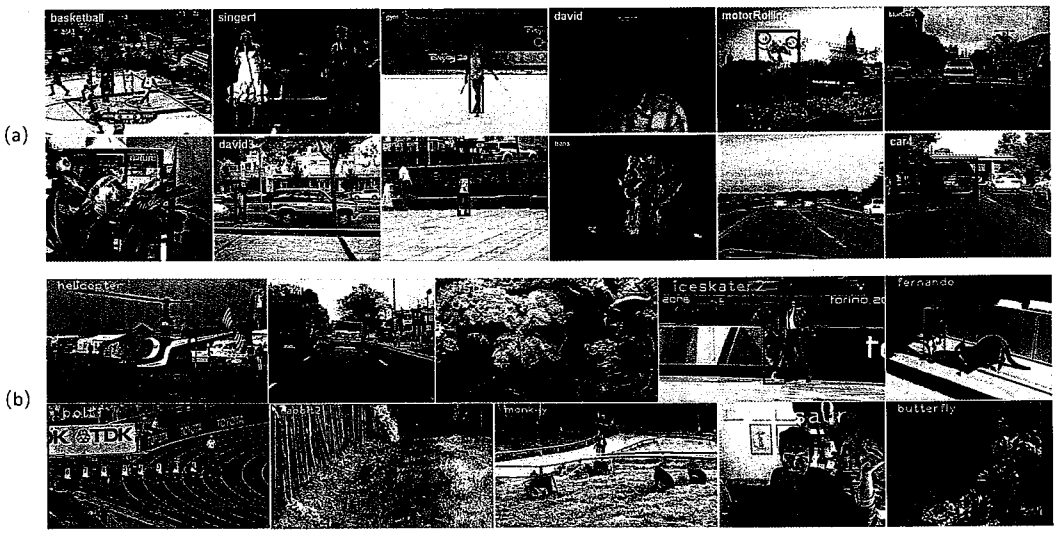
测试过程中，不同于 OTB[3] 数据集采用无复位（no-reset）实验并统计曲线下面积（AUC）进行评测，VOT 采用了一种基于重置的测试方法，在跟踪器预测结果偏离真实目标位置时重置跟踪器。文献 [4] 验证了基于复位的实验结果会降低性能估计的偏差和方差。为了增加可解释性和减少评估偏差，VOT 提出通过精度（Accuracy）和鲁棒性（Robustness）指标对跟踪算法进行评测，同时使用期望平均重叠（Expected Average Overlap，简称 EAO）将跟踪性能的两个方面评测结合起来作为主要的性能评估指标。其中精度指标主要描述跟踪算法的精确度，即跟踪预测结果与目标真实位置的平均重叠率，而鲁棒性指标主要描述算法跟踪失败重启的次数。由于本文算法在 VOT 数据集中主要采用 EAO 进行度量，我们引述 VOT-2015 中 [4] 对于 EAO 的具体定义。以一个序列长度为 T_s 帧的跟踪视频为例，跟踪算法在序列首帧进行初始化，并一直跟踪到视频结束。如果跟踪器偏离目标真实位置，则假定它将一直保持偏离，直到序列结束。跟踪器在每一帧与真实目标位置之间的重叠率记作 Φ^i，跟踪失败之后的所有图像帧的重叠率都记为 0，则跟踪器在该视频跟踪中的平均重叠率为：

$$\Phi^{T_s} = \frac{1}{T_s}\sum_{i=1}^{T_s}\Phi^i. \tag{1.2}$$

通过对许多段长度为 T_s 的视频平均重叠率进行平均，我们可以得到帧长度为 T_s 的视频的期望平均重叠率 $\hat{\Phi}^{T_s}$，类似于 OTB 数据中的平均重叠率曲线图，我们计算一段视频长度间隔 $[T_{lo}, T_{hi}]$ 内的期望平均重叠率的均值，即可稳定估计跟踪算法的期望平均重叠率：

$$\hat{\Phi} = \frac{1}{T_{hi} - T_{lo}}\sum_{T_s=T_{lo}}^{T_{hi}}\hat{\Phi}^{T_s}. \tag{1.3}$$

平均期望重叠率指标同时考虑了算法的精度以及跟踪丢失情况。此外，VOT



(a) OTB 数据集　　**(b) VOT 数据集**

图 1.5 OTB 数据集 [1] 和 VOT 数据集 [2] 的部分视频初始帧可视化示例。

数据集由于是每年组织一次的竞赛，会不断更新困难视频，减少了性能饱和的风险。该数据集在目标旋转变化，背景噪声等困难场景具有良好的指示作用。

除了上述精度指标，在 VOT-2014[128] 引入了用于评估跟踪速度的指标，称为等效滤波操作（Equivalent Filter Operations，简称 EFO）。为了减少评价指标受硬件配置的影响，工具包通过在参赛者硬件平台计算一个预定义的相关滤波操作所需的时间，对算法运行速度进行归一化来公平评价算法速度性能。在 VOT-2017[8] 中首次提出“实时跟踪”实验，模拟跟踪算法在线处理由连续运行的传感器所提供的图像。视频目标的跟踪速度受到研究人员越来越多的关注。

图1.5显示了 OTB 和 VOT 数据集的典型场景，可以看到 OTB 数据集主要围绕以行人和车辆为目标对象，侧重于考察算法对光照变化、干扰对象的适应能力。VOT 数据集使用旋转矩形来标注目标，目标物体种类更加丰富，目标的形变、旋转等变化更具挑战。

1.3　研究内容与主要贡献

基于第 §1.2节中关于视觉目标跟踪的研究现状的总结，我们可以得到如下经验总结：（1）通过滑动窗口对候选区域进行等间隔的密集采样有效枚举了所有平面空间的位移样本，同时由于规则化的采样使得后续的特征提取部分便于优化，有利于计算效率的提升。（2）手工设计特征正逐步被基于深度学习的神经网

络特征所替代，通过数据驱动的方式进行视觉目标跟踪特征提取是跟踪领域的主要趋势。直接使用针对其他任务（如图片识别或检测）预训练的神经网络级联目标跟踪表观模型的跟踪算法并不能充分利用深度学习的建模能力，端到端的神经网络学习有助于特征表示进一步适应视觉跟踪任务。（3）我们可以在跟踪算法设计过程中加入视觉跟踪的领域任务先验，并针对跟踪模型的训练样本生成、网络架构设计、关联关系构建、目标特征表示与损失函数选择等做出全面优化。（4）目标跟踪输出模式的设计通常会被研究人员所忽视，有效的输出模型可以显著提升计算效率。例如，矩形框回归的输出方式可以有效降低跟踪算法对于候选区域的稠密采样需求，并且降低算法理论估计偏差。精细化的分割输出可以降低算法表述的歧义性，减少背景干扰。

基于上述经验结论，本文围绕视觉目标跟踪算法面向现实场景的应用需求，以高效的孪生网络目标跟踪架构为研究对象，提出了多种有效的特征表示学习方法，实现了端到端的跨层级特征融合表示以及类脑注意力机制建模。同时我们将图像分割思想引入到目标跟踪对象的状态表述中，拓展了目标跟踪的表述形式，并首次构建了视觉目标跟踪与视频目标分割的一体化处理框架，建立了目标跟踪新范式。具体地，本论文的研究内容主要包含以下三个部分：

- **提出基于端到端学习的判别相关滤波器高效跟踪算法。**本文通过对判别相关滤波操作的反向传播过程进行推导，创新性地实现了深度特征表示与判别相关滤波模型的联合优化。该方法有效提升了深度特征表示的学习能力，增加了特征网络设计的自由度，同时显著降低了算法的计算存储消耗。在端到端学习过程中，通过基于尺度-位移空间的联合学习，算法引入了尺度空间样本，进而可以提供更准确的目标尺度估计。在此基础上本文又探索了基于深度特征的语义嵌入模型，并提出使用编解码自监督学习孪生网络实现对目标及其周围环境结构信息的有效感知，提升了特征表示的泛化性能与细粒度表示能力。最后本文通过分别构建具有上下文感知能力的判别相关滤波器和自监督学习语义嵌入模型实现了具有互补性的跨层级特征融合表示与学习，显著提升了跟踪性能。

- **提出基于残差注意力机制的孪生网络高效跟踪算法。**本文首先重新形式化了判别式目标跟踪算法框架，将整体跟踪网络解耦为目标特征的表示网络以及用于判别分析的判别网络。然后，本文提出了带有加权的交叉相关操作算子，可以对目标不同空间位置的相关操作赋以自适应调整的权重。模型通过联合学

习判别相关损失以及目标区域的判别系数，本文算法实现了较强的表观形态适应性。本文通过注意力机制实现判别权重表述，并提出将注意力机制分解为用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行权重调整的通道注意力机制。算法通过多种注意力机制的引入，减少了训练过程的过拟合。同时算法通过轻量化的网络设计，保证了良好的跟踪速度。

- **提出了基于孪生网络的视觉目标跟踪与分割一体化高效处理框架。**本文深入分析了当前视觉目标跟踪的输出表述形式，借鉴图像分割表述思想，首次创新性地提出针对于目标跟踪的多任务输出表示方法。本文通过引入独立的分割分支到全卷积孪生网络框架，使得孪生网络架构可以同时估计目标的矩形框位置以及输出精细的目标分割表述。对于分割分支的架构设计，算法采用向量化的分割表述方式获取目标全局信息，并提出自顶向下的堆叠精细化模块来增强分割细节。该框架的离线训练过程可通过多分支任务联合学习进行优化。在线跟踪过程中，算法只需要输入初始帧标注的矩形框初始化，即可同时完成视觉目标跟踪任务与视频目标分割任务。整个框架在完成多个任务的基础上，具有较高的分割效率，运行速度达到接近55fps。最后，本文将上述一体化处理框架扩展到多目标跟踪场景，实现了无输入标签监督的多目标视频实例分割。

1.4 论文组织结构

本文主要针对基于孪生网络的实时视觉目标跟踪算法进行深入研究，论文整体结构组织如下：

第一章为绪论，首先从视觉目标跟踪的角度阐述本文的研究背景与意义，然后深入讨论了视觉目标跟踪算法的理论架构，并结合跟踪评测数据集的发展全面介绍了视觉目标跟踪的研究现状，最后阐明了本文对于孪生网络跟踪算法的创新改进方向，并对相关研究内容进行了简单概述。

第二章提出基于端到端学习的判别相关滤波器高效跟踪算法。首先，我们分析了当前相关滤波跟踪算法的特征表示对跟踪任务感知不足，提出通过对判别相关滤波操作的反向传播过程进行推导，将特征提取与判别模型学习相统一，并对整体架构进行端到端训练，从而有效提升深度学习特征的判别能力，降低算法的计算存储消耗。然后，为了提升算法对于尺度估计的准确性，我们提出尺

度-位移空间联合学习的相关滤波器，并增加多尺度样本进行训练。然后我们又探索了基于深度特征的语义嵌入模型，提出了结构感知的自监督学习模块，并使用编解码孪生网络进行自监督学习，改善了模型的泛化性能与细粒度表示能力。最后，我们利用两个互补的不同层级特征联合学习上下文感知的判别相关滤波器及语义嵌入模型，增加了跟踪器的可靠性。我们在多个视觉目标跟踪数据集中验证了上述端到端学习算法的有效性。

第三章提出基于残差注意力机制的孪生网络高效跟踪算法。我们首先重新形式化了判别式目标跟踪算法框架，将整体跟踪网络解耦为目标特征的表示网络以及用于判别分析的判别网络。然后，我们分析了通用的全卷积孪生网络对于目标形态适应性的不足，提出了带有加权的交叉相关操作算子用来对目标不同空间位置的相关操作赋以自适应调整的权重。我们提出通过注意力机制实现判别权重表述，并将注意力机制分解为用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行调整的通道注意力机制。最后，我们分析了上述三种注意力机制对于训练过程的影响，并在视觉跟踪标准数据集上验证了注意力机制算法的有效性以及局限性。

第四章提出孪生网络视觉目标跟踪与分割的一体化高效处理框架。我们首先深入分析了当前视觉目标跟踪的输出表述形式，借鉴图像分割表述思想，提出针对于目标跟踪的精确输出表示方法。然后，我们提出通过引入独立的分割分支到全卷积孪生网络框架，构建孪生分割网络架构来同时预测目标的矩形框位置以及精细的目标分割表述。接着我们介绍了分割分支的架构设计，提出采用向量化的分割表述方式获取目标全局信息，并设计了自顶向下的堆叠精细化模块来增强分割细节。我们对本章提出的孪生分割网络进行了评测，通过与现有的视觉目标跟踪算法以及视频目标分割算法进行对比，证明了所提出的分割表述对于目标跟踪精度的提升作用。最后，本文将上述一体化跟踪框架扩展到多目标跟踪场景，实现了无输入标签监督的多目标视频实例分割。

最后，我们对本文的研究工作与贡献进行总结，并对未来的研究工作进行了展望。

第 2 章　基于端到端学习的判别相关滤波器高效跟踪算法研究

2.1　引言

对于视觉目标跟踪算法而言，特征表示提取和判别表观学习两者都具有极其重要的地位。在缺少目标类别先验的情况下，跟踪任意类别目标物体通常需要在线进行目标前景与背景的判别信息学习。同时由于目标大小、形状、视角以及背景场景中光照、遮挡物、嘈杂干扰等都会随时间发生剧烈变化，这为在线判别学习带来了极大挑战。鲁棒的特征表示可以有效降低判别器的学习难度，显著提升目标跟踪的质量。然而，由于目标表观的形态分布具有高度的复杂性，手工设计的特征表示方法难以涵盖物体表观的视觉特征变化。因而基于深度学习的特征表示研究受到越来越多的重视，而如何高效地将深度特征表示与在线判别算法结合是处理这些挑战的重要手段。本章主要通过对深度特征表示提取与判别相关滤波模型的联合优化，提升深度特征表示的学习能力。

近年来，卷积神经网络在许多具有挑战性的视觉任务上 [39, 63, 134] 取得了突破性进展，这极大地激发了深度学习在视觉目标跟踪领域的应用与普及。近期基于深度学习的目标跟踪算法主要利用预训练好的深度卷积神经网络特征 [12, 30, 67]，或者设计用于判别性视觉目标跟踪学习的深度架构 [25, 65, 66, 103, 135]。研究工作 [25, 65, 66] 遵循离线训练（offline training）和在线微调（online fine-tuning）的优化策略。虽然卷积神经网络特征通常可以有效地区分不同类别的物体表观，但从每一帧图像中提取深度学习特征并使用神经网络进行训练和更新的跟踪算法需要消耗大量的计算资源。在线微调神经网络以适应特定目标表观的跟踪策略严重影响了算法的速度 [9, 25, 66]。

基于孪生神经网络的跟踪器 [7, 35] 在速度方面显示了强大的应用潜力。全卷积孪生网络跟踪器 SiamFC[7] 将视觉目标跟踪形式化为识别验证问题，构建基于深度特征模板匹配的跟踪算法，通过度量候选图像和模板图像的相似性进行目标识别。该算法的度量学习主要依赖于离线训练，而不进行在线更新，降低了模型更新带来的计算负担。同时，孪生网络算法通过全卷积的网络结构对搜索区域进行均匀采样，共享不同候选图像块的特征计算，实现了超实时的跟踪速度。在网络离线训练阶段，孪生网络方法利用 AlexNet[39] 或 VGGNet[59] 等卷

积神经网络架构作为主干（backbone）网络，在大规模视频检测数据集 ILSVRC VID[6] 上学习用于分类 [7, 10] 或回归 [35] 的语义嵌入空间。在离线训练后的语义嵌入空间中，目标特征表示包含高层语义信息，能够有效区分不同类别的目标对象。同时在大规模的训练集上的训练使算法具有一定的跨数据集的泛化功能，这确保了健壮的跟踪实现。在在线跟踪阶段，孪生网络跟踪器只通过一次网络前向传播来估计目标位置，而不需要进行任何在线网络微调。这些特性是基于深度特征表示对训练数据中不同实例物体进行判别学习的优势。然而在比对具有相同语义属性的两个目标对象时，该类算法通常对细节差异不太敏感。这种缺乏对细粒度（fine-grained）差异的识别能力给基于孪生网络的跟踪器带来以下限制:

- 语义嵌入空间中的特征表示通常具有较低的分辨率，并且会丢失实例目标的部分细节以及细粒度局部信息。当跟踪器遇到来自未知类别的目标或发生突然变形时，可能会出现领域移位问题（domain shift）[25]。
- 孪生网络跟踪算法通常不执行在线网络更新以提高跟踪速度。这不可避免地影响了模型的适应性，从而降低了跟踪精度。基于孪生网络的跟踪器 [7] 由于同时执行特征提取和相关分析取得了极快的跟踪速度。然而，由于算法没有进行在线学习，而仅使用从相关分析中得到的固定度量，因而视频序列时序信息没有得到充分利用，跟踪的时序适应性受到损害。

我们认为基于孪生网络的跟踪器需要学习细粒度的特征，同时高分辨率轻量化的网络特征更适合于准确的跟踪定位。在离线训练中，算法应更多利用数据的结构信息，从而使学习的语义嵌入表示更加健壮，以避免领域移位。

除了对于跟踪精度的提升需求，在实际应用中，维护复杂场景中视觉目标跟踪算法的实时处理能力也是至关重要的。近年来，基于相关滤波的跟踪算法 [12, 29, 89] 因其卓越的跟踪精度和极高的运算速度而备受关注，成为 2011 年至 2018 年间最为主流的跟踪算法这类跟踪算法对目标图像块和目标周边全部邻近图像块的相关性进行建模，通过在傅里叶频域快速地求解脊回归问题实现目标跟踪与学习。基于相关滤波器的跟踪算法通常由特征提取器和相关滤波器两个核心部分组成。Bolme 等人 [21] 首次将相关滤波分析引入视觉跟踪中，算法在简单的灰度特征上运行速度达到 669fps。Henriques 等 [29, 51] 使用循环矩阵来解释相关滤波器工作原理，并推广到多通道 HOG 特征表示进行核（kernel）学习。Danelljan 等人在文献 [44] 中加入了颜色名称（color name）[40] 特征，以提

高相关滤波器跟踪的颜色属性适应性。文献 [38] 通过实验验证具有高度判别性的特征表示可以极大地提升跟踪精度。随后，相关滤波算法的特征表示逐步从手工设计的特征（如原始灰度特征 [21]、方向梯度直方图特征 [29] 和颜色名称 [44]）发展到使用在 ImageNet[6] 中预先训练的多层深度特征 [12, 30]，产生了卷积神经网络和相关滤波器组合的跟踪算法。Danelljan 等人 [67] 分析深度卷积特征不同网络层的跟踪性能，得出第一层特征与空间正则化相关滤波器结合可以取得最好性能。研究人员 [12, 30, 67, 69] 在后续工作中专注于使用基于深度学习的多层特征表示来实现相关滤波跟踪。Ma 等人 [30] 对多个卷积层分别学习相关滤波器进行跟踪。但是，上述相关滤波算法所采用的预训练网络参数主要针对于图像识别任务训练，其特征表示忽视同类个体之间差异且具较强的平移不变性，不利于精确的目标跟踪定位。同时，预训练的深度特征提取网络结构通常具有分辨率低、网络层数深以及输出维度高的结构特点，显著影响了跟踪器的实时运行效率。Valmadre 等人 [11] 提出使用从端到端学习为视觉目标跟踪定制的深度特征，在不损失运行速度的前提下提高跟踪精度。然而，其网络特征具有较低的分辨率以及极高维度的输出，这些结构并不利于精确的目标跟踪实现。

此外，研究人员 [66, 100] 通过同时运行多个相关滤波跟踪器进行协同（ensemble）跟踪，单个跟踪器的错误跟踪可以由其他类别参数的跟踪器进行补偿。Hong 等人 [99] 和 Ma 等人 [52] 同期提出在跟踪中增加重检测机制，实现了长时相关滤波跟踪。Fan 和 Ling[100] 提出并行运行基于相关滤波器的快速跟踪算法和长期保守的目标验证模块。Bertinetto 等人 [75] 将基于颜色直方图统计的模型应用于相关滤波器跟踪，实现颜色特征与边缘特征的互补。近年来，基于相关滤波的跟踪方法在多通道特征 [29, 30]、尺度估计 [32, 89]、减弱边界效应 [53–55] 等方面取得了很大的进展。然而，基于相关滤波器的跟踪算法通常有以下限制：

- 算法使用手工设计的特征提取器或者采用针对图像分类 [39, 59, 61] 或检测 [62] 等其他任务而训练的参数复杂的网络进行特征提取。这种分离的设计方式使得特征提取器与相关滤波器之间通常缺乏有效的衔接。特征提取与目标跟踪的表观建模不能很好的协同适应，从而无法实现最优的跟踪模型设计。
- 特征提取网络具有较深的网络层数，算法结合多个特征层来学习同类的表观模型。日益复杂的模型设计与大规模训练参数不可避免地引入了严重的过拟合风险。同时，精度的提升通常伴随着数倍的速度下降损失，在跟踪精度和速

度之间往往缺乏良好的平衡。

- 目标尺度变换通常不在表观建模过程中予以考虑，这使得跟踪器很难利用目标位置和尺度之间的相关性。例如，文献 [89] 中级联使用的二维空间位置滤波器和一维尺度滤波器来处理物体位置和尺度的变化。

围绕上述两大类算法存在的主要问题，本章提出了基于端到端学习的判别相关滤波网络（End-to-End Discriminant Correlation Filters Network，简称 DCFNet）用于高效视觉目标跟踪，初步改善了目前已有相关算法存在的上述问题。我们的方法具有如下两个特点：1）可以通过视觉跟踪数据驱动特征表示学习，使得特征提取模块适配判别模块以及跟踪任务。通过对判别相关滤波器的反向梯度传播过程进行推导，实现了神经网络特征与判别模块的联合优化。离线训练过程中，算法通过对不同光照、视角、形变样本进行采样，学习具有表观变化鲁棒性的特征表示。算法通过训练得到的特征提取部件同时受到训练数据以及判别模块的影响，提升了深度特征对于跟踪任务针对性。2）可以根据使用场景，调整表观特征网络结构。传统的相关滤波算法通常使用固定的预训练特征网络，无法根据设备场景进行有效的更改。由于端到端学习的使用，算法可以较为自由地调整特征提取网络的拓扑结构，通过减少网络的深度与通道数有效提升跟踪速度。

具体来说，本章提出对判别相关滤波操作的反向传播过程进行推导，创新性地实现了深度特征自动提取与相关滤波判别模型的联合优化。该方法有效提升了深度特征表示的学习能力，增加了特征表示网络设计的自由度，同时显著降低了算法的计算存储消耗。在端到端学习过程中，通过基于尺度-位移空间的联合学习，算法引入了尺度空间样本，进而可以提供更准确的目标尺度估计。在此基础上本文又探索了基于深度特征的语义嵌入模型，并提出使用编解码自监督学习孪生网络实现对目标及其周围环境结构信息的有效感知，提升了特征表示的泛化性能与细粒度表示能力。通过分别构建具有上下文感知能力的判别相关滤波器和自监督学习语义嵌入模型实现了具有互补性的跨层级特征融合表示与学习，显著提升了跟踪性能。最后，我们在数据集 OTB-2013[3]、OTB-2015[1] 以及 VOT-2015 中进行了大量的实验对比，本章所提出的端到端训练的特征表示方法相对于传统的特征表示方法取得了显著的精度提升。

我们将评测结果和统一框架的代码分享在项目网站，可以参见https://github.com/foolwood/DCFNet。

2.2　判别相关滤波跟踪器框架

判别相关滤波方法目前在视觉目标跟踪领域占据主导地位，但是其特征提取部分通常采用手工设计特征（例如 HOG 特征、颜色名称特征）或在 ImageNet 上训练的卷积神经网络特征。本章提出设计轻量化网络架构来端到端学习适应于相关滤波的卷积神经网络。我们将判别相关滤波操作转化为神经网络中特殊的网络层，通过推导其前后向传播过程加入到孪生神经网络的架构学习当中，通过最小化相关滤波器的网络输出与预定义的期望输出的差异来优化神经网络参数。由于前后向推导都是在频域中实现，保留了原有判别相关滤波跟踪算法的速度优势，这使得本章提出的跟踪算法可以在测试过程中达到 60fps。同时，由于使用针对跟踪问题优化的神经网络特征，使得算法的精度显著提升。本小节主要介绍判别相关滤波器的基础方法与前向推导过程。

在传统的判别相关滤波跟踪框架中，算法对目标样本的特征集合和对应的目标响应进行判别回归训练。我们通过 $\varphi^l(\mathbf{x}) \in \mathbb{R}^{M\times N}$ 来表示目标图像块 $\mathbf{x}$ 的特征表示的第 l 个通道，特征表示总共包含 D 个通道，目标图像块的完整特征表示记为 $\{\varphi^l(\mathbf{x})\}_{l=1}^{D}$。算法通过在目标样本周围密集采样来学习相关滤波器 $\mathbf{w}$，这些密集采样的样本通过在向量化的目标样本特征上进行循环移位来构建。具体地，通过对特征向量 $\{\varphi^l(\mathbf{x})\}_{l=1}^{D}$ 进行循环移位，我们将这些循环移位的样本叠加构成样本矩阵 $\boldsymbol{\Phi}(\mathbf{x})$。样本矩阵 $\boldsymbol{\Phi}^l(\mathbf{x})$ 的每一行包含图像特征 $\varphi(\mathbf{x})$ 进行特定循环变换的第 l 通道特征，理想的相关响应 $\mathbf{y} \in \mathbb{R}^{M\times N}$ 确保目标在图像中心位置获得最高得分，呈现单峰高斯分布，在像素 (u, v) 处通常定义如下：

$$\mathbf{y}_{u,v} = e^{-\frac{\left(u-\lfloor\frac{M}{2}\rfloor\right)^2+\left(v-\lfloor\frac{N}{2}\rfloor\right)^2}{\sigma^2}}, \tag{2.1}$$

其中 M 和 N 是响应图 $\mathbf{y}$ 的边长，σ 表示响应带宽，坐标点 $(\lfloor M/2\rfloor, \lfloor N/2\rfloor)$ 是响应图的中心。我们假定 $\mathbf{w}^l$ 为滤波器 $\mathbf{w}$ 的第 l 个通道，通过最小化脊回归损失来求解滤波器 $\mathbf{w}$：

$$\begin{aligned} \mathbf{w} &= \underset{\mathbf{w}}{\operatorname{argmin}} \left\| \sum_{l=1}^{D} \boldsymbol{\Phi}^l(\mathbf{x})\mathbf{w}^l - \mathbf{y} \right\|_2^2 + \lambda \sum_{l=1}^{D} \left\|\mathbf{w}^l\right\|_2^2 \\ &= \underset{\mathbf{w}}{\operatorname{argmin}} \left\| \sum_{l=1}^{D} \mathbf{w}^l \star \varphi^l(\mathbf{x}) - \mathbf{y} \right\|_2^2 + \lambda \sum_{l=1}^{D} \left\|\mathbf{w}^l\right\|_2^2 \end{aligned}, \tag{2.2}$$

其中 $\star$ 代表循环相关操作，λ 是正则项权重系数。循环样本特征矩 $\boldsymbol{\Phi}^l(\mathbf{x})$ 的设计结构确保 $\boldsymbol{\Phi}^l(\mathbf{x})\mathbf{w}^l$ 等同于 $\varphi^l(\mathbf{x})$ 与 $\mathbf{w}^l$ 进行循环相关操作 [29]。同时，时域的卷积

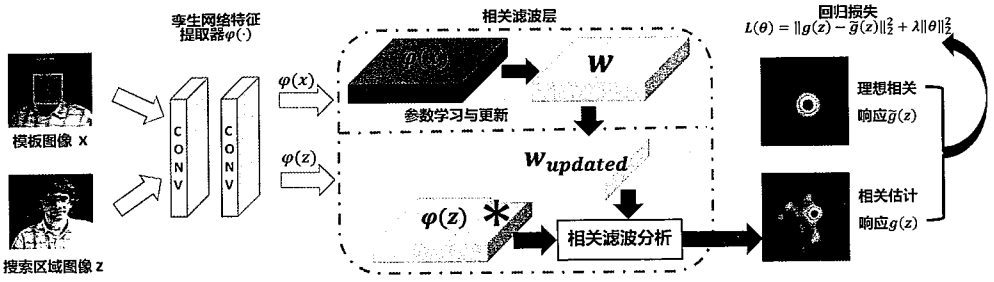


图 2.1 基于端到端学习的判别相关滤波网络 (DCFNet) 结构示意图：模板图像 x 通过孪生网络特征提取模块得到特征表示 $\varphi(\mathbf{x})$，通过公式（2.3）求解得到相应的滤波器 w；搜索区域图像 z 通过同样的特征提取器得到特征表示 $\varphi(\mathbf{z})$；通过公式（2.4）求解后得到相关估计响应，最后与理想相关响应进行回归训练。

操作可以转换为傅里叶频域的哈达玛积操作（Hadamard Product）[21]，公式 (2.2) 的脊回归问题可以在傅里叶域高效求解，算法求解复杂度可由 $\mathcal{O}\left(A^3+DA^2\right)$ 降至 $\mathcal{O}(DA\log A)$[95]，其中 $A=M\cdot N$。公式 (2.2) 的解可以表示为：

$$\hat{\mathbf{w}}^l=\frac{\hat{\varphi}^l(\mathbf{x})\odot\hat{\mathbf{y}}^*}{\sum_{k=1}^{D}\hat{\varphi}^k(\mathbf{x})\odot\left(\hat{\varphi}^k(\mathbf{x})\right)^*+\lambda\mathbb{1}},\tag{2.3}$$

其中符号 $\hat{\cdot}$ 表示将实数向量进行离散傅里叶变换 $\mathcal{F}(\cdot)$，$\cdot^*$ 表示复数共轭，$\odot$ 表示哈达玛积操作，$\mathbb{1}$ 表示大小为 $M\times N$ 的全 1 的向量，公式中的除法为上下两个向量对应项的商。

在后续帧中，给定搜索区域 z 并提取特征表示 $\left\{\varphi^l(\mathbf{z})\right\}_{l=1}^{D}$，该特征与滤波器模板 w 的相关响应图 $g(\mathbf{z})$ 通过下式来计算：

$$g(\mathbf{z})=\mathcal{F}^{-1}\left(\sum_{l=1}^{D}\hat{\mathbf{w}}^{l*}\odot\hat{\varphi}^l(\mathbf{z})\right),\tag{2.4}$$

其中，$\mathcal{F}^{-1}$ 为离散傅里叶变换的逆变换。算法通过在响应图 $g(\mathbf{z})$ 中找到最大响应所对应的位置即为当前帧的跟踪位置结果。上述滤波器求解以及搜索区域响应计算过程是通用相关滤波跟踪算法进行目标跟踪的标准流程。

2.3 基于端到端学习的判别相关滤波网络跟踪框架

本章提出了判别相关滤波网络 (DCFNet)，该网络由两个卷积层组成的共享特征表示模块和一个用于视觉跟踪的相关滤波层组成，其网络结构如图 2.1所示。模型通过将特征提取器与判别相关模块级联得到目标位置的响应估计，以实现

跟踪预测。与常规的相关滤波跟踪算法不同，本章算法通过反向传播相关滤波响应的回归损失，调整特征提取器和目标对象的表观模型参数。传统的基于判别相关滤波的跟踪器只启发式地调整相关滤波的超参数（例如，学习率、正则系数权重等），而特征提取模块并未参与跟踪任务的先验学习。借鉴于神经网络的反向传播训练过程 [136]，本章所提出的判别相关滤波网络同时调整判别相关滤波器的超参数以及特征表示网络的模型参数。在离线训练阶段，利用大规模视频目标检测数据集 ILSVRC VID[6] 对 DCFNet 进行端到端的训练，有效提升了深度特征表示的学习能力。在在线跟踪阶段，为了节省运行时间、避免过拟合以及减少跟踪漂移，算法固定卷积层特征参数。同时算法对相关滤波层进行不断更新，以适应跟踪目标表观的变化。跟踪过程根据相关响应的最大值的位置同时估计目标的中心位置和尺度。本章提出的 DCFNet 模型的关键贡献是对相关滤波器的反向传播以及在线模型更新的推导。

2.3.1　判别相关滤波反向传播推导

如图2.1所示，判别相关滤波网络由以模板样本为输入的滤波器学习分支和以搜索图像为输入的跟踪分支组成。滤波器学习分支利用目标样本特征在相关滤波层学习判别滤波器，目标跟踪分支在候选搜索区域上计算相关响应值。相关滤波器模型期望目标在搜索图像的正确位置产生峰值响应。然而由于搜索图像相对于目标模板图像发生了表观变化，而当特征表示网络没有建模这种变化，会使得算法估计的真实响应峰值位置产生偏移，甚至错误地定位在其他物体处。常规的相关滤波算法由于只采用前向学习过程，无法针对这种错误估计进行修正。

本章提出通过最小化搜索目标的真实响应 $g(\mathbf{z})$ 和理想的二维高斯响应 $\mathbf{y}$ 的差异来训练特征表示网络，增加特征提取器对于物体表观变化的鲁棒性以及对于干扰对象的判别力。在训练过程中，判别相关滤波网络输入图像样本对：目标图像 $\mathbf{x}$ 和搜索区域图像 $\mathbf{z}$。采用 $\boldsymbol{\theta}$ 来表示当前特征提取网络的权重参数。通过该特征提取器提取目标模板图像块 $\mathbf{x}$ 的特征 $\left\{\varphi_{\theta}^{l}(\mathbf{x})\right\}_{l=1}^{D}$。将目标模板特征 $\left\{\varphi_{\theta}^{l}(\mathbf{x})\right\}_{l=1}^{D}$ 代入公式（2.3）获得判别滤波器 $\left\{\mathbf{w}^{l}\right\}_{l=1}^{D}$。然后算法通过特征提取器获得搜索区域图像 $\mathbf{z}$ 的特征表示 $\left\{\varphi_{\theta}^{l}(\mathbf{z})\right\}_{l=1}^{D}$，搜索区域 $\mathbf{z}$ 对应的相关响应 $g_{\theta}(\mathbf{z}|\mathbf{x})$ 通过公式（2.4）来计算获得。算法通过最小化预测目标响应与理想目标响应的差异来优化网络参数：

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \left(\left\| g_{\theta}(\mathbf{z}) - \mathbf{y} \right\|_2^2 + \gamma \|\theta\|_2^2 \right). \tag{2.5}$$

我们通过上述损失函数来优化特征提取参数 θ，此外显式增加的正则项 $\|\theta\|_2^2$ 有助于模型更快的收敛，并采用神经网络参数优化方法权值衰减（weight decay）来实现这种正则的优化。

为了使得特征提取和相关滤波分析连接并进行协同训练，我们完整推导了优化神经网络参数 θ 的反向传播过程。我们将 $g_\theta(\mathbf{z})$ 看作 $\varphi^l(\mathbf{z})$ 和 $\varphi^l(\mathbf{x})$ 的函数。我们同时假定各通道特征独立，通过链式法则 [136] 对网络的两个分支分别回传误差，可以得到：

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta}=\sum_{l=1}^{D}\frac{\partial \mathcal{L}(\theta)}{\partial g_\theta(\mathbf{z})}\frac{\partial g_\theta(\mathbf{z})}{\partial \varphi^l(\mathbf{z})}\frac{\partial \varphi^l(\mathbf{z})}{\partial \theta}+\sum_{l=1}^{D}\frac{\partial \mathcal{L}(\theta)}{\partial g_\theta(\mathbf{z})}\frac{\partial g_\theta(\mathbf{z})}{\partial \varphi^l(\mathbf{x})}\frac{\partial \varphi^l(\mathbf{x})}{\partial \theta}+2\gamma\theta, \tag{2.6}$$

根据公式 (2.6)，我们需要依次推导 $\frac{\partial \mathcal{L}(\theta)}{\partial g_\theta(\mathbf{z})}$、$\frac{\partial \mathcal{L}}{\partial(\varphi^l(\mathbf{z}))}$ 以及 $\frac{\partial \mathcal{L}}{\partial(\varphi^l(\mathbf{x}))}$。下面我们从 $\frac{\partial \mathcal{L}(\theta)}{\partial g_\theta(\mathbf{z})}$ 开始推导，并将损失函数 $\mathcal{L}(\theta)$ 简化为 $\mathcal{L}$，响应函数 $g_\theta(\mathbf{z})$ 简化为 $\mathbf{g}$。通过公式（2.5）中的损失函数以及搜索区域响应计算公式（2.4），我们可以得出：

$$\frac{\partial \mathcal{L}}{\partial \mathbf{g}}=2\left(g(\mathbf{z})-\mathbf{y}\right)=2\left(\mathcal{F}^{-1}\left(\sum_{l=1}^{D}\hat{\mathbf{w}}^{l^*}\odot\hat{\varphi}^l(\mathbf{z})\right)-\mathbf{y}\right). \tag{2.7}$$

而上式中相关响应计算需要经过傅里叶变换操作，对于中间变量需要考虑其复数值（complex）的特性。而针对于相关滤波器中的傅里叶变换的反向传播推导是本章的核心贡献之一。下面我们首先形式化定义离散傅里叶变换和离散傅里叶变换的逆变换，假定向量 $\mathbf{g}$ 为元素个数为 N 的向量，那么我们可以得到：

$$\hat{\mathbf{g}}_f=\frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\mathbf{g}_n e^{-j\frac{2\pi}{N}nf}=\underset{n\to f}{\mathcal{F}}\left(\mathbf{g}_n\right), \tag{2.8}$$

$$\mathbf{g}_n=\frac{1}{\sqrt{N}}\sum_{f=0}^{N-1}\hat{\mathbf{g}}_f e^{j\frac{2\pi}{N}nf}=\underset{f\to n}{\mathcal{F}^{-1}}\left(\hat{\mathbf{g}}_f\right), \tag{2.9}$$

其中 $\mathbf{g}_n$ 是离散时域信号，$\hat{\mathbf{g}}_f$ 是其转换到傅里叶频域的结果，f 的范围为 $\{0,\ldots,N-1\}$。$\hat{g}_f$ 关于 $\mathbf{g}_n$ 和 $\mathbf{g}_n^*$ 之间的偏导数关系为：

$$\begin{cases}\frac{\partial \hat{\mathbf{g}}_f}{\partial \mathbf{g}_n}=\frac{1}{\sqrt{N}}e^{-j\frac{2\pi}{N}nf}\\ \frac{\partial \hat{\mathbf{g}}_f}{\partial \mathbf{g}_n^*}=0\end{cases}. \tag{2.10}$$

$\mathbf{g}_n$ 关于 $\hat{\mathbf{g}}_f$ 和 $\hat{\mathbf{g}}_f^*$ 之间的偏导数关系为：

$$\begin{cases}\frac{\partial \mathbf{g}_n}{\partial \hat{\mathbf{g}}_f}=\frac{1}{\sqrt{N}}e^{j\frac{2\pi}{N}nf}\\ \frac{\partial \mathbf{g}_n}{\partial \hat{\mathbf{g}}_f^*}=0\end{cases}. \tag{2.11}$$

因而我们可以得到如下推论:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{g}_n^*} &= \sum_{f=0}^{N-1}\left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_f^*}\right)^* 0+\frac{1}{\sqrt{N}}\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_f^*}\left(e^{-j\frac{2\pi}{N}nf}\right)^*\right) \\ &= \frac{1}{\sqrt{N}}\sum_{f=0}^{N-1}\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_f^*}e^{j\frac{2\pi}{N}nf}=\mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_f^*}\right)\end{aligned}, \tag{2.12}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_f^*} &= \sum_{n=0}^{N-1}\left(\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_n^*}\right)^* 0+\frac{1}{\sqrt{N}}\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_n^*}\left(e^{j\frac{2\pi}{N}nf}\right)^*\right) \\ &= \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1}\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_n^*}e^{-j\frac{2\pi}{N}nf}=\mathcal{F}\left(\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_n^*}\right)\end{aligned}. \tag{2.13}$$

因为 $\mathbf{g}$ 是我们所预测的相关响应，其所有元素均是实数，也即 $\mathbf{g}\in\mathbb{R}^N$。所以我们可以得出 $\mathbf{g}=\mathbf{g}^*$，其离散傅里叶变换和离散傅里叶变换的逆变换间导数关系表述为:

$$\begin{cases}\hat{\mathbf{g}}=\mathcal{F}(\mathbf{g}) \\ \dfrac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}^*}=\mathcal{F}\left(\dfrac{\partial \mathcal{L}}{\partial \mathbf{g}^*}\right)=\mathcal{F}\left(\dfrac{\partial \mathcal{L}}{\partial \mathbf{g}}\right) \\ \dfrac{\partial \mathcal{L}}{\partial \mathbf{g}}=\dfrac{\partial \mathcal{L}}{\partial \mathbf{g}^*}=\mathcal{F}^{-1}\left(\dfrac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}^*}\right)\end{cases}. \tag{2.14}$$

下面我们将公式（2.6）进行展开:

$$\begin{aligned}\frac{\partial \mathcal{L}(\theta)}{\partial \theta} &= \sum_{l=1}^{D}\frac{\partial \mathcal{L}(\theta)}{\partial \varphi^l(\mathbf{z})}\frac{\partial \varphi^l(\mathbf{z})}{\partial \theta}+\sum_{l=1}^{D}\frac{\partial \mathcal{L}(\theta)}{\partial \varphi^l(\mathbf{x})}\frac{\partial \varphi^l(\mathbf{x})}{\partial \theta}+2\gamma\theta \\ &= \sum_{l=1}^{D}\mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}(\theta)}{\partial\left(\hat{\varphi}^l(\mathbf{z})\right)^*}\right)\frac{\partial \varphi^l(\mathbf{z})}{\partial \theta} \\ &\quad+\sum_{l=1}^{D}\mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}(\theta)}{\partial\left(\hat{\varphi}^l(\mathbf{x})\right)^*}\right)\frac{\partial \varphi^l(\mathbf{x})}{\partial \theta}+2\gamma\theta\end{aligned}. \tag{2.15}$$

因为网络的前向运算过程中主要包含了基于元素的 Hadamard 乘除运算，我们可以按照像素空间位置在频域进行推导:

$$\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_{uv}^*}=\left(\mathcal{F}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{g}}\right)\right)_{uv}. \tag{2.16}$$

对于跟踪分支的反向传播过程，公式（2.15）中部分偏导数 $\partial\mathcal{L}/\partial\left(\varphi^l(\mathbf{z})\right)$ 由公式（2.4）可知:

$$\frac{\partial \hat{g}_{uv}^*(\mathbf{z})}{\partial\left(\hat{\varphi}_{uv}^l(\mathbf{z})\right)^*}=\hat{\mathbf{w}}_{uv}^l, \tag{2.17}$$

因而跟踪分支的反向传播可以推导:

$$\frac{\partial \mathcal{L}}{\partial\left(\hat{\varphi}_{uv}^l(\mathbf{z})\right)^*}=\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_{uv}^*}\frac{\partial \hat{\mathbf{g}}_{uv}^*(\mathbf{z})}{\partial\left(\hat{\varphi}_{uv}^l(\mathbf{z})\right)^*}=\frac{\partial \mathcal{L}}{\partial \hat{\mathbf{g}}_{uv}^*}\hat{\mathbf{w}}_{uv}^l. \tag{2.18}$$

该分支完整的反向传播关系表述为:

$$\frac{\partial \mathcal{L}}{\partial \varphi^l(\mathbf{z})}=\mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}}{\partial\left(\hat{\varphi}^l(\mathbf{z})\right)^*}\right)=\mathcal{F}^{-1}\left(\mathcal{F}\left(2(g(\mathbf{z})-\mathbf{y})\right)\odot\hat{\mathbf{w}}^l\right). \tag{2.19}$$

对于滤波器学习分支的反向传播，算法需要求解部分偏导数 $\partial\mathcal{L}/\partial\left(\varphi^{l}(\mathbf{x})\right)$。我们通过分别推导 $\partial\mathcal{L}/\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})$ 和 $\partial\mathcal{L}/\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}$，然后合并得到最终的偏导数 $\partial\mathcal{L}/\partial\left(\varphi^{l}(\mathbf{x})\right)$：

$$\begin{cases}\dfrac{\partial\mathcal{L}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\dfrac{\partial\mathcal{L}}{\partial\hat{\mathbf{g}}_{uv}^{*}}\dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}\\ \dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\mathbf{w}}_{uv}^{l}}\dfrac{\partial\hat{\mathbf{w}}_{uv}^{l}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\dfrac{\partial\hat{\mathbf{w}}_{uv}^{l}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}\end{cases},\tag{2.20}$$

其中根据公式（2.4）可得：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\mathbf{w}}_{uv}^{l}}=\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}.\tag{2.21}$$

下面，我们使用 μ 来代表公式（2.3）中滤波器 $\mathbf{w}$ 的分母：

$$\mu=\sum_{k=1}^{D}\hat{\varphi}_{uv}^{k}(\mathbf{x})\left(\hat{\varphi}_{uv}^{k}(\mathbf{x})\right)^{*}+\lambda.\tag{2.22}$$

根据公式（2.3）中滤波器 $\mathbf{w}$ 学习过程，我们可得：

$$\frac{\partial\hat{\mathbf{w}}_{uv}^{l}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\frac{\hat{\mathbf{y}}_{uv}^{*}\mu-\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}}{\mu^{2}},\tag{2.23}$$

并将上式代入公式（2.20），得到：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\frac{\hat{\mathbf{y}}_{uv}^{*}\mu\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}-\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}}{\mu^{2}}.\tag{2.24}$$

根据公式（2.3），我们可以得到：

$$\mu=\frac{\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}}{\hat{\mathbf{w}}_{uv}^{l}},\tag{2.25}$$

以及 $\mu\hat{\mathbf{w}}_{uv}^{l}=\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}$，我们将上式代入公式（2.24）得到：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\frac{\hat{\mathbf{y}}_{uv}^{*}\mu\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}-\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\mu\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}},\tag{2.26}$$

然后使用 $\mu\hat{\mathbf{W}}_{uv}^{l}$ 代入公式（2.26）得到：

$$\begin{aligned}\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}&=\frac{\hat{\mathbf{y}}_{uv}^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}-\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\mu\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}}\\&=\frac{\hat{\mathbf{y}}_{uv}^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}-\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\mu}\end{aligned},\tag{2.27}$$

其中分母中的 $\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\mathbf{y}}_{uv}^{*}$ 和分子中的项进行抵消，并将公式（2.22）代入上式得到：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})}=\frac{\hat{\mathbf{y}}_{uv}^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}-\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\sum_{k=1}^{D}\hat{\varphi}_{uv}^{k}(\mathbf{x})\left(\hat{\varphi}_{uv}^{k}(\mathbf{x})\right)^{*}+\lambda}.\tag{2.28}$$

将公式（2.26）和公式（2.16）代入公式（2.20）可以得到偏导数 $\partial\mathcal{L}/\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})$ 的解：

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} = \dfrac{\partial\mathcal{L}}{\partial\hat{\mathbf{g}}_{uv}^{*}}\dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\left(\hat{\varphi}_{ul}^{l}(\mathbf{x})\right)^{*}} \\ \dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} = \dfrac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\hat{\mathbf{w}}_{uv}^{l}}\dfrac{\partial\hat{\mathbf{w}}_{uv}^{l}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} \end{cases}. \tag{2.29}$$

根据公式（2.4），我们可以推导：

$$\frac{\partial\hat{\mathbf{w}}_{uv}^{l}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} = -\frac{\hat{\mathbf{y}}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\varphi}_{uv}^{l}(\mathbf{x})}{\mu^{2}}. \tag{2.30}$$

将公式（2.21）和公式（2.30）代入公式组（2.29）的第二行得到：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} = -\frac{\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}}{\mu^{2}}. \tag{2.31}$$

将公式（2.22）代入公式（2.31）：

$$\begin{aligned} \frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} &= -\frac{\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\mu\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})} \\ &= -\frac{\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\mu} \end{aligned}, \tag{2.32}$$

其中分子和分母中的 $\hat{\mathbf{y}}_{uv}^{*}\hat{\varphi}_{uv}^{l}(\mathbf{x})$ 消除，代入公式（2.22）到公式（2.32）：

$$\frac{\partial\hat{\mathbf{g}}_{uv}^{*}}{\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}} = \frac{-\hat{\varphi}_{uv}^{l}(\mathbf{x})\left(\hat{\varphi}_{uv}^{l}(\mathbf{z})\right)^{*}\hat{\mathbf{w}}_{uv}^{l}}{\sum_{k=1}^{D}\hat{\varphi}_{uv}^{k}(\mathbf{x})\left(\hat{\varphi}_{uv}^{k}(\mathbf{x})\right)^{*}+\lambda}. \tag{2.33}$$

最后，我们代入公式（2.32）和公式（2.16）到公式组（2.29）的第一行可以得到部分偏导数 $\partial\mathcal{L}/\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})\right)^{*}$ 的解，并合并 $\partial\mathcal{L}/\partial\hat{\varphi}_{uv}^{l}(\mathbf{x})$ 和 $\partial\mathcal{L}/\partial\left(\hat{\varphi}_{uv}^{l}(\mathbf{x})^{*}\right)$ 可以得到：

$$\frac{\partial\mathcal{L}}{\partial\varphi^{l}(\mathbf{x})} = \mathcal{F}^{-1}\left(\frac{\partial\mathcal{L}}{\partial\left(\hat{\varphi}^{l}(\mathbf{x})\right)^{*}} + \left(\frac{\partial\mathcal{L}}{\partial\hat{\varphi}^{l}(\mathbf{x})}\right)^{*}\right). \tag{2.34}$$

通过上面小节，我们完整地推导了目标函数梯度反向传播到特征表示 $\varphi(\mathbf{x})$ 和 $\varphi(\mathbf{z})$ 的过程，通过特征表示梯度反向传播到网络参数的过程可以通过卷积神经网络优化进行实现 [136]。同时，因为在相关滤波层中的所有操作都是在傅里叶频域按逐元素操作（Hadamard operations），相关滤波的高效特性得以保留。在离线训练过程，本章所提出的算法可以在大规模的数据集中进行训练。在离线训练过程完成之后，算法获得了专门针对于判别相关网络优化的特征提取器。

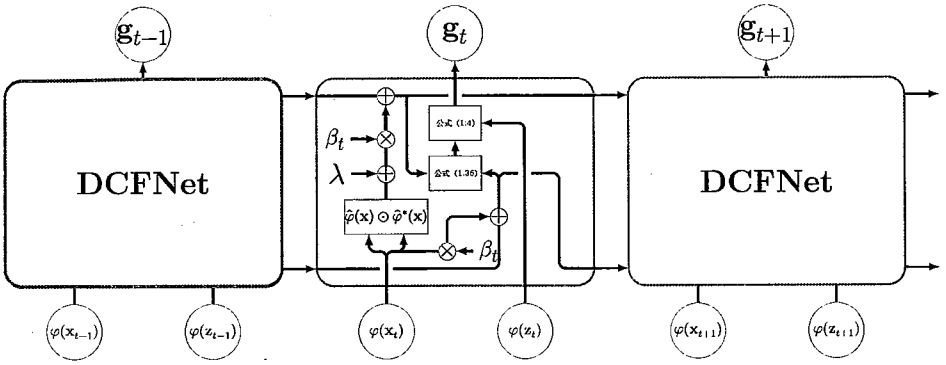


图 2.2 DCFNet 的在线跟踪过程示意图：公式（2.36）中的分子（底部水平箭头）和分母（顶部水平箭头）被递归地向前传播和更新。

2.3.2 判别相关滤波在线模型更新

不同于传统的孪生网络中采用固定相似性度量网络，判别相关滤波网络在线跟踪过程中不断更新滤波器 $\mathbf{w}$。在第 T 帧时，公式（2.2）中的单一模板损失随时间序列增量扩展为：

$$\varepsilon = \sum_{t=1}^{T} \beta_t \left(\left\| \sum_{l=1}^{D} \mathbf{w}^l \star \varphi^l\left(\mathbf{x}_t\right) - \mathbf{y} \right\|_2^2 + \lambda \sum_{l=1}^{D} \left\| \mathbf{w}^l \right\|_2^2 \right), \tag{2.35}$$

其中 t 表示时间索引序号，参数 $\beta_t \geq 0$ 是相应的样本 $\mathbf{x}_t$ 的权重，公式（2.3）中的闭式解扩展到时间维度：

$$\hat{\mathbf{w}}_T^l = \frac{\sum_{t=1}^{T} \beta_t \hat{\mathbf{y}}^* \odot \hat{\varphi}^l\left(\mathbf{x}_t\right)}{\sum_{t=1}^{T} \beta_t \left(\sum_{k=1}^{D} \hat{\varphi}^k\left(\mathbf{x}_t\right) \odot \left(\hat{\varphi}^k\left(\mathbf{x}_t\right)\right)^* + \lambda \mathbb{1} \right)}, \tag{2.36}$$

其中滤波器可以按照下式增量更新：

$$\hat{\mathbf{w}}_T^l = \frac{\beta_T \hat{\mathbf{y}}^* \odot \hat{\varphi}^l\left(\mathbf{x}_T\right) + \sum_{t=1}^{T-1} \beta_t \hat{\mathbf{y}}^* \odot \hat{\varphi}^l\left(\mathbf{x}_t\right)}{\begin{pmatrix} \beta_T \left(\sum_{k=1}^{D} \hat{\varphi}^k\left(\mathbf{x}_T\right) \odot \left(\hat{\varphi}^k\left(\mathbf{x}_T\right)\right)^* + \lambda \mathbb{1} \right) \\ + \sum_{t=1}^{T-1} \beta_t \left(\sum_{k=1}^{D} \hat{\varphi}^k\left(\mathbf{x}_t\right) \odot \left(\hat{\varphi}^k\left(\mathbf{x}_t\right)\right)^* + \lambda \mathbb{1} \right) \end{pmatrix}}, \tag{2.37}$$

该等式表明算法只需要存储当前滤波器的分子与分母，与当前帧滤波器的分子分母部分线性叠加即可实现更新。这种增量更新确保了我们不需要维护线性增加的样本集，减少了内存的消耗。同时 DCFNet 的在线跟踪过程可以看作递归神经网络（Recurrent Neural Network，简称 RNN）的学习过程，如图 2.2所示。

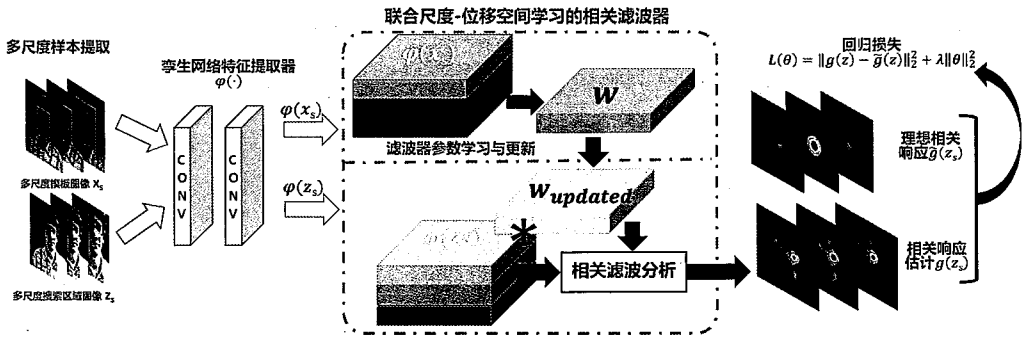


图 2.3 尺度位移相关滤波网络结构示意图。

2.3.3　判别相关滤波的尺度空间扩展

在第 §2.3小节中，我们通过对相关滤波器的反向传播进行推导，实现了特征表示网络与相关滤波器的端到端训练，有效提升了深度特征表示的学习能力。然而，针对于相关滤波器本身而言，所建模的训练样本集合仅包含目标图像的平面位移变化，缺乏对于尺度变化的学习。在本节中我们将尺度和位移空间的联合学习引入判别相关滤波网络，形成如图2.3所示的尺度位移相关滤波网络（Scale Position Correlation Network，简称 SPCNet）。SPCNet 同样采用了轻量化的网络设计，整个架构同样只包含两个卷积层和一个相关滤波层，采用增加不同尺度样本的方式进行算法性能提升。该方法在目标状态的联合尺度和位置空间上具有端到端的可训练性，使得在基于相关滤波的跟踪框架中可以实现较为精确的尺度估计学习。

为了增强判别相关网络对于尺度变化的识别能力，本节所提出的相关滤波层将目标和上下文外观的相关分析建模在联合尺度-位移空间中，而不局限在位移空间中。因此，在跟踪过程中，SPCNet 可以同时学习并精确估计目标大小和位置的变化。不同于原始判别相关滤波只采集单一尺度目标图像，我们进行多尺度采样学习。具体地，算法采集 S 个不同尺度的目标样本 $\mathbf{x}_s$，其中各样本的尺度系数依次为 $\left\{\sigma^s | s=\left\lfloor -\frac{S-1}{2}\right\rfloor, \left\lfloor -\frac{S-3}{2}\right\rfloor, \ldots, \left\lfloor \frac{S-1}{2}\right\rfloor\right\}$。参考相关滤波器在平面位移建模时的理想响应，算法设定在目标尺度下的响应最大，当尺度过小或者过大时相应的理想响应下降。因而算法设定联合尺度目标响应表示为 $\mathbf{y}_s=\mathbf{y}\odot\eta^{s^2}$，$\eta$ 表示在尺度维度上的响应衰减，其响应分布可以参考图2.3右侧所示。我们将公式（2.2）中的单一尺度脊回归损失扩展为多尺度脊回归损失：

$$\mathbf{w}=\underset{\mathbf{w}}{\operatorname{argmin}}\sum_{s=1}^{S}\left\|\sum_{l=1}^{D}\mathbf{w}^l\star\boldsymbol{\varphi}^l(\mathbf{x}_s)-\mathbf{y}_s\right\|_2^2+\lambda\sum_{l=1}^{D}\left\|\mathbf{w}^l\right\|_2^2, \tag{2.38}$$

由于上式可以等价为将原始特征进行了线性扩展，该问题同样具有傅里叶频域的闭式解表示：

$$\hat{\mathbf{w}}^{l}=\frac{\sum_{s=1}^{S}\hat{\varphi}^{l}\left(\mathbf{x}_{s}\right)\odot\hat{\mathbf{y}}_{s}^{*}}{\sum_{s=1}^{S}\sum_{k=1}^{D}\hat{\varphi}^{k}\left(\mathbf{x}_{s}\right)\odot\left(\hat{\varphi}^{k}\left(\mathbf{x}_{s}\right)\right)^{*}+\lambda\mathbb{1}},\tag{2.39}$$

上式表明，通过对尺度样本的联合学习，滤波器学习过程对于错误尺度的判别性会有所提高。在后续视频帧 $\mathbf{I}_{t}$ 中，我们通过在上一帧所预测的目标位置中心位置采集多尺度候选样本 $\mathbf{z}_{s}$ 分别与学习到的尺度感知滤波器进行相关分析：

$$g\left(\mathbf{z}_{s}\right)=\sum_{l=1}^{D}\mathbf{w}^{l}\star\varphi^{l}\left(\mathbf{z}_{s}\right)=\mathcal{F}^{-1}\left(\sum_{l=1}^{D}\hat{\mathbf{w}}^{l^{*}}\odot\hat{\varphi}^{l}\left(\mathbf{z}^{s}\right)\right),\tag{2.40}$$

可以发现，由于滤波器 $\mathbf{w}$ 的通道数 D 相对于单尺度判别相关滤波学习（见公式(2.3)）中的通道数并没有提升，因而算法保持了良好的测试速度。下面我们将对具有尺度位移空间联合学习特性的 SPCNet 进行在线学习扩展。我们利用图像帧 $\mathbf{I}_{t}$ 进行多尺度测试构成新的训练样本，在第 t 帧的多尺度响应估计为：

$$\left\{g\left(\mathbf{x}_{s,t}\right)=\sum_{l=1}^{D}\mathbf{w}^{l}\star\varphi^{l}\left(\mathbf{x}_{s,t}\right)\right\}_{s=1}^{S}.\tag{2.41}$$

此外，我们进一步地将公式（2.39）中的联合尺度位移空间的相关滤波器学习扩展到时间维度中，进行增量约束：

$$\varepsilon=\sum_{i=1}^{T}\beta_{t}\left(\sum_{s=1}^{S}\left\|g\left(\mathbf{x}_{s,t}\right)-\mathbf{y}_{s}\right\|_{2}^{2}+\lambda\sum_{l=1}^{D}\left\|\mathbf{w}^{l}\right\|_{2}^{2}\right),\tag{2.42}$$

其中 $\beta_{t}\geq 0$ 表示不同时刻 t 所对应的训练样本的权重，固定的正实数 $\lambda\geq 0$ 表示滤波器正则系数。通过将公式（2.41）转换到傅里叶频域进行求解。由于该目标函数为实数、非负且为凸函数，其全局最优解处导数为 0。问题的解为：

$$\hat{\mathbf{w}}^{l}=\frac{\sum_{i=1}^{T}\beta_{i}\left(\sum_{s=1}^{s}\hat{\mathbf{y}}_{s}^{*}\odot\hat{\varphi}^{l}\left(\mathbf{x}_{st}\right)\right)}{\sum_{i=1}^{T}\beta_{i}\left(\sum_{s=1}^{s}\sum_{k=1}^{D}\hat{\varphi}^{k}\left(\mathbf{x}_{st}\right)\odot\left(\hat{\varphi}^{k}\left(\mathbf{x}_{st}\right)\right)^{*}+\lambda\mathbb{1}\right)},\tag{2.43}$$

由于上式同样只需要进行快速傅里叶变换以及逐元素的乘除，其计算复杂度仍然可以保持为 $\mathcal{O}(SDA\log A)$，可以十分高效运行。此外，由于算法保留了闭式解，计算过程避免了使用较为复杂的迭代优化策略 [54]。

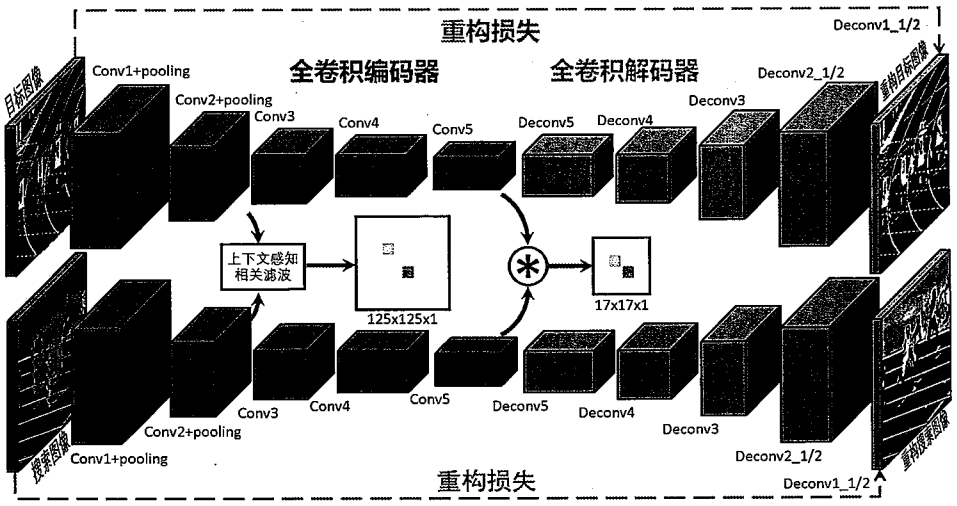


图 2.4 编解码相关滤波跟踪算法网络结构示意图。

2.4　基于编解码相关滤波网络的跟踪算法

我们在上一节中提出了端到端学习的相关滤波跟踪框架，在此基础上本节主要探索基于深度特征的语义嵌入模型，并提出使用编解码自监督学习孪生网络实现对目标及其周围环境结构信息的有效感知，提升了特征表示的泛化性能与细粒度表示能力。

基于判别相关滤波器的端到端学习架构更关注于网络的在线判别学习，具有较强的目标适应性。但是由于其在线更新跟踪特性，不可避免地会累计学习错误的跟踪估计，形成跟踪漂移。而基于孪生卷积网络 [7] 的目标跟踪主要利用网络特征离线学习一个具有高度判别力的度量空间映射，在线过程不进行更新。为了同时保留两种跟踪算法各自的适应性以及高效特性，我们提出利用目标的底层特征和高层特征融合跟踪。本节所构建的孪生网络框架同时学习底层的细节表示以及高层的语义描述，进行互补增强。

本节所提出的编解码相关滤波（Encoder-Decoder Correlation Filters，简称 EDCF）跟踪框架如图2.4所示，该框架充分利用具有多分辨率特征表示的编解码网络结构。算法利用底层高分辨率特征进行相关滤波分析，实现了目标的精确定位。同时，我们在相关滤波分析中引入了全局上下文约束进行正则化，并对该滤波器进行端到端的训练。此外，我们提出使用编解码自监督学习孪生网络实现对目标图像的结构信息的感知。整体算法基于多任务学习策略实现了端到端的跨层级特征融合表示，增强了网络的鲁棒性。

2.4.1 通用语义嵌入学习模型

不同于只关注判别学习的孪生网络跟踪算法 [7, 10]，本节提出通过引入额外的重构误差损失来学习更加通用的高层特征表示网络。由于图像重构是自监督学习的任务，且训练过程与图像物体类别无关，相比于判别式学习对于训练数据的类别分布更加鲁棒。算法通过自监督损失进行正则使得语义嵌入层更加鲁棒，增加了整体跟踪的鲁棒性。更为重要的是，重构约束使得语义嵌入空间保留了所有的纹理以及结构信息，增加了跟踪算法的精度。

本节提出的通用语义嵌入学习模型基于编解码网络架构，其中编码器网络 ϕ：将原始图像表示投影到语义嵌入空间 $\mathbb{R}^{M\times N\times 3} \rightarrow \mathbb{R}^{P\times Q\times D}$，网络架构由 5 个卷积层以及两个最大池化层（max pooling）组成。解码器 ψ：将低分辨率的语义嵌入表示还原到原始图像分辨率的图像空间 $\mathbb{R}^{P\times Q\times D} \rightarrow \mathbb{R}^{M\times N\times 3}$。解码器网络由 7 个反卷积层（transposed convolution）组成。

对于高层语义跟踪架构，算法通过向量内积来度量目标图像和搜索图像间的相似度：

$$f_{u,v} = \sum_{i=0}^{P-1}\sum_{j=0}^{Q-1} \left\langle \phi_{i,j}(\mathbf{x}), \phi_{u+i,v+j}(\mathbf{z}) \right\rangle \tag{2.44}$$

其中 $\phi_{i,j}(\mathbf{x})$ 表示模板图像 $\mathbf{x}$ 的高层语义特征表示在 (i,j) 处的 D 维特征向量，$P\times Q$ 为模板图像特征的平面分辨率。公式 (2.44) 左边的 $f_{u,v}$ 表示搜索图像特征中左上角坐标为 (u,v) 所对应的相关响应。算法通过二值标签 $y(u,v) \in \{+1,-1\}$ 对于搜索图像的每个空间位置进行分类。高层语义跟踪的损失函数采用 Logistic 损失：

$$\mathcal{L}_{cls} = \frac{1}{|\mathcal{D}|}\sum_{(u,v)\in\mathcal{D}} \log\left(1+\exp\left(-y(u,v)f_{u,v}\right)\right), \tag{2.45}$$

其中 $\mathcal{D}$ 是搜索图像的响应图集合。

孪生网络方法离线在大规模标注数据集中进行监督学习，网络训练需要消耗大量的标注样本。例如，算法 SiamFC[7] 需要 4000 多段视频序列进行训练，其后续改进工作 SiamRPN[70] 甚至需要 20 万段视频序列用来训练网络，这为数据标注的使用带来了极大的困难。与监督学习（supervised learning）不同，自监督学习（self-supervised learning）并不需要特殊标注的数据，其学习过程利用数据本身的结构特征进行特征学习。算法通过对图像中的结构信息学习，提升网络的泛化性能。本节提出通过联合优化图像重构损失 $\mathcal{L}_{recon}$ 和跟踪分类损失 $\mathcal{L}_{cls}$ 来

学习孪生网络架构参数：

$$\mathcal{L}_{\text{high}} = \mathcal{L}_{recon} + \mathcal{L}_{cls}, \tag{2.46}$$

$$\mathcal{L}_{\text{recon}} = \left\| \psi\left(\phi\left(\mathbf{x};\theta_e\right);\theta_d\right) - \mathbf{x} \right\|_2^2 + \left\| \psi\left(\phi\left(\mathbf{z};\theta_e\right);\theta_d\right) - \mathbf{z} \right\|_2^2, \tag{2.47}$$

其中编码器和解码器的网络参数分别是 θ_e 和 θ_d，$\mathbf{x}$ 是目标模板图像，$\mathbf{z}$ 是搜索区域图像。我们利用编解网络得到的复原目标图像 $\psi\left(\phi\left(\mathbf{z};\theta_e\right);\theta_d\right)$ 和原始目标图像 $\mathbf{z}$ 以及复原搜索区域图像 $\psi\left(\phi\left(\mathbf{z};\theta_e\right);\theta_d\right)$ 和原始搜索区域图像 $\mathbf{z}$ 构成两个样本对进行自监督重构损失训练。同时，我们也可以把这个重构损失看作是监督学习的正则项，通过额外的正则学习来增加网络的泛化性能。重构约束增强了语义嵌入学习的鲁棒性，同时使得高层特征表示保留了图像的结构信息。

2.4.2　上下文空间感知的自适应相关滤波跟踪算法

为了更准确的对目标进行定位，算法仍然需要对低层高分辨率的特征表示进行相关滤波分析。传统的相关滤波算法在线学习过程中缺乏训练样本，通常难以适应具有干扰对象的场景。本节提出在相关滤波分析中加入全局上下文约束来抑制干扰物的影响，这个约束是通过一个可微分的相关滤波器层进行实现。

如图2.4所示，我们在目标图像块 $\mathbf{x}_0$ 周围采样 k 上下文图像块 $\mathbf{z}_i$（图中红色区域）。它们的低层高分辨率特征表示分别为 $\varphi(\mathbf{x}_0)$ 和 $\varphi(\mathbf{x}_i)$。上下文图像块可以看作是包含干扰对象的困难负样本。因而我们将上下文图像块的理想响应设置为零，上下文感知的相关滤波器的目标函数定义为：

$$\min_{\mathbf{w}} \left\| \mathbf{w} \star \varphi(\mathbf{x}_0) - \mathbf{y} \right\|_2^2 + \lambda_1 \left\| \mathbf{w} \right\|_2^2 + \lambda_2 \sum_{i=1}^{k} \left\| \mathbf{w} \star \varphi(\mathbf{x}_i) \right\|_2^2, \tag{2.48}$$

相比于公式（2.2），上式额外增加了一个损失项用于减弱困难样本的影响。同时，该优化目标在傅里叶频域的闭式解为：

$$\hat{\mathbf{w}} = \frac{\hat{\varphi}(\mathbf{x}_0) \odot \hat{\mathbf{y}}^*}{\hat{\varphi}(\mathbf{x}_0) \odot \hat{\varphi}^*(\mathbf{x}_0) + \lambda_1 \mathbb{1} + \lambda_2 \sum_{i=1}^{k} \hat{\varphi}(\mathbf{x}_i) \odot \left(\hat{\varphi}(\mathbf{x}_i)\right)^*}. \tag{2.49}$$

我们提出通过将上述相关滤波器转换成网络中的可微相关滤波层，级联在编码器的低层卷积特征之后，自动学习适合相关滤波的低层精细特征表示。该设计允许算法对整个基于编解码结构的孪生网络进行端到端训练。通过相关滤波损失来优化低层卷积网络的特征表示：

$$\mathcal{L}_{cf} = \left\| \mathbf{g}(\mathbf{z}) - \mathbf{y} \right\|_2^2. \tag{2.50}$$

公式（2.50）中的相关滤波损失 $\mathcal{L}_{cf}$ 的反向传播过程推导如下:

$$\frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{g}}^*} = 2(\hat{\mathbf{g}}(\mathbf{z}) - \hat{\mathbf{y}}). \tag{2.51}$$

对于搜索图像跟踪分支的梯度反向传播可以表示为:

$$\frac{\partial \mathcal{L}_{cf}}{\partial \varphi(\mathbf{z})} = \mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{g}}^*} \odot \hat{\mathbf{w}}^*\right). \tag{2.52}$$

对于模板图像作为输入的滤波器学习分支，我们首先计算滤波器 $\mathbf{w}$ 的反向传播梯度:

$$\frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{w}}} = \frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{g}}^*} \odot (\hat{\varphi}(\mathbf{z}))^*, \tag{2.53}$$

对于目标图像块特征 $\varphi(\mathbf{x}_0)$ 和上下文图像块特征 $\varphi(\mathbf{x}_i)$ 的梯度分别如下:

$$\frac{\partial \mathcal{L}_{cf}}{\partial \varphi(\mathbf{x}_0)} = \mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{w}}} \odot \frac{\hat{\mathbf{y}}^* - 2\,\mathrm{Re}\left(\hat{\varphi}^*(\mathbf{x}_0) \odot \hat{\mathbf{w}}\right)}{\hat{\mathbf{D}}}\right), \tag{2.54}$$

$$\frac{\partial \mathcal{L}_{cf}}{\partial \varphi(\mathbf{x}_i)} = \mathcal{F}^{-1}\left(\frac{\partial \mathcal{L}_{cf}}{\partial \hat{\mathbf{w}}} \odot \frac{-2\,\mathrm{Re}\left(\hat{\varphi}^*(\mathbf{x}_0) \odot \hat{\mathbf{w}}\right)}{\hat{\mathbf{D}}}\right), \tag{2.55}$$

其中 $\hat{\mathbf{D}} := \hat{\varphi}^l(\mathbf{x}_0) \odot \left(\hat{\varphi}^l(\mathbf{x}_0)\right)^* + \lambda_1 \mathbb{1} + \lambda_2 \sum_{i=1}^{k} \hat{\varphi}^l(\mathbf{x}_i) \odot \left(\hat{\varphi}^l(\mathbf{x}_i)\right)^*$ 是滤波器 $\hat{\mathbf{w}}$ 的分母部分，$\mathrm{Re}(\cdot)$ 表示复数矩阵的实数部分。

2.4.3 编解码相关滤波网络的多任务学习

上面两小节分别构建了自监督学习语义嵌入模型和具有上下文感知能力的判别相关滤波器，这两种算法在跟踪过程中可以相互补充。本节提出利用多任务学习策略（multi-task learning）对网络进行端到端训练，同时强化这两个组件。我们所构建的多任务损失函数定义为:

$$\mathcal{L}_{all} = \mathcal{L}_{cf} + \mathcal{L}_{cls} + \mathcal{L}_{recon} + \mathcal{R}(\theta), \tag{2.56}$$

其中 $\mathcal{R}(\theta)$ 表示对所有的网络参数进行 L2 范数正则，该正则项用来增加网络的泛化性能。

2.4.4 编解码相关滤波网络的高效目标跟踪

在跟踪过程，给定第 t 帧图像输入，我们基于上一帧图像跟踪结果的中心位置在当前图像裁剪具有多个尺度的搜索图像 $\mathbf{z}_s$。这些搜索图像输入到编码器网络分别获得低层细粒度特征表示以及高层语义表示。底层特征表示通过公式

(2.4) 计算相关滤波响应，高层语义特征被代入公式 (2.44) 中求解语义相似度响应。然后通过寻找融合相关响应的最大值位置来估计目标状态:

$$\underset{(u,v,s)}{\operatorname{argmax}} \; f_{u,v}(\mathbf{z}_s) + g_{u,v}(\mathbf{z}_s), \tag{2.57}$$

注意，我们使用双线性插值方法对高层特征的相关响应 $f_{u,v}(\mathbf{z}_s)$ 进行向上采样，以获得与底层相关滤波响应 $g_{u,v}(\mathbf{z}_s)$ 一样的空间分辨率。由于跟踪过程只涉及网络的前向传递计算，因此算法具有较好的实时性。

2.5　实验评估与分析

本章所提出的 DCFNet、SPCNet 以及 EDCF 均使用神经网络工具包 MatConvNet[137] 在 MATLAB 平台实现。本章所有的实验在一个配备 Intel Xeon 2630 的 CPU 和 NVIDIA GeForce GTX 1080 的 GPU 的工作站运行测试。我们使用下列数据集进行对照实验以及总体跟踪性能评估:

- **OTB 数据集**: OTB 数据集包含 OTB-2013[3] 和 OTB-2015[1] 两个子集，是当前最主流的目标跟踪测试集。两个子集分别包含隶属于 11 个不同属性场景的 50 段视频和 100 段视频。我们在 OTB 数据集上的部分实验使用距离精度 (Distance Precision) 和重叠精度 (Overlap Precision) 评价指标。距离精度表示中心定位误差小于 20 像素阈值的图像帧比例，重叠精度表示矩形框交并比超过 0.5 阈值的帧的百分比。重叠成功曲线是在不同阈值下绘制的算法重叠精度。在此图中，我们使用图中显示的曲线下面积 (Area Under Curve，简称 AUC) 对跟踪器进行排序。

- **VOT 数据集**: VOT 挑战赛是跟踪领域最具影响力和规模的年度赛事之一。在 VOT-2015 数据集 [4] 中，期望平均重叠率 (Expected Average Overlap，简称 EAO) 指标被用来定量分析跟踪器的性能，该指标可以用来估计在大量的具有类似属性的短时序列中，跟踪算法的平均重叠率。该测量方法有效解决了由于序列长度变化引起的平均重叠测量偏差和方差增大问题 [4]。

下面分别介绍实验具体设置，本节首先对所提出的主要部件的有效性与速度性能进行对照实验研究。然后我们将本章所提出的跟踪器与当前较为先进的跟踪器进行整体性能比较。

2.5.1 实验设置

算法主要的实现设置包括网络架构、相关滤波器参数、离线训练数据、优化器参数设置。

- **网络架构**：本章提出的判别相关滤波网络 DCFNet 和尺度位移相关滤波网络 SPCNet 的特征提取网络由两个卷积层组成，卷积核大小设置为 3×3，每个卷积层后都级联 ReLU 层进行非线性变换，最终的输出特征表示通道数 D 被设定为 32。在对照实验中，我们分析了网络不同通道数所取得的性能。编解码相关滤波网络 EDCF 使用与 SiamFC[7] 相同的网络架构，通过移除 AlexNet 的全连接的层（fully connected layer）来实现。

- **相关滤波参数设定**：对于相关滤波层，公式（2.2）中的正则化系数 λ 被设置为 10^{-4}。公式（2.36）中的在线学习率 β_t 设置为 0.008。尺度位移相关滤波网络中的多尺度响应衰减系数 η 设置为 0.9。在上下文感知的相关滤波器中，我们将公式 (2.50) 中的正则化参数设置为 $\lambda_1 = 10^{-4}$，$\lambda_2 = 0.1$。对于在线跟踪和离线训练中，理想目标响应公式（2.1）中高斯空间带宽系数 σ 设置为 0.1。同时与 [32] 相类似，我们使用带比例因子的图像金字塔进行多尺度测试：

$$\left\{\sigma^s|\sigma = 1.01, s = \left\lfloor -\frac{S-1}{2} \right\rfloor, \left\lfloor -\frac{S-3}{2} \right\rfloor, \ldots, \left\lfloor \frac{S-1}{2} \right\rfloor \right\}. \tag{2.58}$$

- **离线训练数据**：为了增加特征表示的泛化能力和判别能力，同时避免在稀缺的跟踪数据上过拟合（over-fitting），本章所提出的算法在大规模视觉识别挑战视频检测数据集（ILSVRC VID）[6] 上端到端地进行训练。该训练集由 7911 个目标对象组成，与 OTB 和 VOT 测试数据集没有重叠。该数据集包含 4000 多个序列和近 200 万个目标图像矩形框标注。为了训练 DCFNet，在每个视频片段中选择一个目标，然后从最近的 10 帧内收集样本对，并将裁剪好的具有 2 倍上下文的目标图像块输入网络。总共构成 550 万对样本用于 DCFNet 训练。在线跟踪过程中，裁剪的输入被调整为与离线训练阶段相一致的空间分辨率。通过速度和精度的平衡曲线（见图2.5），我们建议使用 125×125 分辨率进行训练与测试。

- **网络优化器参数**：算法采用动量（momentum）为 0.9 的随机梯度下降（SGD）优化器对网络参数 θ 进行随机初始化训练。公式（2.5）中的权重衰减（weight decay）系数 γ 被设置为 0.0005。学习率从 10^{-2} 到 10^{-5} 进行指数衰减。该模型训练了 50 个数据循环（epoch），批数量（batch size）设置为 32。

表 2.1 基于数据集 OTB-2013[3] 本章提出的判别相关滤波网络 DCFNet 在不同参数设置下的平均重叠精度（阈值为 0.5）、平均距离精度（阈值为 20 像素）和平均速度对比展示。

算法名称	重叠精度（%）	距离精度（%）	速度（fps）
DCFNet-*conv1*	61.0	70.4	211
DCFNet-*conv2-dilation*	66.3	77.3	120
DCFNet-*conv3*	64.3	75.3	61
DCFNet-*conv2-1s*	67.7	79.1	187
DCFNet-*conv2-3s*	78.5	86.7	109
DCFNet-*conv2-5s*	76.3	83.8	69
DCFNet-*conv2-7s*	77.4	88.0	53

2.5.2　创新点有效性验证

本节针对判别相关滤波网络 DCFNet、尺度位移相关滤波网络 SPCNet 以及编解码相关滤波网络 EDCF 的有效性进行了对照实验分析。

首先我们对于判别相关滤波网络 DCFNet 中的结构参数以及多尺度测试性能进行对照分析。

- 对于神经网络架构，随着卷积层数量的增加，训练参数和感受野（receptive field）的大小逐渐增加。表2.1显示，在 OTB-2013 测试集中，两层卷积的 DCFNet-conv2 与三层卷积的 DCFNet-conv3 相比具有更好的精度。为了更好地理解这一观察结果，我们使用膨胀卷积（dilated convolution）对 DCFNet-conv2 进行了修正，通过膨胀卷积来近似更深层次的卷积感受野。这种结构相比于 conv3 需要更少的参数并取得更好的跟踪精度。也即实验验证，对于判别相关滤波网络，直接加深网络层数并不能带来性能提升。
- 在多尺度性能测试方面，DCFNet-3s、DCFNet-5s 和 DCFNet-7s 仅在跟踪过程中对 DCFNet 进行了 3、5、7 个相邻尺度级别的尺度估计，在学习的过程中有一个尺度模板。表2.1结果表明，三个尺度搜索的设计在跟踪精度和跟踪速度之间具有良好的平衡。

然后，我们通过对照实验验证第 §2.3小节提出的端到端学习的判别相关滤波特征相对于传统特征网络的优势以及第 §2.3.3小节提出的尺度位移空间联合学习对于精度的提升。下面将本章提出的 DCFNet 跟踪器、SPCNet 跟踪器与包括 DCF+VGG、DCF+SiamFC、DCF ([29] 的线性相关滤波版本)、SAMF[32] 和

表 2.2 基于数据集 OTB-2013[3]、OTB-2015[1] 本章提出的基于端到端学习的判别相关滤波网络的跟踪算法与相关算法的平均重叠精度（阈值为 0.5）、平均距离精度（阈值为 20 像素）和平均速度对比展示。

算法名称		OTB-2013[3]		OTB-2015[1]		速度（fps）
		重叠精度（%）	距离精度（%）	重叠精度（%）	距离精度（%）	
本章算法	DCFNet	67.7	79.1	63.7	76.8	187
	DCFNet-3s	78.5	86.7	72.8	79.4	109
	SPCNet	**84.3**	**88.1**	**77.6**	**82.9**	67
预训练算法	DCF+VGG[59]	62.1	66.1	61.7	66.9	88
	DCF+SiamFC[7]	66.8	74.2	64.0	68.0	77
对照算法	DCF[29]	61.6	72.8	54.8	68.9	**292**
	SAMF[32]	67.7	78.5	64.0	74.3	12
	DSST[89]	67.1	74.7	60.9	68.9	46

DSST[89] 在内的一些基线进行了比较。结果如表2.2所示，其中 DCFNet 在训练和跟踪测试过程中均没有考虑尺度因素，而 DCFNet-3s 仅在跟踪过程中对 DCFNet 进行了 3 个相邻尺度采样的尺度估计，SPCNet 使用了尺度-位移空间的联合学习。通过分析表2.2中结果，我们可以得出：

- 与传统使用手工特征的判别相关滤波器跟踪算法 DCF 相比，基于端到端学习的 DCFNet 在保持实时跟踪速度的同时，在 OTB-2013 数据集和 OTB-2015 数据集分别获得 6.3% 和 7.9% 的距离精度提升，以及 6.1% 和 9.9% 的重叠精度提升。该结果证明了深度学习特征的有效性，相比于手工设计的特征，深度学习特征具有更好的光照、纹理适应性。为了消除深度学习网络带来的提升对对比的影响，我们设置 DCF+VGG 和 DCF+SiamFC 两个分别使用的是 VGG[59] 和 SiamFC[7] 中的原始特征网络来替换 DCFNet 中的特征提取模块。与这两个采用其他任务预训练的判别相关滤波算法对比，本章所提出的 DCFNet 专门学习针对目标跟踪问题的特征表示，在 OTB-2013 数据集的距离精度指标中提升达到 4.9%。因此，实验验证了通过端到端训练获得的特征表示对跟踪是有效的。

- 通过将候选空间搜索空间从单一尺度位置空间扩展到多尺度位置空间，DCFNet-3s 优于传统的多尺度跟踪器 SAMF[32] 和 DSST[89]。该结果同样得益于端到端学习对于特征表示的提升。DCFNet-3s 通过多尺度测试的方法来估计目标

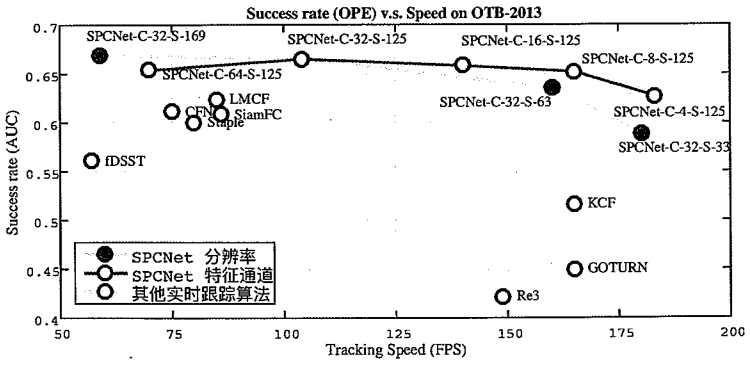
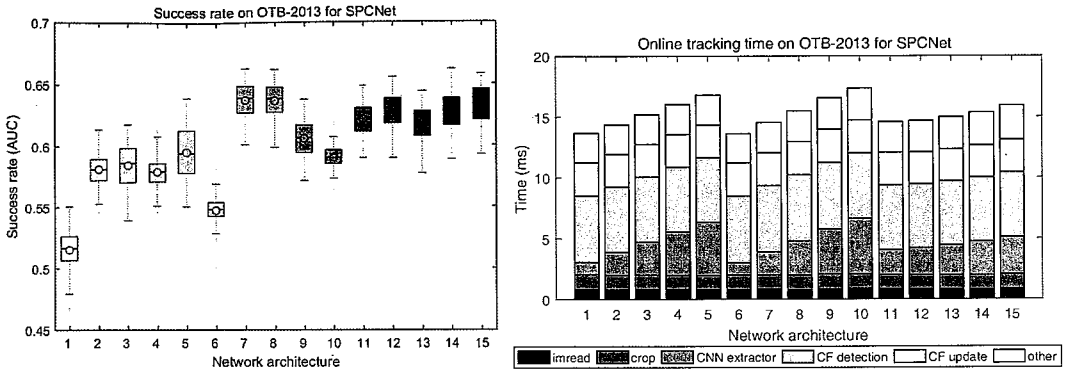


图 2.5 基于数据集 OTB-2013[3] 本章提出的 SPCNet 与其他常用实时跟踪器的 AUC 跟踪速度与精度对比展示，以及 SPCNet 的跟踪速度与精度平衡曲线：通过调整图像网络的通道数量（黄色曲线所示）以及图像分辨率（蓝色曲线所示）可以有效达到速度与精度的平衡。

尺度变化可以取得比多阶段的 DSST 更快的跟踪速度。通过对比 DCFNet-3s 和 SPCNet 的结果，可以得出通过多尺度样本的训练可以得到更加精确的尺度估计，相比于只进行多尺度测试的 DCFNet-3s，SPCNet 的重叠精度指标提升达到 5.8% 和 4.8%，滤波器学习过程中加入了多尺度样本，有利于对于尺度变化的感知。

- 本章所提出的 SPCNet 在表2.2中的各项精度指标排名第一，相对于基于传统特征的尺度估计相关滤波器在速度上同样具有优势。这是因为通过端到端的特征学习，本章提出的深度特征表示网络较手工设计特征具有更高的判别性、光照变化鲁棒性以及尺度变化的适应性，降低了表观模型对于尺度采样的密集程度，节约了计算开销。

为了突出 SPCNet 在跟踪精度和速度之间的平衡，我们比较 SPCNet 在空间分辨率和网络通道数设置上的影响：通过使用 169×169、125×125、63×63 和 33×33 这四个不同的输入空间分辨率和使用 64、32、16、8 和 4 个通道作为网络的输出特征。图2.5显示了不同设定下 SPCNet 模型和其他常用实时跟踪器 [7, 11, 29, 35, 68, 75, 138, 139] 在 OTB-2013[3] 上 AUC 跟踪精度和速度对比。通过图中蓝色曲线可以看出，输入空间分辨率的降低会导致算法的 AUC 精度大幅降低，尽管它提供了显著的加速效果。当网络分辨率过低时，目标表观信息损失较为严重，这种信息损失会显著降低跟踪精度性能。通过图中黄色曲线可以得出，少量输出特征通道使得 AUC 性能下降 4%，而运行速度提升超过 3 倍。与对比算法中超过 100fps 的快速跟踪器 KCF[29]、GOTURN[35] 以及 Re3[139] 相比，本章所提出的 SPCNet 在精度和速度上都有更好的表现。通过端到端学习，



(a) 不同网络结构在 OTB-2013 上的 AUC。　　(b) 不同网络结构的各部件时间消耗。

图 2.6 基于数据集 OTB-2013[3] 本章提出的 SPCNet 在 3 种不同网络架构下的性能指标和各部件时间消耗对比展示。

我们使得特征网络的设计自由度显著增加，而直接采用预训练的跟踪算法通常无法调整网络设计。根据可用的计算资源不同，SPCNet 模型可以实现跟踪速度从 60fps 到 190fps 的不同设置，在实际应用中具有更好的可行性。

此外，本节通过设计 3 组不同的网络架构来搜索适合判别相关跟踪算法的网络结构：（1）第一组（编号 1-5）采用 1×1 的卷积核进行堆叠，网络结构层数依次增加，网络感受野保持不变而特征表示能力依次增加。（2）第二组（编号 6-10）采用 3×3 的卷积核进行堆叠，网络结构层数依次增加，网络感受野和特征表示能力依次增加。（3）第二组（编号 11-15）采用两个 3×3 的卷积核进行堆叠，分别采用 64、96、128、160、192 的作为中间层输出通道数量，网络感受野保持不变而特征表示能力依次增加。我们可以通过图2.6得出以下经验结论：

- 在相同感受野的网络设计中，随着网络的加深（网络 1-5）以及中间层通道数量（网络 11-15）的增加，网络的性能会逐步提升，但同时影响跟踪速度。
- 在不同感受野的网络中，两层网络的性能较为突出，过深（大于 3 层）或者过浅（1 层）的网络设计都会降低网络的性能。判别相关滤波器较为适应浅层特征表示。
- 由于轻量化的网络设计，算法的主要时间开销在相关滤波操作，特征提取部分耗时相对较少。

通过上述实验分析，我们证明了端到端的特征表示学习较传统手工设计特征和预训练神经网络在性能与跟踪效率上有明显提升；基于尺度-位移空间联合学习的判别相关滤波在精度性能上有显著提升。

表 2.3 基于数据集 OTB-2013[3]、OTB-2015[1] 本章提出的基于编解码相关滤波网络的跟踪算法与相关算法的平均重叠精度（阈值为 0.5）、平均距离精度（阈值为 20 像素）和平均速度对比展示。

算法名称	OTB-2013[3]		OTB-2015[1]		速度（fps）
	重叠精度（%）	距离精度（%）	重叠精度（%）	距离精度（%）	
SiamFC[7]	77.8	80.9	73.0	77.0	86
EDSiam	79.0	83.9	75.4	80.7	86
CFNet[11]	71.7	76.1	70.3	76.0	75
CACF[140]	75.4	80.3	68.9	79.1	13
CACFNet	83.8	87.6	77.7	82.7	109
CACFNet+	83.9	88.3	78.0	83.1	109
EDCF	84.2	88.5	78.5	83.6	65

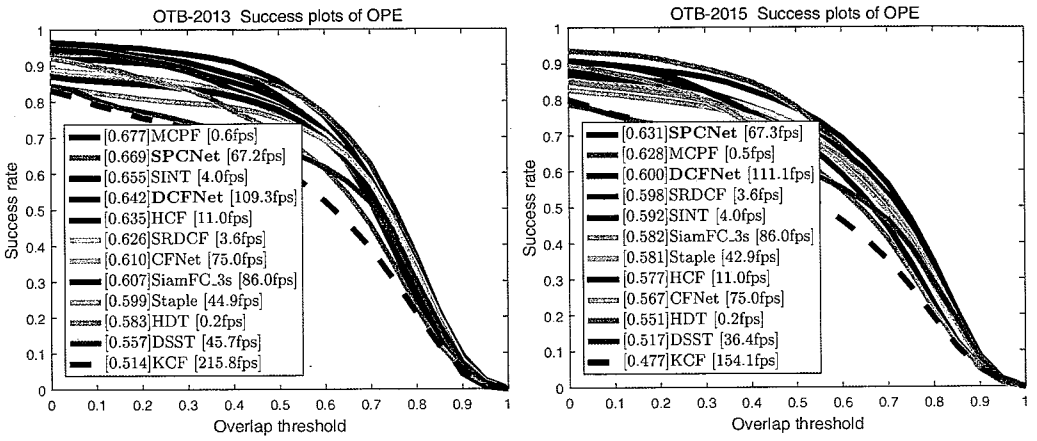
最后我们将本章提出的编解码相关滤波网络 EDCF 与 CFNet[11] 和 CACF[140] 进行对比。通过表2.3中结果，我们可以得出以下结论：

通用语义嵌入学习模型：通过在基于编解码网络架构改进的 SiamFC[7] 跟踪器中引入图像重构约束（记作 EDSiam），在 OTB-2015 数据集上获得了 3.7% 的距离精度收益。该自监督正则约束提高了语义嵌入表示的泛化能力，保证了鲁棒跟踪。

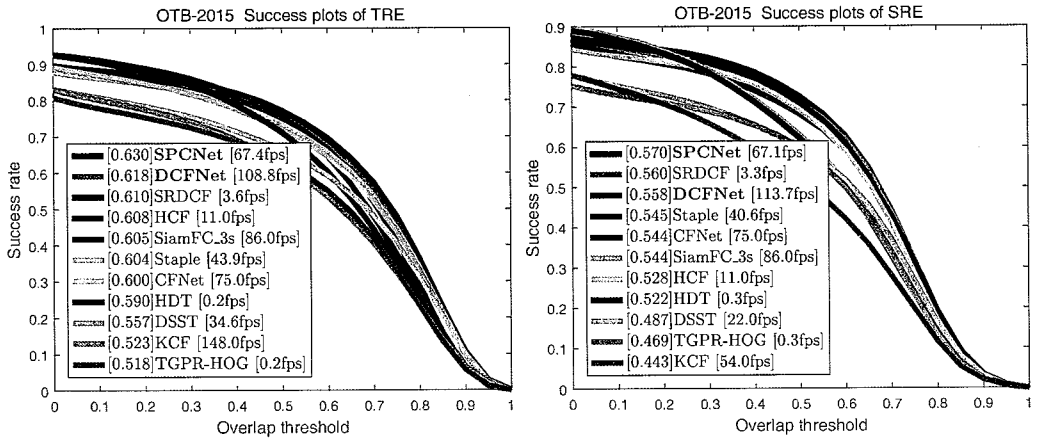
上下文相关过滤器：我们在第 §2.4.2节中上下文空间感知的相关滤波算法记为 CACFNet，该方法与使用 HOG 特征的上下文感知相关滤波器算法 CACF[140] 相比，在 OTB 数据集上获得了超过 8% 的重叠精度提升，这证明了端到端学习的特征表示比传统的手工设计特征更有判别力。与同样采用端到端学习的 CFNet[11] 跟踪器相比，CACFNet 取得了超过 10% 的重叠精度提升，该结果证明了全局上下文约束的有效性。

多任务学习：CACFNet+ 表示将语义嵌入学习和重构约束引入到 CACFNet 的训练过程中。在跟踪阶段，CACFNet+ 仅使用上下文感知的相关滤波器估计目标位置。CACFNet+ 相对于 CACFNet 的性能提升证明了多任务学习强化了相关滤波器特征表示。最后，通过公式（2.57）融合多层次响应的 EDCF 的性能优于 CACFNet+，这表明相关滤波模型和语义嵌入模型的融合是有效的。

表2.3也显示了编解码相关滤波网络 EDCF 的跟踪速度，本章提出的所有跟踪算法均实现了实时目标跟踪。



(a) OTB-2013 的 OPE　　**(b) OTB-2015 的 OPE**



(c) OTB-2015 的 TRE　　**(d) OTB-2015 的 SRE**

图 2.7 基于数据集 OTB-2013[3]、OTB-2015[1] 本章提出的 SPCNet 与相关算法的 OPE 性能对比展示，基于数据集 OTB-2015[1] 本章提出的 SPCNet 与相关算法的 TRE、SRE 性能对比展示。

2.5.3　基于数据集 OTB-2013 和 OTB-2015 的评测结果分析

图2.7显示了本章提出的判别相关滤波网络 DCFNet 和尺度位移空间相关滤波网络 SPCNet 与典型的相关滤波器 KCF[29]、DSST[89]、HCF[30]、SRDCF[54]、CFNet[11]、MCPF[93] 以及其他领先的跟踪算法 TGPR[24]、Staple[75]、SINT[10] 和 SiamFC[7] 在 OTB-2013[3] 以及 OTB-2015[1] 上的性能对比。图2.7a以及图2.7b显示了对比算法在 OTB-2013 以及 OTB-2015 中一次运行评估（One Pass Evaluation，简称 OPE）设定下的成功曲线。本章提出的端到端的特征训练算法 DCFNet 在 OTB-2015 上相比于使用 HOG 特征的 KCF[29] 和 DSST[89] 分别取得了 10.0% 和 6.2% 的性能提升。基于端到端学习的联合尺度空间相关滤波器 SPCNet 在 OTB-

2013 以及 OTB-2015 上相比于使用 HOG 特征的 KCF[29] 和 DSST[89] 取得了超过 11% 的提升。

尽管本章设计的只包含两个卷积层的特征网络结构相比于算法 HCF[30] 要浅很多，但是由于使用了端到端的学习策略，算法性能要大幅超过使用预训练的相关滤波算法。直接使用较深的网络特征可能破坏图像的边缘纹理信息，反而造成算法性能退化。同时，相比于较好性能的 MCPF[93]，本章提出的 SPCNet 取得了超过 100 倍的速度提升，同时保持了类似的精度。MCPF 采用粒子滤波采样与相关滤波模型相结合，通过粒子滤波进行大量采样测试的方式会显著影响运行速度。由于本章采用了更加自适应的在线调整策略（详见第 §2.3.2节），我们的算法相对于孪生网络算法 SINT[10] 和 SiamFC[7] 取得了更为良好的适应性。这两种孪生网络跟踪器为了保持速度，过于激进地去除了在线学习模块，使得算法难以适应目标的表观大幅变化。相比于同样使用端到端学习的 CFNet[11]，端到端的尺度空间联合学习增加了网络对于尺度回归的准确性，使得精度性能提升超过 6%。

此外，为了验证算法的鲁棒性，我们增加使用时间鲁棒性分析（Temporal Robustness Evaluation，简称 TRE），空间鲁棒性分析（Spatial Robustness Evaluation，简称 SRE）两项鲁棒性指标。TRE 通过将视频序列切分出多个起点来构成更多的测试视频，SRE 通过在初始帧增加多种位移扰动来测试算法的抗扰动性。图2.7c和图2.7d可以看到，本章提出的算法在 TRE 和 SRE 设定下中都取得了当前最好结果，显著优于使用深度特征的 HCF[30]，这同样证明了算法通过端到端学习带来的性能提升。

2.5.4　基于数据集 VOT-2015 的评测结果分析

在 VOT-2015 挑战赛 [4] 中，我们比较了 SPCNet、EDCF 和其他 62 个参赛跟踪器在平均期望重叠率 EAO 指标下的性能，算法性能按照 EAO 降序排列。EAO 结果如图2.8所示，水平的灰色横线代表 VOT-2015 官方认可的领先算法性能，本章提出的 SPCNet 和 EDCF 都超过了该基准。根据 EAO 指标的整体表现评估，SPCNet 和 EDCF 的整体精度表现分别排在第 8 位和第 3 位。相比于其他基于相关滤波的算法（例如，DeepSRDCF[67]、NSAMF[32]、MUSTer[99] 以及 KCF[29]），本章所提出的算法取得了更好的性能与精度的平衡。NSAMF 相对于原始采用 HOG 特征进行多尺度相关分析测试的 SAMF[32] 进行改进，通过采

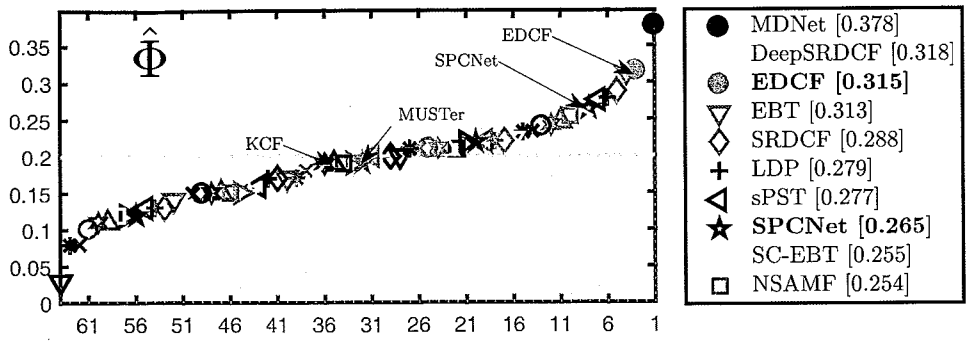


图 2.8 基于数据集 VOT-2015[4] 本章提出的 SPCNet、EDCF 与相关算法的 EAO 性能对比展示。

用颜色名称特征与边缘轮廓特征相结合的方法，多模型融合得到最终的目标估计。本章所提出的 SPCNet 算法仅使用单一深度学习特征，通过端到端学习训练，取得更好的跟踪性能。MUSTer[99] 算法采用相关滤波与关键点跟踪算法相结合，尽管关键点在目标旋转时对于目标具有较好的适应性，但是对于图像模糊以及光照变化等情况，该算法由于采用手工设计特征，并不能良好地适应这些情况。本章提出的编解码相关滤波网络 EDCF 采用相关滤波与孪生判别网络相结合的多任务学习框架，在 VOT-2015 上取得 0.315 的 EAO 值，相比于 MUSTer 在 EAO 指标下提升了 0.12。本章提出 EDCF 的跟踪精度得分与 DeepSRDCF[67] 和 MDNet[25] 相当，但运行速度快了几个数量级。这是由于本章算法通过深度特征提取与相关滤波判别模型的联合优化，增加了特征表示网络设计的自由度，因而显著降低了算法的计算存储消耗。

2.5.5 算法通用性验证

为了验证本章所提出的端到端学习的判别相关滤波网络 DCFNet 和尺度位移空间联合学习的相关滤波器网络 SPCNet 在视觉目标跟踪中对于目标形态变化以及尺度估计的显著提升。我们在 *human3* 和 *skiing* 这两个极具挑战性的视频中测试对比了本章提出的算法以及相关滤波算法 MCPF[93]、SAMF[32]、DSST[89] 和孪生网络算法 CFNet[11] 与 SiamFC[7]。

图2.9显示了算法在 *human3* 序列中的对比结果。在 *human3* 序列的第 100 帧左右，目标跟踪对象依次被电线杆以及其他具有类似表观的行人遮挡。而对于遮挡场景（Occlusion）以及背景嘈杂（Background Clutter）的场景，算法需要特征提取模块有效区别目标与周围干扰，否则极易被遮挡物或干扰对象混淆，造成错误

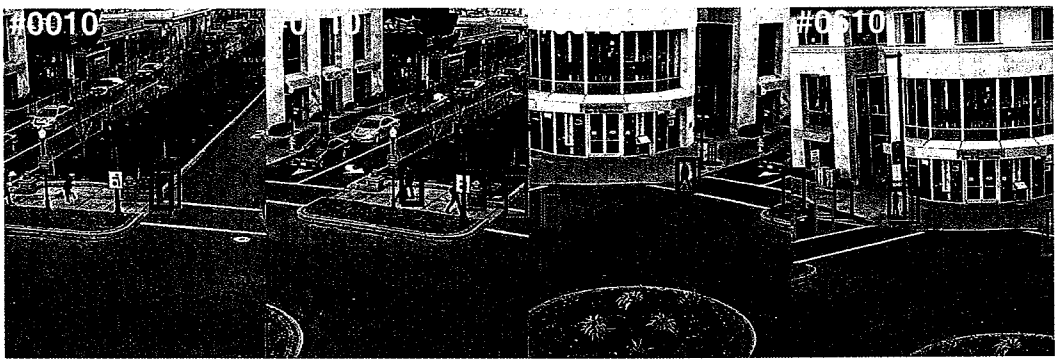


图 2.9 在视频序列 *human3* 上本章提出的 DCFNet、SPCNet 与相关算法的跟踪结果对比展示。

跟踪。SAMF[32] 由于仅采用了 HOG 特征，对于类似形态的行人缺乏足够的判别力，造成了跟踪漂移。在 300 帧的时候，只有本章提出的算法以及 MCPF[93] 可以良好跟踪。级联进行尺度估计的 DSST[89] 被干扰对象错误吸引；没有进行表观在线学习的孪生网络算法 SiamFC[7] 同样无法适应遮挡场景。在 300 帧到 600 帧之间，摄像机的焦距发生剧烈变化，目标物体在图像中的尺度发生跳变。可以看到，由于 SPCNet 采用了尺度位移空间的联合学习，对于尺度变化的适应性较强。而基于粒子滤波的判别相关学习算法 MCPF 尽管采用了更多的尺度样本采样，但是由于没有将多尺度样本进行联合学习，无法估计出目标正确的尺度大小。这些场景中取得的优势归功于算法在联合尺度-位移空间的判别学习以及端到端特征学习两个重要特性。

图2.10展示了算法在 *skiing* 视频序列中的跟踪效果，*skiing* 中不仅目标尺度（scale variation）、形态（deformation）发生了较大变化，同时低分辨（low resolution）的目标物体在相机视角中快速移动（fast motion）。由于多种困难场景叠加在同一个视频中，该视频对于视觉目标跟踪算法极具挑战。由图中第 10 帧结果可以发现，SAMF[32] 由于采用非常简单的特征表示，在跟踪的最开始阶段就无法正确跟踪目标。在图像 30 帧之后只有本章所提出的 DCFNet 以及 SPCNet 可以正确跟踪，SPCNet 在尺度发生较大变化的时候估计更加精确。这同样验证了尺度位移空间的联合学习带来的尺度估计优势。

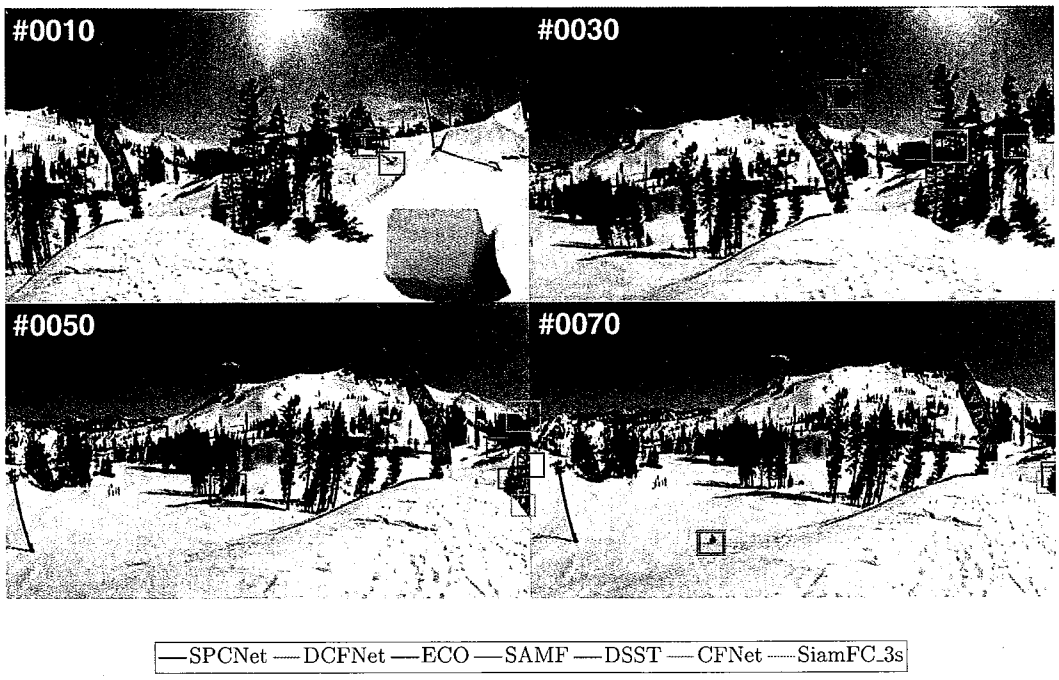


图 2.10 在视频序列 *skiing* 上本章提出的 DCFNet、SPCNet 与相关算法的跟踪结果对比展示。

2.6 本章小结

在本章中，我们主要关注于实时视觉目标跟踪的特征表示研究，提出了端到端学习的判别相关滤波网络 DCFNet 跟踪器，创新性地实现了深度特征自动提取与相关滤波判别模型的联合优化，有效提升了深度特征表示的学习能力。通过端到端的特征网络学习，我们可以定制设计特征表示网络的拓扑结构，显著降低了特征表示网络的计算时间。DCFNet 同时得益于轻量级的特征学习网络设计和相关滤波层中基于傅里叶频域的快速相关建模，具有较高的运算效率。然后，本章将 DCFNet 扩展到基于联合尺度-位置空间的相关分析网络 SPCNet，通过多尺度相关滤波学习能够对目标尺度和位置做出准确估计。在此基础上本章又探索了基于深度特征的语义嵌入模型，并提出使用编解码自监督学习孪生网络 EDCF 实现对目标及其周围环境结构信息的有效感知，提升了特征表示的泛化性能与细粒度表示能力。最后，本章通过大量的对照实验分析以及在公开数据集上的评测表明，端到端的特征学习显著提高了跟踪的鲁棒性，本章所提出的 DCFNet、SPCNet 以及 EDCF 做到了精度与速度的兼顾与平衡。

第 3 章　基于残差注意力机制的孪生网络高效目标跟踪算法研究

3.1　引言

近年来，基于相关滤波器的踪器算法 [12, 21, 29, 54, 89] 取得令人瞩目的发展。本文第 §2章提出基于端到端学习的判别相关滤波网络用以提升深度特征表示的学习能力，但其表观模型仍然保持传统线性相关滤波的在线优化学习策略，降低了跟踪算法的速度。目前，基于深度学习的目标跟踪研究 [7, 70, 71, 109] 主要于利用孪生网络学习目标对象的语义嵌入表示，算法通过离线训练来辨别两个图像块是否包含相同的物体。这类算法的核心思想是利用语义嵌入表示度量目标对象的相似度，绕过在线学习更新的步骤。然而，孪生网络跟踪算法使用固定的度量来比较目标表观会阻碍算法利用特定场景中的上下文信息增强判别能力。在本章中，我们通过重新形式化判别式目标跟踪框架，提出融合多种注意力机制学习跟踪的判别表述，增强孪生网络的在线适应性。

基于相关滤波器的跟踪算法具有较高的跟踪速度和较好的跟踪精度。其中最为典型的代表是 MOSSE 跟踪器 [21]，其运行速度达 669fps。该算法取得较高速度的主要原因是采用快速傅里叶变换的逐元素乘法代替了矩阵求逆的运算，并采用了相对简单的灰度特征。然而，对于复杂的跟踪场景，目标的表观会发生较大的光照变化、非刚体形变等，而这些变化通常在手工特征中呈现非线性的流形分布（manifold）。相关滤波器跟踪算法 [21, 51] 的表观模型本质上是线性判别模型，因而使用手工设计特征的相关滤波器的性能在复杂场景中通常会显著下降。

近年来，深度学习模型由于其模型容量大、特征表示能力强，已成为提高跟踪精度的重要方法。在 ImageNet[6] 等大规模图像分类任务上训练的高层特征可以良好地区分不同类别物体。Krizhevsky 等 [39] 在深度学习代表作 AlexNet 中通过网络隐含层中的 4096 维向量表示进行图片相似性度量，该特征可以有效检索类似表观的图像。Zeiler 和 Fergus[141] 通过反卷积可视化网络的不同层次特征表示，发现底层网络主要提取颜色、边缘轮廓等局部特征；高层网络主要提取具有判别性的全局语义特征，这些高层特征具有较强的平移、尺度和旋转不变性。同时，文献 [141] 也分析了深度神经网络对于遮挡的敏感性，图像不同空间位置

的表观对于最终特征响应的影响具有较大差异。这些实验分析极大地启发了深度学习特征在视觉目标跟踪中的应用。研究人员通过在大型数据集上对深度卷积网络进行离线训练（offline learning），并在目标序列上进行在线学习（online learning）实现目标跟踪。基于深度学习的跟踪算法在所有最近发布的基准测试集 [1, 3] 和挑战赛 [2, 4, 8, 13, 77] 中都取得了具有绝对优势的精度指标。

尽管有了这些显著的进步，但是大多数基于深度学习的跟踪方法仍然存在一些缺陷。在跟踪过程中，离线学习的深度特征通常不能很好地适应特定目标场景。同时，在跟踪过程中有时会存在密集的同类物体，预训练的网络通常难以学习到同类物体间的细粒度差异。深度学习算法要求最大化训练集中不同物体之间的差异，同时最小化同一物体在不同视角、光照、形态下的特征距离。但目标跟踪中的物体识别不仅仅由表观所决定，目标所处的上下文环境会影响目标判别。例如，人们可以利用目标上下文的信息同时跟踪两个一模一样的物体。一些基于深度学习的跟踪算法 [11, 25] 采用在线学习的方式进行上下文环境感知，但深度学习网络具有极大的模型容量，可以无误差地拟合训练样本集合，对于深度学习网络的在线学习会带来过度拟合的风险，从而难以适应目标的光照或形态变化。Bhat 等 [96] 提出通过使用多种数据增强方式来提升判别器的泛化性能与跟踪精度，但对于视觉目标跟踪算法的运行速度造成了极大的损失。特征表示提取器以及表观模型的在线学习、更新过程以及推理过程的计算开销都非常巨大，这种高密度的计算消耗使得跟踪算法无法在每一帧实时执行所有操作 [25, 66]。

基于离线训练的孪生网络目标跟踪 [7, 10] 由于其速度和精度的均衡表现，近来受到了极大的关注。孪生网络学习方法将目标跟踪形式化为验证问题，通过对比目标图像和搜索区域图像的相似度进行目标识别。文献 [7] 由于采用全卷积孪生网络结构，有效共享了不同搜索图像特征计算，达到了极高的跟踪速度。算法通过全卷积网络提取目标模板和搜索区域的表观特征，并在内积空间进行度量。然而，由于直接迁移图像分类的网络设计，对该算法的特征提取带来了不可忽视的影响。基于通用卷积神经网络的图像分类模型由于采用共享的卷积核在图像平面内移动，具有较强的平移不变性，各空间位置计算相互独立，预测结果和当前环境上下文无关。但对于视觉目标跟踪中的目标模板图像，判别信息由其上下文决定。在进行判别学习时，不同空间位置之间的表观应产生相互影响，根据当前场景调整判别属性的权重。这种根据场景自适应调整的能力是目标跟踪

与图像分类问题本质区别之一。

基于上述考虑，本章提出了残差注意力孪生网络（Residual Attentional Siamese Network，简称 RASNet）来进行高效目标跟踪。为了增加孪生网络跟踪算法的形态适应性，本章提出了带有加权的交叉相关操作算子，可以对目标不同空间位置的相关操作赋以自适应调整的权重。然后，本章提出利用注意力机制来学习目标的权重系数，并将注意力机制分解为用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行权重调整的通道注意力机制。通过注意力机制的学习，本章所提出的算法不仅减少了孪生卷积神经网络在训练中的过拟合，同时轻量化的网络设计，保证了良好的跟踪速度。最后，本章通过在 OTB[1, 3] 和 VOT[4, 8] 数据集上的大量实验结果，证明所提出的注意力机制的有效性。

本章余下的部分做如下安排：第 §3.2节介绍相关工作，并分析本章提出的残差注意力孪生网络与已有方法的异同点；第 §3.3节详细阐述残差注意力孪生网络的框架构成和跟踪流程；第 §3.4节对模型进行评估和分析，并与已有的视觉目标跟踪方法进行对比。最后，第 §3.5节对全章进行总结。

3.2　相关工作介绍

在本节中我们主要讨论最相关的跟踪方法和技术。特别是基于深度特征的跟踪方法、基于端到端网络学习的跟踪方法以及注意机制相关研究。

基于深度学习的跟踪算法：近年来，深度学习网络因其优异的表观建模能力而被广泛应用于提高跟踪算法性能。研究人员 [12, 30, 31, 67] 将深度特征与相关滤波器结合实现鲁棒的目标跟踪。文献 [30] 将来自图像识别预训练的卷积神经网络 VGGNet[59] 的不同层特征连接到相关滤波器中进行融合跟踪。文献 [67] 通过实验分析不同层特征与空间正则相关滤波器结合带来的性能差异，最终决定采用神经网络的第二层特征作为特征表示。同时，由于深度学习特征通常具有较低的分辨率，不利于精确定位，而目标跟踪问题需要在每一帧进行精确的位置估计。为了消除这一矛盾，Danelljan 等在文献 [31] 中将离散相关滤波运算推广到连续相关滤波器操作，通过将滤波操作连续化产生连续位置估计，并融合不同分辨率特征得到跟踪结果。但该算法由于采用了深度学习特征以及连续卷积操作进行精确的位置估计，运算速度相较于原始的相关滤波算法 [29] 下降达数百倍，

这使得该算法难以在实际应用中得到使用。随后，文献 [12] 通过分解卷积操作对深度学习的特征进行有效降维，去除了大量冗余运算；通过混合高斯模型来建模训练样本分布，选取少量具有代表性的样本进行训练；最后减少更新次数来降低在线更新时间消耗。尽管通过上述这一系列的加速优化，算法仍然无法在深度特征中取得实时跟踪效率。本章所提出的方法采用更加直接的更新策略，在跟踪过程中避免对目标滤波器的在线优化学习。算法着重于利用神经网络在初始帧对于目标场景进行自适应调整。同时，算法在大规模的数据集中进行特征训练，这也使得特征网络的判别能力要显著优于直接采用预训练的神经网络。

除了使用预训练好的深度学习特征与相关滤波器相结合的跟踪框架，视觉目标跟踪通常被构建为基于深度学习网络的分类或回归框架。因此，这类方法主要借鉴图像识别任务的网络设计思想进行在线训练。例如，文献 [142] 使用了 CNN 模型以及显著性图像估计，结合 SVM 分类器进行学习，其判别性能以及运行速度受到了 SVM 算法的制约。Wang 等 [66] 在全卷积神经网络回归框架中提出了结合多层次互补特征的跟踪策略。通过分析目标特征在不同通道分布的稀疏程度，提出特征选择模块减少特征优化计算。文献 [135] 将跟踪形式化为目标前景背景分类问题，并使用多个卷积神经网络构建的候选池作为目标对象的不同实例进行数据驱动学习。文献 [105] 提出利用时间和空间信息进行编码的卷积神经网络进行判别学习。这些方法的优点是使用了深度网络进行特征表示，然而，仅进行在线学习的深度学习方法无法利用离线数据集上的跟踪先验，这限制了模型特征的丰富性以及判别针对性。而上述方法对深度网络进行在线训练或更新，会显著降低跟踪速度。

基于端到端学习的跟踪算法：为了获得针对于视觉目标跟踪任务的专用网络，研究人员 [7, 10, 25, 35, 71] 在离线视频序列上端到端地训练深度学习模型，并在目标跟踪基准上对模型进行在线跟踪评估。Nam[25] 将跟踪形式化为一个分类问题，通过多分支的网络结构，在大量视频序列中离线学习共享深度特征提取器。然后在线跟踪过程中，算法通过添加一个全连接层进行在线学习。该算法在粒子滤波框架 [20] 内通过大量采样，构建训练样本集以及候选测试样本集合。算法相对于之前使用手工特征算法取得了超过 10% 的精度提升，该算法的成功启发了后续的深度学习目标跟踪算法研究。Held 等 [35] 将前一帧目标图像与当前帧搜索区域图像并联拼接，通过矩形框回归学习来表述目标跟踪状态。该方法

摒弃了基于大规模采样决策的跟踪方式，通过直接预测目标位置得到跟踪结果，成为首个实时运行的深度学习跟踪算法。但由于其离线训练过程中需要枚举所有的位移样本，训练难度极大，在线跟踪效果并不稳定。文献 [10] 将视觉目标跟踪描述为目标验证问题，并训练了孪生网络来学习用于在线目标匹配的特征表示。Bertinetto 等 [7] 将互相关（cross correlation）引入到一个全卷积的孪生网络中，取得了非常优异的跟踪效率。该算法获得 VOT-2017[8] 第一届实时目标跟踪挑战赛冠军。Valmadre 等 [11] 随后将相关滤波学习器解释为深度神经网络中的可微层，改善了孪生网络跟踪算法的在线跟踪性能。这些方法推动了端到端的深度学习跟踪框架的发展，并在最近的基准测试中取得了非常好的结果。在本章中，我们通过分析判别跟踪算法的学习架构，提出了将目标跟踪分解为具有通用语义描述的特征表示网络和具有自适应调整功能的判别网络，这种解耦方式可以有效提升网络的泛化性能与判别能力。

注意力机制与基于注意力机制的跟踪算法：注意力机制在神经科学领域首先被提出，然后扩展到其他应用领域，如图像分类 [5, 143]、行人重识别 [144]、视频行为理解 [145] 等。对于视觉目标跟踪任务，Choi 等人 [146] 提出基于注意力机制的相关滤波网络，算法维护了大量不同参数设置的判别相关滤波跟踪器，通过注意机制来选择判别相关滤波器的子集进行集成目标跟踪。文献 [147] 通过一个多向的递归神经网络将注意力吸引到目标可能存在的位置上。文献 [55] 通过使用颜色直方图来构造前景空间可靠性分布图，利用自适应的前景分割显著性区域约束相关滤波学习。与上述基于相关滤波学习的注意力机制不同，本章提出了通过端到端的深度神经网络学习注意力机制的方法。算法提出将空间注意力机制分解为离线训练数据统计得到的先验注意力和残差网络估计的残差注意力两部分组成，融合了离线训练数据集和在线实时跟踪目标的优点。

本章提出一种新的端到端的深度学习跟踪架构——残差注意力孪生网络（Residual Attentional Siamese Network，简称 RASNet），它被设计用来同时学习通用的特征表示和具有自适应性的判别器。本章提出了带有加权的交叉相关操作算子，可以对目标不同空间位置的相关操作赋以自适应调整的权重。模型通过联合学习判别相关损失以及目标区域的判别系数，实现了较强的表观形态适应性。同时，RASNet 广泛探索了不同结构的注意力机制，将离线学习的特征表示快速调整到一个特定的跟踪目标场景。算法借鉴残差网络学习思想 [61]，将空间

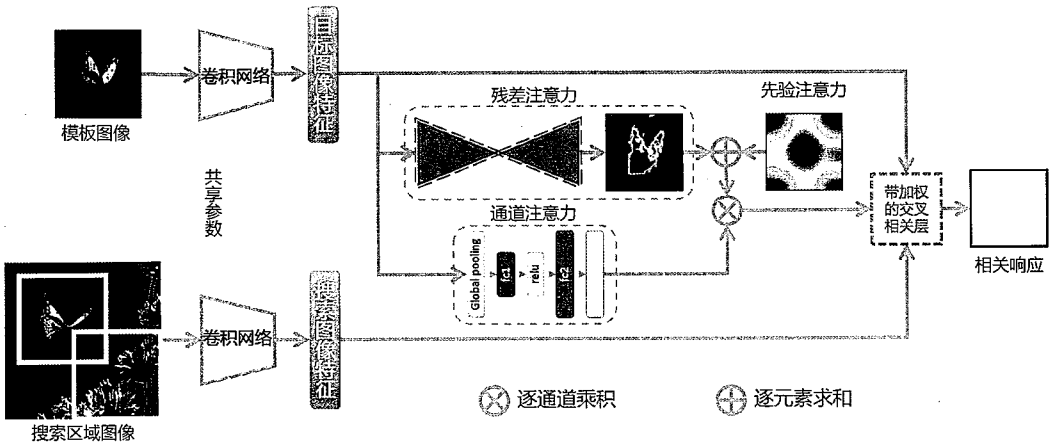


图 3.1 残差注意力孪生神经网络（Residual Attentional Siamese Network，简称 RASNet）结构示意图。

注意力机制分解为先验注意力以及具有较强个体适应性的残差注意力。RASNet 中残差注意力模块的主干采用一个类似于沙漏形状的卷积神经网络（CNN）模型 [148] 学习上下文表示和多尺度特征表示。为了保证算法具有较高的跟踪效率，这些注意力机制网络都采用轻量化的网络设计，神经网络的参数只在离线训练阶段进行训练学习。本章最后对最新的跟踪基准 [1, 3, 4, 8] 进行了广泛的分析和评估，验证了该模型的有效性和运算效率。

3.3 残差注意力孪生网络跟踪算法

为了得到高效的自适应跟踪算法，本章提出残差注意力机制孪生网络（Residual Attentional Siamese Network）。图3.1展示了本章所提出算法的网络结构示意图。RASNet 主要由一个共享的特征提取网络，注意力机制（包含先验注意力、残差注意力以及通道注意力）以及带加权的交叉相关层所构成。当一对目标模板图像以及搜索区域图像块输入到网络后，首先通过孪生网络共享的特征提取器提取图像的表观特征。基于模板图像的深度特征，算法提取三种特殊设计的注意力权重。最后将模板特征、搜索图像特征以及注意力权重输入带加权的交叉相关模块得到最终的相关响应图。不同于传统的深度学习跟踪算法，RASNet 从回归的角度重新形式化构建孪生网络跟踪器，并提出带加权的交叉相关层来端到端地学习整个孪生网络参数。如图3.1所示，带加权的交叉相关网络层利用了多种注意力机制来调整离线训练的网络特征表示以适应在线目标跟踪。

3.3.1　孪生网络简介

通常的判别目标跟踪框架可以被形式化为训练样本回归问题：

$$\min_{\mathbf{w}} \|\mathbf{A}\mathbf{w}-\mathbf{y}\|_2^2+\lambda\|\mathbf{w}\|_2^2 \tag{3.1}$$

其中矩阵 $\mathbf{A}$ 是训练样本特征集合，$\mathbf{y}$ 是相应的样本标签集合，λ 表示正则系数，$\|\cdot\|_2$ 表示向量的 L2 范数（ℓ_2 -norm）。问题的解可表述为：

$$\mathbf{w}=\left(\mathbf{A}^{\mathrm{T}}\mathbf{A}+\lambda\mathbf{I}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y}. \tag{3.2}$$

由于上式中存在矩阵的逆的计算，使得跟踪过程中直接计算公式（3.2）进行在线学习的复杂度较高。通常算法会在对偶空间进行求解：

$$\mathbf{w}=\mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}+\lambda\mathbf{I}\right)^{-1}\mathbf{y}=\mathbf{A}^{\mathrm{T}}\boldsymbol{\alpha}, \tag{3.3}$$

从公式（3.3）中我们可以看到，这种对偶形式实际上是将特征提取和判别学习相分离，利用对偶变量系数 $\boldsymbol{\alpha}$ 来反映判别分量。对于基于回归的跟踪算法，例如 KCF[29]，其本质是快速求解对偶变量 $\boldsymbol{\alpha}$。

对于基于孪生网络的跟踪算法（例如，文献 [7, 10]）而言，直接通过离线训练进行度量学习函数 $f(\mathbf{x},\mathbf{z})$，该函数用来比较模板图像 $\mathbf{x}$ 与同样大小的候选区域 $\mathbf{z}$ 的相似度。网络优化目标函数最大化目标模板 $\mathbf{x}$ 与搜索空间中的负样本 $\mathbf{z}_{i,\mathbf{y}_i=-1}$ 的距离，同时最小化目标模板 $\mathbf{x}$ 和搜索空间中的正样本 $\mathbf{z}_{j,\mathbf{y}_j=+1}$ 的距离。算法在较大的候选区域中通过卷积网络进行共享特征计算，结合交叉相关操作得到每个位置的目标响应，这种共享特征计算显著提升了运算效率。

$$f(\mathbf{x},\mathbf{z})=\phi(\mathbf{x})*\phi(\mathbf{z})+b\cdot\mathbb{1}, \tag{3.4}$$

由公式（3.4）可以看出，孪生网络跟踪算法需要利用同一个特征网络 $\phi(\cdot)$ 同时进行特征提取和判别学习。假定使用 $\phi(\mathbf{x})$ 表示训练样本 $\mathbf{x}$ 的特征，对比公式（3.3）可以推断全卷积孪生神经网络相对应的判别对偶向量为只有一个样本的全一向量，即 $\boldsymbol{\alpha}=1$。判别学习部分无法针对场景中的样本进行调整，这从理论上解释了原始孪生网络固有的缺乏场景适应性的缺陷。更重要的是，通过特征表示和判别器的耦合学习会增加模型的训练难度，我们在第 §3.4.2小节实验验证了原始全卷积孪生网络在训练过程中存在的过拟合现象。

为了克服上述孪生网络跟踪算法的适应性不足缺陷，相关滤波网络 CFNet[11] 提出使用循环移位样本在线学习，该算法通过循环矩阵来近似密集的滑动窗口样本。这种循环移位样本的近似有效地减少了滤波器求解计算量，但该操作也不可避免地引入了边缘效应 [95]，使得网络特征无法适应低分辨率的特征表示。同时由于 CFNet 非常激进的在线更新策略破坏了网络的泛化性能，造成更差的跟踪精度表现。本文第 §2章对于判别相关滤波网络在网络设计方面要优于 CFNet。在本章中我们更多地关注于网络结构设计和注意力机制的应用，提出了一个更具鲁棒性与目标适应性的孪生网络跟踪框架，通过设计使用带加权的交叉相关操作以及多种注意力机制来解耦判别学习和表观学习，最终使得神经网络学习判别系数估计过程替代现有的在线优化过程。

3.3.2 带有加权的交叉相关操作

为了克服原始孪生网络跟踪器的不足，本节提出带有加权的交叉相关操作来重构孪生网络跟踪算法。所提出的带有加权的交叉相关层具有较强的通用性，可以被嵌入到其他的孪生网络跟踪器中 [7, 10, 70]。本节首先对交叉相关操作的数学表达进行分析，发现原始交叉相关操作中每个位置对于最终的相似性度量的贡献是一致均匀的。

为了对目标不同空间位置的相关操作赋以自适应调整的权重，本章提出的注意力孪生网络架构扩展了原有的交叉相关算法。我们首先定义 $\phi(\mathbf{x}) \in \mathbb{R}^{m\times n\times d}$ 为目标图像的特征表示，$\phi(\mathbf{z}) \in \mathbb{R}^{p\times q\times d}$ 为搜索区域的特征表示，两者的相关响应输出为 $f(\phi(\mathbf{x}), \phi(\mathbf{z})) \in \mathbb{R}^{p'\times q'}$，其中 $p \geqslant m$ 且 $q \geqslant n$，输出响应大小满足 $p' = p - m + 1$，$q' = q - n + 1$。将公式（3.4）对于每个空间位置的相似性估计更精确地展开：

$$f_{p',q'} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\sum_{c=0}^{d-1} \phi_{i,j,c}(\mathbf{x})\phi_{p'+i,q'+j,c}(\mathbf{z}) + b, \tag{3.5}$$

公式（3.5）表明全卷积孪生网络 SiamFC[7] 的目标模板特征 $\phi(\mathbf{x})$ 中的每个空间位置对于最终相关响应的贡献一致。而在实际跟踪过程中，每个空间位置对于最终相似度度量的贡献并不一致，在当前场景中调整目标的判别特征权重是判别分析算法的核心，因而本章提出带有加权的交叉相关操作对目标不同空间位置的相关操作赋以自适应调整的权重。

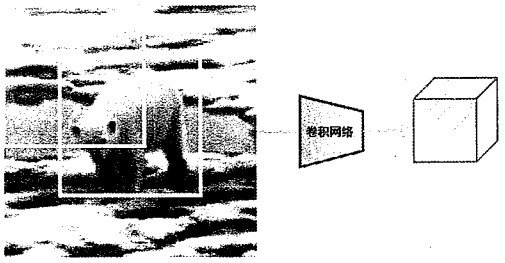


图 3.2 目标特征表示过程中特征空间位置与原始图像的位置对应关系示意图：样例图像通过卷积神经网络输出具有一定分辨率的特征表示。

如图3.2所示，目标模板图像中的蓝色矩形框内的特征在交叉相关操作运算时应比绿色矩形框区域内的特征的重要性权重更大。文献 [141] 通过反卷积操作对特征网络进行了可视化分析，实验结果也表明不同空间位置的遮挡对于最终的特征网络影响具有较大差异。因而本节提出对于不同位置的特征予以一个特殊的权重项 γ 来进行加权:

$$f_{p',q'} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\sum_{c=0}^{d-1}\gamma_{i,j,c}\phi_{i,j,c}(\mathbf{x})\phi_{p'+i,q'+j,c}(\mathbf{z}) + b, \tag{3.6}$$

公式（3.6）可以简化为下式:

$$f(\mathbf{x},\mathbf{z}) = (\gamma \odot \phi(\mathbf{x})) * \phi(\mathbf{z}) + b \cdot \mathbb{1}, \tag{3.7}$$

其中 $\odot$ 表示 Hadamard 乘积，公式（3.7）中 γ 的作用等效于目标特征的注意力，或者可以称为完全注意力。同时，与公式（3.3）中的对偶变量 α 相对应，算法通过这种注意力机制来等效判别学习。判别学习算法需要根据当前样本之间的关联关系优化求解判别系数，本章提出使用神经网络来学习注意力机制，用于替代在线判别学习。带加权的交叉相关层可以通过权重变量编码空间位置的重要性，通过不同权重来表示目标图像中的 $m \cdot n$ 个图像区域的重要性以及不同特征层的重要性。然而通过神经网络直接预测得到 γ 会引入过多的计算量以及网络参数，我们提出将整体的注意力 γ 分解为空间对偶注意力 ρ 以及通道注意力 β:

$$f_{p',q'} = \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\sum_{c=0}^{d-1}\rho_{i,j}\beta_c\phi_{i,j,c}(\mathbf{x})\phi_{p'+i,q'+j,c}(\mathbf{z}) + b. \tag{3.8}$$

这种分解设计得公式 (3.7) 需要估计的参数量从 $m \cdot n \cdot d$ 降低至 $m \cdot n + d$，分解注意力机制有助于降低训练的难度，减少了过拟合的风险。我们会在第 §3.4.2小节验证独立使用各种注意力机制的性能表现。下面我们将分别介绍空间注意力机制和通道注意力机制的构成。

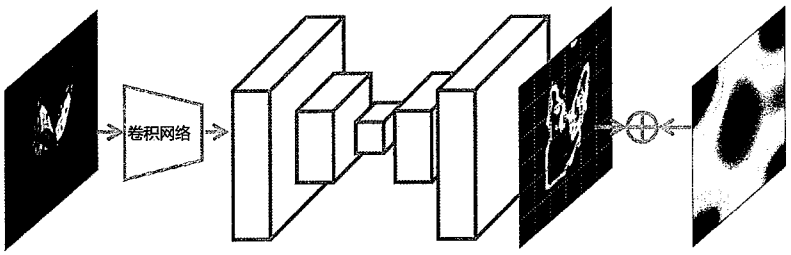


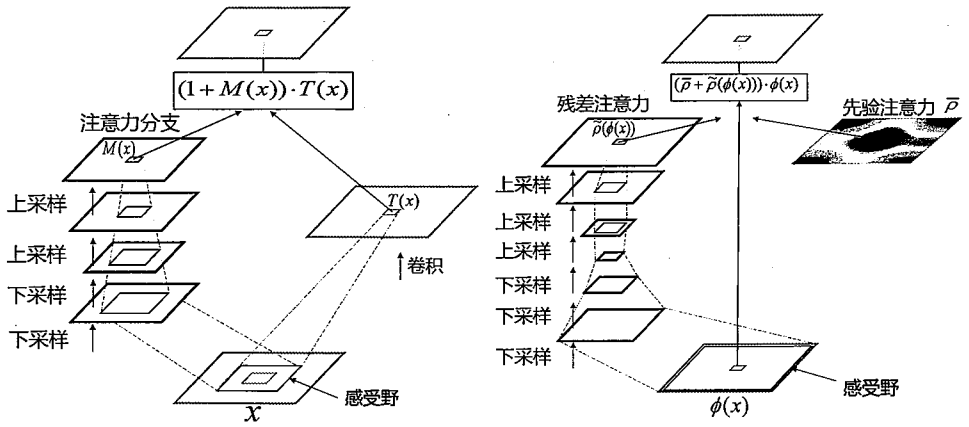
图 3.3 对偶注意力机制基本结构示意图。

3.3.3 空间对偶注意力机制

对于目标图像不同空间位置的权重调整实际上是帮助算法识别出当前场景中的关键判别信息位置。例如，在人流较多的场景中，人体的发型、服饰等细节信息对目标的跟踪较为重要；而在开阔的场景中，人的类别信息较为关键，权重应当调整到人的整体形态中。这种场景相关的判别信息描述难以通过规则定义进行学习，因而我们通过深度神经网络学习公式（3.8）中的空间对偶注意力 ρ。由于在线目标跟踪过程中缺乏足够的样本信息来完全在线训练神经网络，本小节主要关注于孪生神经网络跟踪器的离线训练过程。

一种较为直观地获取注意力机制的方式是约束所有的训练数据共享一个通用的注意力，这种注意力并不随着不同的输入样本而变化。通过统计所有样本的全局先验信息有助于我们对目标注意力分布产生初步认知。算法首先通过初始化一个大小为 $m \times n$ 的全 1 向量来训练得到先验注意力 $\bar{\rho}$。这种先验注意力机制广泛应用于目标跟踪算法中，例如，算法 SRDCF[54] 中加入了一个固定的高斯分布正则系数约束滤波器学习。我们所学习得到的先验注意力 $\overline{\rho}$ 结果类似于高斯响应（详细训练过程请参考图3.7a），这也和通常的直觉相一致。CFNet[11] 同样实验验证了离线学习一个固定的对偶变量，但是该方法由于缺乏自适应性，而且同时使用了循环矩阵来编码空间样本，引入了边缘效应 [53]。

在实际应用中，使用一个固定模式的权重分布来约束所有的训练样本以及在线跟踪的目标对象的假定过于严格。目标的空间注意力机制需要对于不同目标对象进行自适应调整，因此我们提出使用对偶注意力（dual attention）来编码目标对象的空间注意力。对偶注意力机制由一个先验注意力（prior attention）和一个残差注意力（residual attention）共同组成，其连接方式如图3.3所示。对偶注意力是先验注意力和残差注意力输出的总和，残差注意力通过编解码网络来提



(a) AttentionNet[5]　　**(b) RASNet**

图 3.4 用于图片分类的 AttentionNet[5] 算法与本章提出的 RASNet 算法的网络结构对比展示。

高目标边界附近的注意力估计。通过空间固定偏置（先验注意力）和目标自适应变化（残差注意力）进行联合学习有助于降低训练难度。同时，文献 [61] 指出通过残差学习机制可以有效消除深度神经网络训练中存在的梯度消失和梯度爆炸等问题。利用残差网络来学习适应于每个目标个体的注意力机制被我们称作对偶注意力：

$$\rho = \bar{\rho} + \tilde{\rho}, \tag{3.9}$$

先验注意力 $\bar{\rho}$ 编码了来自所有训练数据的先验统计信息，而残差注意力 $\tilde{\rho}$ 描述了不同目标各自的判别信息。不同目标对象通过残差注意力机制进行自适应的调整，因而残差注意力可以看作是判别参数估计器。对于判别器的学习，需要结合所有训练样本的全局信息。因而在网络结构设计中，本章所提出的残差注意力通过编解码结构获取了目标对象的全局信息并且保持较低的计算复杂度。

文献 [5] 同样提出了残差注意力机制网络 AttentionNet，该算法用来解决图像分类问题，因而采用了均匀的注意力作为先验注意力。而我们采用离线数据学习先验注意力，同时我们还在图3.4对比了 AttentionNet 和 RASNet 的网络结构以及感受野（receptive field）。可以看到本章所提出的残差注意力网络具有全局感受野，这对于判别跟踪的学习至关重要。总的来说，通过这种简单的注意力分解以及精细的网络设计，算法可以获得任意形式的注意力响应，允许不同输入具有特异性的注意力，使得 RASNet 跟踪器获得比一般注意力机制更好的调节性能。

3.3.4 网络通道注意力机制

卷积神经特征中的每一个通道通常对应一种特定的视觉模式[141]。因此，在特定的目标场景下，不同通道特征的权重同样需要做相应的调整。本章提出的通道注意力（channel attention）模块可以看作是为不同上下文选择语义属性的过程，这种通道特征的调整目标是保持深度网络对目标表观变化的适应能力。文献[55]在其跟踪器设计中也包含了一个通道权值，但该权重需要通过求解联合优化问题进行估计，本章提出使用深度神经网络来学习通道注意力。通道注意力权重只在目标跟踪的前向过程中进行运算，该设计相比于在线优化求解效率有明显提升。给定一组 d 个通道特征表示 $\mathbf{x}=\left[\mathbf{x}^1,\mathbf{x}^2,\ldots,\mathbf{x}^d\right]$，其中 $\mathbf{x}^i\in\mathbb{R}^{m\times n},i=1,2,\ldots,d$。网络的最终输出（记为 $\tilde{\mathbf{x}}=\left[\tilde{\mathbf{x}}^1,\tilde{\mathbf{x}}^2,\ldots,\tilde{\mathbf{x}}^d\right]$，其中 $\tilde{\mathbf{x}}_i\in\mathbb{R}^{m\times n},i=1,2,\ldots,d$）是通过执行如公式（3.10）逐通道的尺度变换求解：

$$\tilde{\mathbf{x}}^i=\boldsymbol{\beta}_i\cdot\mathbf{x}^i,\quad i=1,2,\ldots,d, \tag{3.10}$$

其中，$\boldsymbol{\beta}\in\mathbb{R}^d$ 是网络预测的通道注意力参数。

3.3.5 网络架构与离线训练

本章所提出的残差注意力孪生网络架构通过带加权的交叉相关操作整合三种注意力机制。其中，先验注意力初始化为全一向量后直接进行离线训练优化；对偶注意力机制如图3.3所示使用一种沙漏网络结构[148]，具有全局感受野；通道注意力机制通过两个全连接层的级联，可以生成任意的给定变换。具体网络架构由一个降维层与升维层级联经过 Sigmoid 激活函数变换构成。

在孪生网络的离线训练过程中，成对的训练样本通过分类损失函数进行训练。给定目标图像 $\mathbf{x}$、搜索区域图像 $\mathbf{z}$ 以及搜索区域对应的正负样本标签 $\mathbf{y}\in\{-1,+1\}$，u 表示每个和目标图像相同大小的候选框在搜索图像中的位置，$\mathcal{U}$ 表示搜索图像中的独立样本集合。训练过程采用的 Logistic 损失函数记为：

$$\mathcal{L}(\mathbf{x},\mathbf{z})=\frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\log(1+\exp(-\mathbf{y}[u]\cdot f(\mathbf{x},\mathbf{z})[u])), \tag{3.11}$$

在离线过程中需要成对的目标图像与搜索图像进行训练，因而训练样本对的选择对于相似性度量网络的判别能力至关重要。假定第 v 个视频序列的第 t 帧目标模板记作 $\{\mathbf{x}_t\}_v$，搜索区域记作 $\{\mathbf{z}_t\}_v$。由于视频帧编码图像沿时间轴的变化，通常较为邻近的图像帧表观较为相似，通过比对没有任何变化的图像帧难以让网

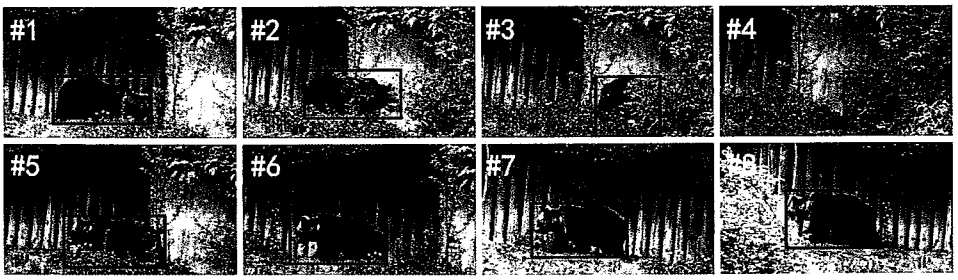


图 3.5 数据集 ILSVRC VID[6] 的一个视频序列的样本选择策略示例。其中，蓝色矩形框中的目标处于完全遮挡状态。

络学习具有判别力的表观特征。而变化差异较大的长时间间隔样本对发生遮挡的风险显著增加，导致网络学习到噪声标签，影响网络的泛化性能。我们在每一组训练样本中选定的样本对需要来自不同帧图像，同时我们提出降低较远帧对于损失函数的影响，设计损失函数：

$$\begin{aligned} \mathcal{L}_{all} &= \sum_i \sum_j \mathcal{L}\left(\mathbf{x}^i, \mathbf{z}^j\right) \cdot \Omega(i,j) \\ \text{s.t.}\quad \Omega(i,j) &= \begin{cases} 0, & i = j \\ \exp\left(-\frac{|i-j|}{\sigma}\right), & i \neq j \end{cases} \end{aligned} \tag{3.12}$$

其中，我们增加一个距离权重方程 $\Omega(i,j)$ 来表示时序样本对的可靠性。基线算法 SiamFC 采用在最近的 100 帧内随机采样的策略，该策略可以等价为使用阶梯函数作为加权方程。本节所提出的损失函数使得较为邻近帧的样本对产生更大的贡献，这可以有效降低较远帧产生的遮挡问题。如图3.5所示，本节提出的损失函数更倾向于选择样本对 (#3, #4) 以及样本对 (#5, #6)，而给予 (#1, #4) 较低的训练权重，这种选择策略有助于网络降低学习噪声。

在网络离线学习阶段，我们使用 ILSVRC VID[6] 所构成的 300 万个样本对。对于一个目标模板帧，我们同样只采样最近的 100 帧图像。对视频序列以及目标模板帧和搜索图像帧采用随机选择策略。训练样本对由一个模板图像 $\mathbf{x}$ 和一个搜索区域图像块 $\mathbf{z}$，以及搜索区域理想响应 $\mathbf{y}$ 所组成。目标模板和搜索区域首先被输入到它们各自的孪生网络特征提取分支以获得特征图 $\phi(\mathbf{x})$ 和 $\phi(\mathbf{z})$。模板特征图 $\phi(\mathbf{x})$ 同时进入对偶注意力网络和通道注意力网络得到注意力权重 ρ 与 β。通道注意力网络描述了特征表示通道之间的重要性分布，模板特征在平面空间与通道上进行注意力叠加（见公式（3.8））。该操作由加权交叉相关层实现，并生成响应映射 $f(\mathbf{x}, \mathbf{z})$。损失函数按照公式（3.12）进行加权求和。

表 3.1 本章提出的残差注意力孪生网络的主干网络、残差注意力网络、通道注意力网络的结构参数展示。其中 *conv*、*dconv*、*pool*、*fc* 和 *sigmoid* 分别代表卷积层、反卷积层、池化层、全连接层和 Sigmoid 变换层。

网络层	模板图像	搜索图像	通道数
	127×127	255×255	$\times3$
$conv(11,96,2)$	59×59	123×123	$\times96$
$pool(3,2)$	29×29	61×61	$\times96$
$conv(5,256,1)$	25×25	57×57	$\times256$
$pool(3,2)$	12×12	28×28	$\times256$
$conv(3,384,1)$	10×10	26×26	$\times384$
$conv(3,384,1)$	8×8	24×24	$\times384$
$conv(3,256,1)$	6×6	22×22	$\times256$

网络层	$\phi(\mathbf{x})$	通道数
	6×6	$\times256$
$conv(3,256,1)$	4×4	$\times256$
$conv(3,356,1)$	2×2	$\times256$
$conv(2,256,1)$	1×1	$\times256$
$dconv(2,256,1)$	2×2	$\times256$
$dconv(3,256,1)$	4×4	$\times256$
$dconv(3,1,1)$	6×6	$\times1$

网络层	$\phi(\mathbf{x})$	通道数
	6×6	$\times256$
$globalpool$	1×1	$\times256$
$fc(64)$	1×1	$\times64$
$fc(256)$	1×1	$\times256$
$sigmoid$	1×1	$\times256$

在在线跟踪阶段，算法只在目标模板第一帧图像上进行注意力机制推理，避免了在线漂移问题，这同时使得跟踪器具有较高的运行速度。第一个帧的权重调整改变了相似性度量权重分配，是针对特定目标的特殊适应。为了进行目标的尺度估计，RASNet 输入第一帧的目标模板图像 $\mathbf{x}$ 和三个不同缩放比例的搜索区域图像 $\mathbf{z}^s$ 组成的样本对，并生成三个尺度的响应估计 $\mathbf{y}^s$。目标尺度和目标位置是通过寻找响应 $\mathbf{y}^s$ 中最大响应位置来获得的。

3.4 实验评估与分析

3.4.1 实验设置

网络架构设计：本章所提出的残差注意力孪生网络主要由三个子网络组成，分别是用于共享特征提取的主干网络、用于对空间位置进行动态调整的残差注意力网络以及用于对网络通道进行权重调整的通道注意力网络。它们的网络结构参数如表3.1所示，网络主干仅包含 5 个卷积层，可以快速将模板图像和搜索区域图像的分辨率下降到较低尺度，以提升计算效率。

训练数据：为了提高特征表示的泛化能力和判别力，同时避免在稀缺的跟踪数据上过度拟合，离线过程在大规模视觉识别挑战赛（ImageNet Large Scale Visual Recognition Challenge，简称 ILSVRC）[6] 的训练数据集中训练网络参数。该数据集包含超过 4000 个视频序列以及约 130 万个矩形框标注，具有较为丰富多样的视频场景分布。该数据集被广泛使用在最近提出的孪生网络跟踪算法 [7, 70] 训练中。

优化器参数设定：本章使用动量为 0.9 的随机梯度下降（SGD）优化器从随机初始化的网络参数开始训练，并将权值衰减参数设置为 0.0005。学习率从 10^{-2} 指数衰减到 10^{-5}。该模型经过 50 个迭代周期的训练，每个批量大小为 8 个样本对。公式（3.12）中的损失函数权重参数 σ 设置为 100。

跟踪参数设定：为了适应目标尺度的变化，算法采用多个尺度的搜索图像进行测试，尺度系数设定为 $\left\{q^s | q=1.03, s=\left\lfloor -\frac{S-1}{2} \right\rfloor, \left\lfloor -\frac{S-3}{2} \right\rfloor, \ldots, \left\lfloor \frac{S-1}{2} \right\rfloor, S=3\right\}$。跟踪器在当前帧的目标尺度是由一个比例因子为 0.56 的线性插值函数在新预测的尺度上进行平滑估计。

本章所提出的残差注意力孪生网络跟踪器使用 MatConvNet[137] 在 MATLAB 平台上实现，所有实验在装配有 Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz 和 NVIDIA TITAN Xp GPU 的工作站上执行。

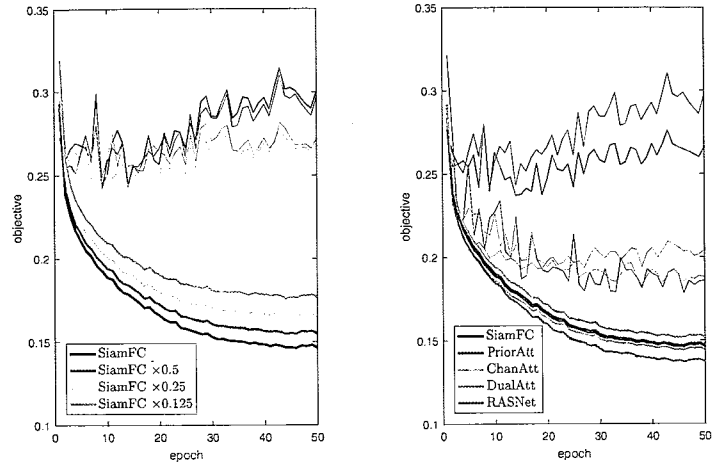
3.4.2　创新点有效性验证

首先本节分析了所提出的残差注意力孪生网络算法在 ILSVRC VID[6] 上的训练过程。SiamFC[7] 将作为本节的基线算法，训练曲线通过使用带有默认参数设置的公开代码[1]重新进行训练绘制。

图3.6表述了算法在训练集和验证集的损失函数与训练迭代次数的关系，可以观察发现基线算法 SiamFC 在训练集的损失函数逐渐下降收敛，但在验证集上的损失函数很快（大约在第 15 个 epoch 后）就有明显增加，相比于训练集上的损失函数曲线有着明显的差异，这是一种明显的过拟合现象。通常，当网络可学习参数量下降的时候，网络的过拟合现象会有所减弱。因此我们实验验证了三种轻量化的 SiamFC 网络，它们分别将网络通道数降低为 0.5 倍、0.25 倍以及 0.125 倍。但是我们观察到和原始 SiamFC 类似的结果，降低模型大小的跟踪网络在验证集上的损失函数依然会产生过拟合现象。我们认为造成这种现象的主要原因是特征表示和判别学习的同步进行，使得孪生网络无法学习到具有泛化性能的特征表示。相对于原始的全卷积孪生网络 SiamFC，本章提出利用注意力机制和带加权的交叉相关层解耦了表示学习和判别学习。

为了验证 RASNet 网络的每一个部件的效果，我们首先分析了先验注意力 $\overline{\rho}$ 的学习过程，先验注意力 $\overline{\rho}$ 由空间分辨率大小为 6×6 的单位矩阵进行初始化。

[1] https://github.com/bertinetto/siamese-fc

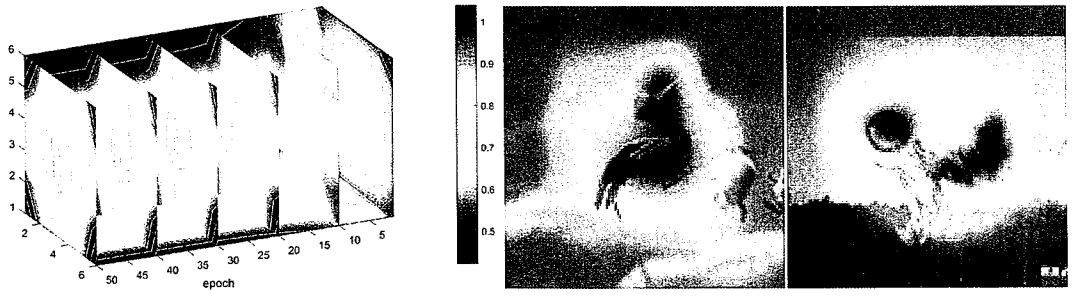


(a) SiamFC 及不同变种　　(b) RASNet 及不同变种

图 3.6 基于数据集 ILSVRC VID[6] 的网络训练过程对比展示：(a) SiamFC 及其轻量化变种的训练和验证损失函数曲线；(b) 与 SiamFC 对照，本章提出的先验注意力 (PriorAtt)、对偶注意力 (DualAtt)、通道注意力 (ChanAtt)、以及完整的 RASNet 跟踪器在训练集和验证集的损失函数曲线。其中，粗线表示训练集目标函数曲线，细线表示验证集目标函数曲线。

如图3.7a所示，可以发现随着训练的进行，先验注意力的权值逐渐聚集到矩阵的中心处。先验注意力权重的分布类似于高斯分布，其中心的权重显著大于边缘位置的权重。在图3.6b中，相对于 SiamFC 而言，先验注意力机制的过拟合现象已有所改善，但其验证集损失函数仍然保持上升。我们推测这种现象是由于判别网络的训练需要和特定的目标场景关联，而先验注意力机制不具有这种调整能力。

算法引入残差注意力来强化先验注意力，我们称这种组合为对偶注意力模型（详见第 §3.3.3小节）。图3.7b展示了对偶注意力在一些目标图像中的注意力预测结果，可以看到权重在目标前景的位置有较高的响应。与先验注意力相比，对



(a) 先验注意力学习过程可视化结果　　(b) 对偶注意力模型的预测结果

图 3.7 先验注意力学习过程与对偶注意力模型预测结果的可视化示例。

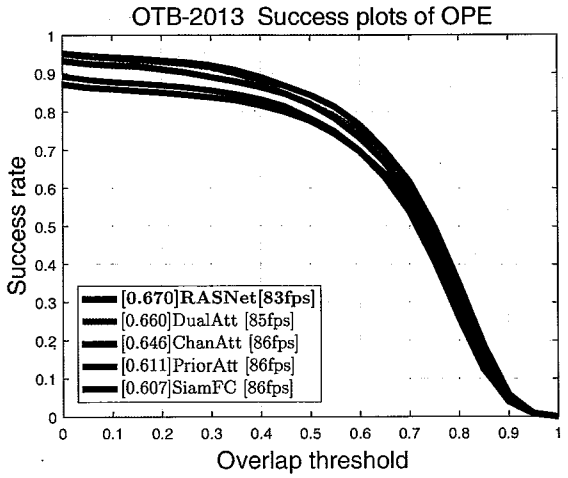
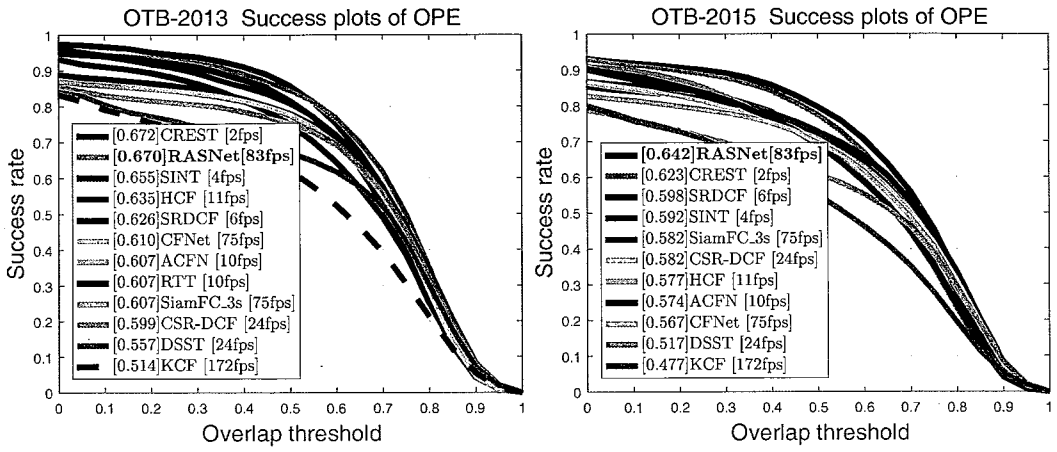


图 3.8 基于数据集 OTB-2013[3] 四个对照跟踪器（PriorAtt、DualAtt、ChanAtt 和 RASNet）和基线跟踪器 SiamFC 的跟踪结果对比展示。其中，SiamFC[7] 作为基准，PriorAtt 表示仅使用先验注意力机制的模型，ChanAtt 表示仅使用通道注意力的模型，DualAtt 表示仅使用对偶注意力的模型，RASNet 表示完整的残差注意力模型。

偶注意力的目标函数曲线更加合理，特征表示网络更专注于通用特征表示学习，因而减少了对于训练数据的类别偏见（bias）。在此基础上，我们构造了只包含通道注意分支的孪生网络模型（ChanAtt），该模型在验证集上的目标函数曲线也很快收敛。最后，我们提出的三种注意力机制共同组成的残差注意力孪生网络 RASNet 获得了最优的验证集损失。

除了通过训练损失函数曲线的验证，本节实验分析了每个组件对整体跟踪性能的贡献。四个对照跟踪器（PriorAtt、DualAtt、ChanAtt 和 RASNet ）和基线跟踪器 SiamFC 通过在 OTB-2013[3] 数据集上的成功图评分 AUC 进行评估，结果如图3.8所示。与 SiamFC 相比，先验注意力模型 PriorAtt 只增加了一个具有 36 个浮点参数的注意力常量，而 AUC 分数度量的性能提升 0.4%。在此基础上，由于考虑了目标场景的自适应判别，对偶注意力模型 DualAtt 的 AUC 评分比先验注意力模型提高了 4.9%，显著提高了跟踪精度。另一方面，通道注意力 ChanAtt 在基线的基础上提高了近 4% 的性能，该提升同样来自于对目标表观的自适应。如果算法将通道注意力简化为二进制版本，可以将其视为 [12, 66] 中使用的特征选择器。与 SiamFC 相比，整体的 RASNet 算法在 AUC 评分上获得了 6.3% 的性能提升，证明了注意机制在实际跟踪中的有效性。

在跟踪速度方面，由于本章提出的算法仅增加了非常轻量化的分支结构，并



(a) OTB-2013 的 OPE **(b) OTB-2015 的 OPE**

图 3.9 基于数据集 OTB-2013[3]、OTB-2015[1] 本章提出的 RASNet 算法与其他领先算法的 OPE 性能对比展示。

且各分支共享主干网络的计算结果，因而速度并没有过多下降。本章提出的完整模型 RASNet 相比于 SiamFC 仅下降了 3fps，但跟踪精度取得显著提升。

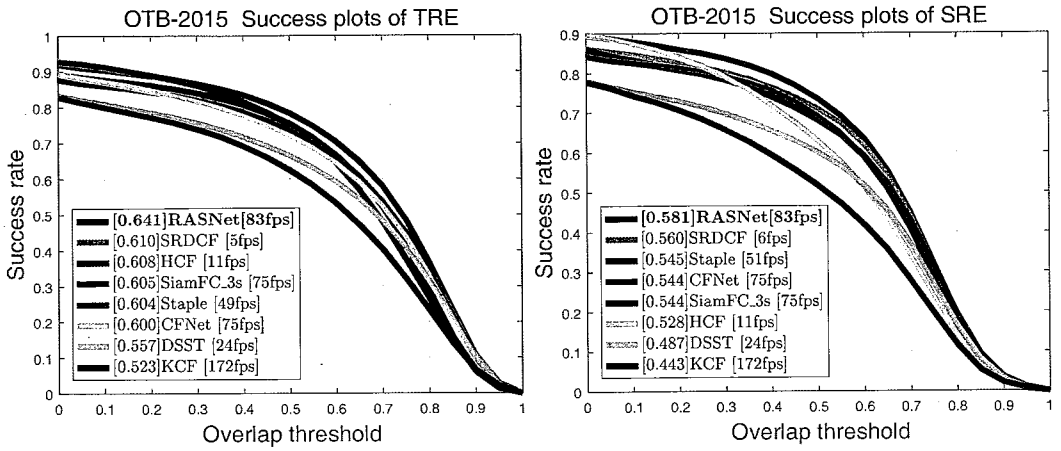
3.4.3 基于数据集 OTB-2013 和 OTB-2015 的评测结果分析

OTB-2013 数据集 [3] 是视觉目标跟踪领域最广泛使用的公开测试集，由 50 个完整标注的视频序列组成。OTB-2015 数据集 [1] 扩展了 OTB-2013 的视频序列，在该跟踪基准中包含 100 个目标对象。我们通过与许多最先进的跟踪器（包括 CREST[9]、CFNet[11]、SiamFC[7]、SINT[10]、ACFN[146]、CSR-DCF[55]、RTT[147]、HCF[30]、SRDCF[54]、KCF[29] 和 DSST[89]）的比较来评估所提出的算法的有效性。其中 CFNet、SiamFC 和 SINT 是最新的基于孪生网络的跟踪算法，CSR-DCF、RTT 和 ACFN 都采用了注意力机制用于目标跟踪。在第一帧中，所有跟踪器都使用真实标签进行初始化，并通过重叠率成功图的曲线下面积（AUC）指标进行算法性能排序。

图3.9 展示了各算法在 OTB-2013[3] 和 OTB-2015[1] 数据集上成功曲线的 AUC 以及运行速度（fps）。在 OTB-2013 的结果中，本章所提出的 RASNet 跟踪器在超实时运行速度（83fps）下的 AUC 分数达到 67.0%。CREST[9] 跟踪器在所有对比算法中表现最好，该算法通过对卷积层优化来等价判别相关滤波器的学习过程，在线收集训练样本对网络参数进行优化。同时该算法也提出了残差学习机制，通过残差网络学习滤波器在时间和空间的残差变化。但是，CREST 算

法跟踪过程中需要在线学习更新网络，运行速度仅为 2fps。本章提出的残差注意力孪生网络算法 RASNet 在精度指标上取得了与 CREST 较为接近的性能，但是速度相比于 CREST 提升了近 20 倍，这是由于算法充分利用了离线学习来增强特征的表示，降低了在线训练的时间消耗。采用注意力机制的相关滤波网络 ACFN[146] 在 OTB-2013 数据集上的 AUC 评分为 60.7%，该算法同时使用 260 个不同参数的跟踪器构成候选集合，通过注意机制选择跟踪器用于判别决策。因而该算法效率较低，尽管设置了不同的学习参数，但是单个算法的判别性能都相对较弱，造成了大量冗余低效运算。另外两个采用注意力机制与相关滤波器结合的跟踪算法 RTT[147] 和 CSR-DCF[55] 的 AUC 评分分别为 60.7% 和 59.9%。它们都利用显著性估计结合手工设计的特征来调整相关滤波器学习。这些注意力机制没有进行充分的训练学习，算法难以通过手工设计的注意力实现较为理想的目标适应性。本章提出的注意力机制与特征表示联合优化训练，取得了极大的跟踪精度提升。相比于使用预训练深度学习特征与相关滤波相结合的 HCF[30]，本章所提出的算法在 OTB-2013 上取得了 3.5% 的性能提升，这再次验证了基于端到端学习的跟踪算法的有效性。同时本章提出的 RASNet 算法采用的网络架构要远小于 HCF 算法中的 VGGNet[59]，因而在跟踪速度上取得了近 8 倍的提升。

在更大规模的 OTB-2015 的评测结果中，本章提出的孪生注意力网络 RASNet 取得了最好的 AUC 精度性能，相比于第二好的跟踪器 CREST 在 AUC 指标下提升了 1.9%。同时，我们的运行速度比 CREST 快了一个数量级。CREST 和本章所提出的算法都使用了残差学习机制，但是 CREST 算法采用过于激进的在线学习策略，在具有挑战性的视频中容易造成跟踪漂移，反而不利于算法的鲁棒跟踪。在使用孪生网络的跟踪器中，本章算法性能优于 SINT[10]，在 AUC 分数指标中提高了 5%。此外，基于粒子滤波的采样方式显著降低了 SINT 的运算速度，本章算法采用滑动窗口采样结，合全卷积孪生网络实现了高效运行。具有实时跟踪速度的 SiamFC[7] 算法是一个开创性的跟踪框架，但其跟踪精度仍然落后于传统的相关滤波跟踪算法。其改进版 CFNet[11] 增加了一个可微分的相关滤波层进行时序特征学习，但获得的性能增益较为有限。与 SiamFC 和 CFNet 相比，本章提出将注意力机制整合到孪生网络学习过程中，将算法在 OTB-2015 上的 AUC 分数提升到 64.2%，从而使 AUC 得分的增益分别达到 6% 和 7.5%。这一显著提升归功于网络的解耦设计以及多种注意力机制的高效协作。



(a) OTB-2015 的 TRE　　**(b) OTB-2015 的 SRE**

图 3.10 基于数据集 OTB-2015[1] 本章提出的 RASNet 算法与其他领先算法的 TRE、SRE 性能对比展示。

除了图3.9 所示的一次运行评估（One Pass Evaluation，简称 OPE）外，本节还报告了图3.10a 所示的时间鲁棒性评估（Temporal Robustness Evaluation，简称 TRE）和图3.10b 所示的空间鲁棒性评估（Spatial Robustness Evaluation，简称 SRE），用以检验残差注意力孪生网络 RASNet 对时间和空间初始化的鲁棒性。本章所提出的 RASNet 跟踪器在 TRE 和 SRE 曲线上获得了最好的跟踪性能，这表明我们的方法对不同的时空初始化具有较强的鲁棒性。

3.4.4　基于数据集 VOT-2015 和 VOT-2017 的评测结果分析

在本节中，将使用视觉目标跟踪工具包[2]的最新版本进行实验对比。该工具包采用基于目标重置的跟踪性能评估方法。评估算法一旦检测到跟踪器发生错误跟踪（即跟踪器预测和真实目标标签的重叠率为 0），就会在错误帧后的第五帧重新初始化跟踪器。测评性能使用期望平均重叠率（EAO）来度量排序，EAO 指标同时定量反映算法的鲁棒性和准确性。我们在 VOT-2015[4] 和 VOT-2017[8] 数据集上与当前领先的跟踪器进行比较。此外，VOT-2017 挑战赛还新增了一个实时（real-time）目标跟踪实验。

本章所提出的残差注意力孪生网络 RASNet 与其他 62 个最先进的跟踪器在 VOT-2015 数据集上评估的 EAO 曲线如图3.11所示。RASNet 跟踪器的 EAO 得分与最先进的跟踪算法相当，取得了第二好的性能，其 EAO 分数为 0.327。在

[2]https://github.com/votchallenge/vot-toolkit

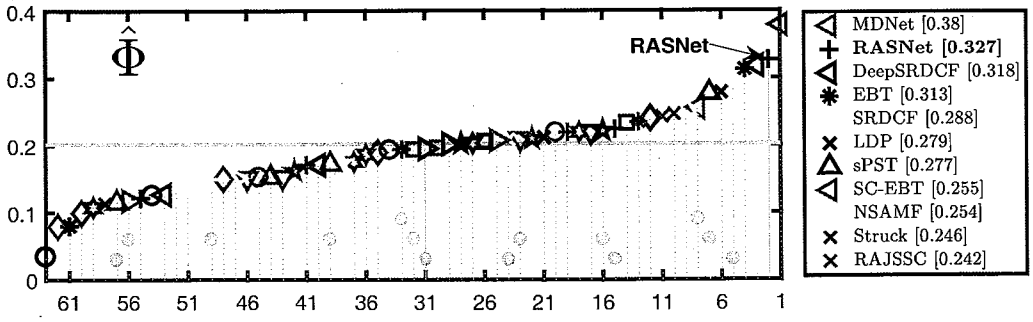


图 3.11 数据集 VOT-2015[4] 中算法的期望平均重叠率（EAO）排序图。

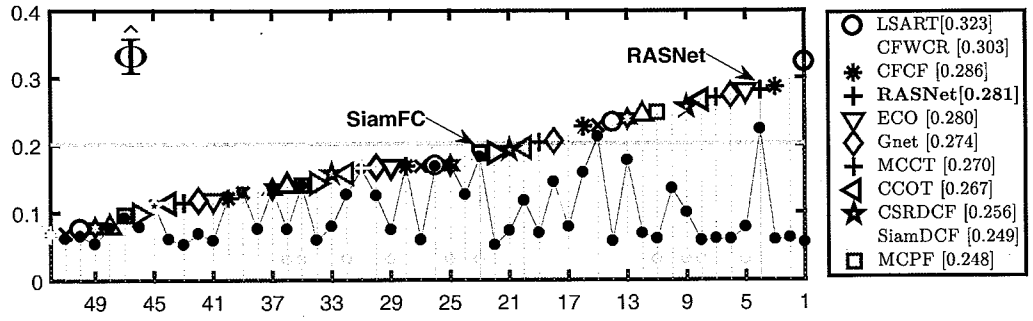


图 3.12 数据集 VOT-2017[8] 中算法的期望平均重叠率（EAO）排序图。

VOT-2015 数据集中最好的跟踪算法 MDNet[25] 使用 OTB 跟踪基准进行训练，而本章所提出的跟踪器在离线训练中没有使用任何跟踪数据集。此外，由于我们所提出的孪生网络跟踪器不需要在线训练优化神经网络参数，使得算法在跟踪速度方面比 MDNet 快了近 80 倍。相比于使用深度学习特征进行空间正则判别相关滤波学习的 DeepSRDCF[67]，本章提出的算法利用了端到端联合优化的优势，取得了更加优异的跟踪性能。在对于空间位置注意力先验的选择上，DeepSRDCF 采用手工设计的高斯分布进行正则约束，这种约束和本章所提出的先验注意力机制相类似。但由于没有参考目标样本进行调整，无法适应不同的目标形态变化。本章提出的算法通过残差注意力网络 RASNet 对不同的目标输入进行自适应预测，增加了算法对于目标形态的适应性。相比于使用 HOG 特征的 SRDCF[54] 以及混合颜色特征与 HOG 特征的 NSAMF[32]，本章算法通过神经网络预测判别系数，避免了复杂的在线优化过程，同时取得了更高的跟踪精度与速度。本章的算法在 EAO 指标下相比于 NSAMF 的相对提升超过 28%。

对于 VOT-2017 的评估中，图3.12报告了我们与其他 51 个最先进的跟踪器在 EAO 指标下的比较结果。本章所提出的 RASNet 在总体的 EAO 指标下排在第四位。在性能优于我们的三个跟踪器中，CFCF[97] 和 CFWCR[149] 采用连续

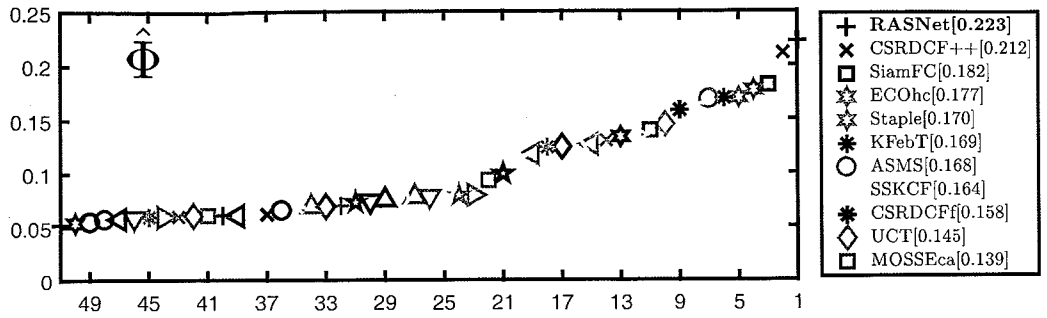


图 3.13 数据集 VOT-2017[8] 中算法进行实时跟踪的期望平均重叠率（EAO）排序图。

卷积算子作为基线方法，这类方法由于采用连续卷积操作，可以有效结合多种分辨率的网络特征，在定位精度和鲁棒性方面具有较大的优势。最优性能算法 LSART[150] 结合核回归与卷积神经网络表示。上述两类算法在线过程都需要大量的迭代优化，运行速度无法满足实时运行需求，本章所提出的算法在速度和精度方面具有较好的平衡性。相比于孪生网络跟踪算法 SiamFC[7]，本章提出的注意力机制以及带加权的交叉相关操作带来 9.3% 的性能提升。

本节进一步在 VOT-2017[8] 中评测了算法的实时跟踪性能，实时跟踪实验模拟了跟踪器在线处理连续运行的传感器的真实使用过程。如图3.13所示，本章所提出的残差注意力孪生网络跟踪算法 RASNet 在实时性能评估中排名第一，取得了 0.223 的 EAO。相比于 VOT-2017 实时跟踪挑战赛中取得最好性能的 CSRDCF++[55]，本章提出的算法将 EAO 指标提升了 1.1%。CSRDCF++ 通过多种优化方式将 CSRDCF 移植到 C++ 实现，将其性能从 0.158 提升至 0.212。本章所提出的方法由于在线推理过程更加简洁，具有更高的在线运行效率。

上述实验结果验证了本章所提出的残差注意力孪生网络 RASNet 具有较快的处理速度和优异的跟踪精度，具有实际跟踪应用的潜力。

3.4.5 基于 OTB 数据集的不同属性评测结果分析

为了更加深入地分析本章所提出的注意力机制对于视觉目标跟踪精度性能提升的来源，我们对于 OTB-2013 和 OTB-2015 中的不同属性场景下的性能进行统计，并与其他算法进行全面对比。在图3.14和图3.15中分别展示了算法在 OTB-2013 与 OTB-2015 所标注的 11 种具有代表性的属性场景下的评测结果，场景属性具体包括：水平旋转、垂直旋转、低分辨率、目标出画面、遮挡、背景混杂、非刚性形变、光照变化、尺度变化、快速运动和运动模糊。

由于本章算法采用全卷积孪生网络架构进行离线训练，因而可以获得具有高层语义描述的特征表示，对于目标的光照、旋转具有较强的适应性。在 OTB-2013 数据集中，对比于整体性能较强的 CREST[9]，RASNet 算法在水平旋转场景中有 1.6% 的提升。而对于图片信息更加模糊的低分辨率场景，基于端到端学习的特征同样具有较强的适应性。本章所提出的 RASNet 算法与基线算法 SiamFC[7] 都显著优于其他类算法。RASNet 相较于本章的基线算法 SiamFC，在低分辨率场景中有 8.7% 的提升。但是当目标出现了非刚性形变和运动模糊时，孪生网络算法的性能排名出现显著下降。这是由于我们只对目标图像的第一帧进行了注意力调整，而没有在后续图像中进行进一步的更新，使得算法的长时适应性有所下降。本章提出的残差注意力网络在 OTB-2015 中所有的挑战场景中都取得了前两名的成绩。算法相比于 CREST 在水平旋转场景提升达到 3.7%，在低分辨率场景中相比于第二名 CFNet 提升达到 6.6%。

整体来讲，在 OTB 数据集标注的所有属性场景下，本章所提出的基于残差注意力的孪生网络都带来了显著提升。

3.4.6　算法通用性验证

为了直观地对比本章所提出的残差注意力孪生网络与其他领先算法的性能，图3.16显示了 RASNet 与其他算法的跟踪结果对比，对比算法包括 CREST[9]、SINT[10]、SiamFC[7] 和 CFNet[11]。图中展示的视频包含多种具有挑战性的场景。与当时领先的跟踪算法相比，本章所提出的 RASNet 算法在目标的跟踪精度以及尺度估计方面都取得了较为显著的提升。在 *jump* 和 *motorRolling* 视频序列中，目标的表观发生了较大的旋转变化，本章所提出的算法由于使用注意力机制进行调节，较好地适应这种形态变化。在低光照嘈杂背景的 *ironman* 与 *matrix* 视频序列中，只有本章提出的残差注意力孪生网络算法成功跟踪目标。其他对比的算法尽管同样采用深度学习方法，但对于嘈杂环境的适应性要显著低于本章所提出的跟踪算法。

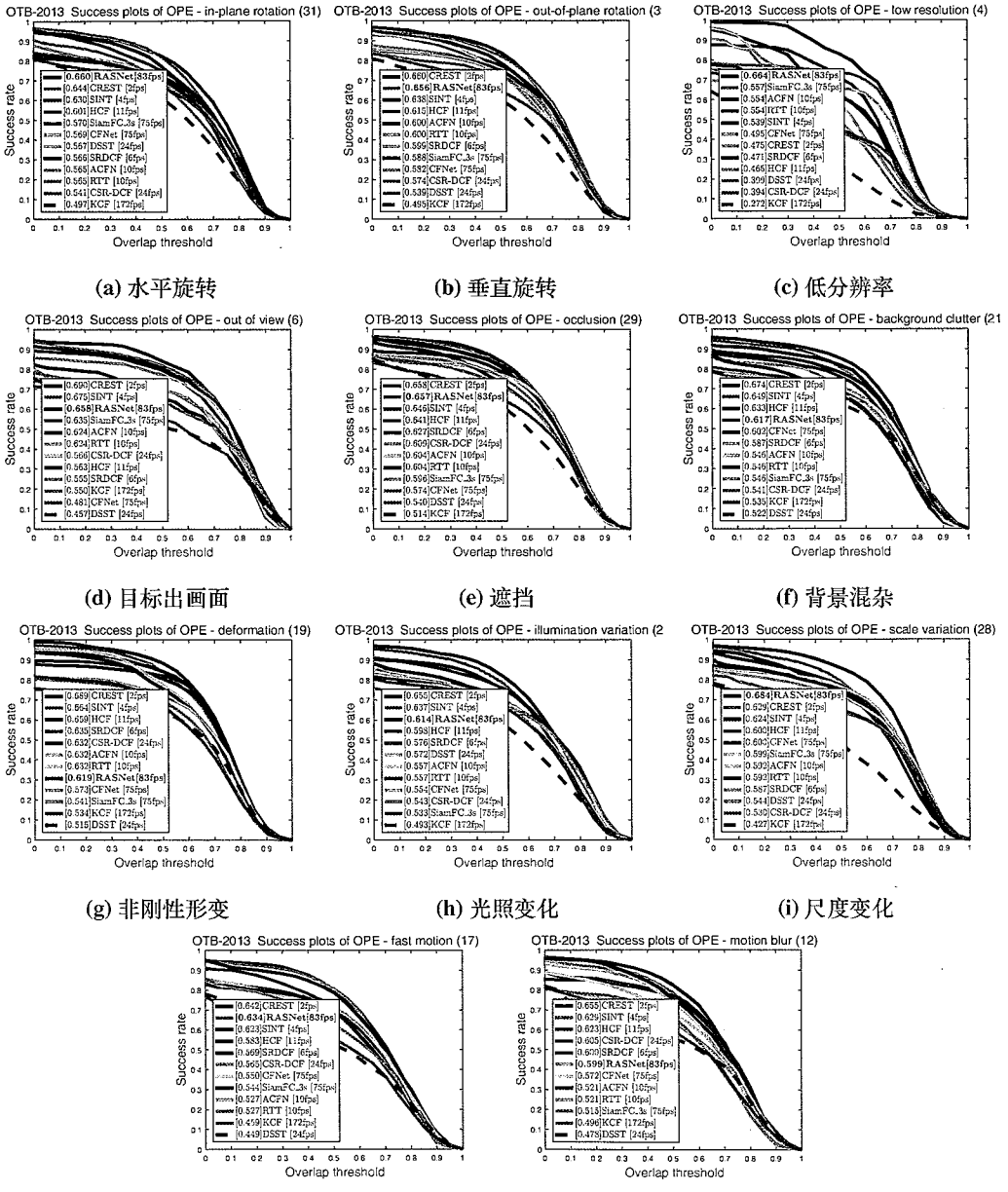


图 3.14 基于数据集 OTB-2013[3] 的 11 种属性标注的算法跟踪性能对比展示。

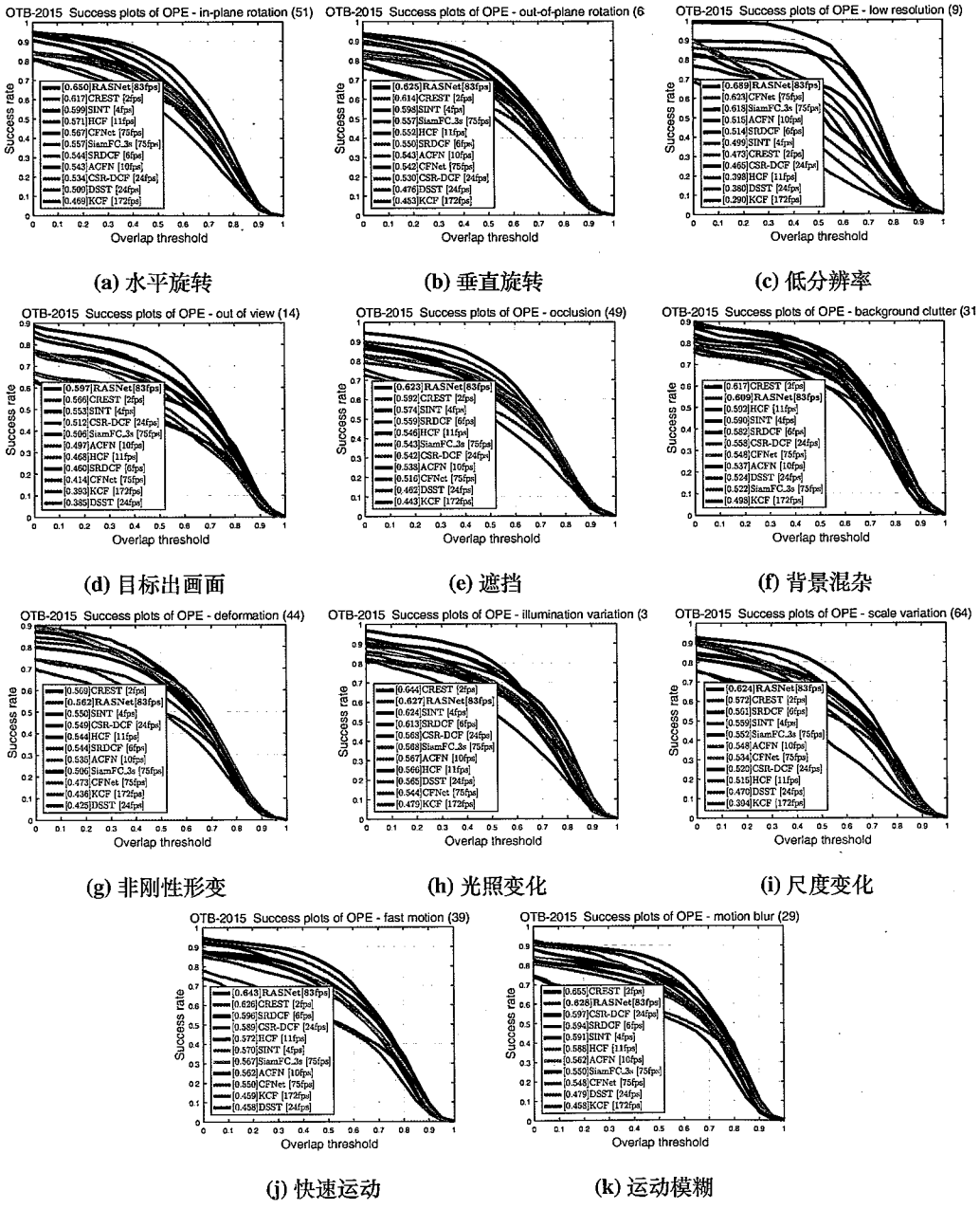
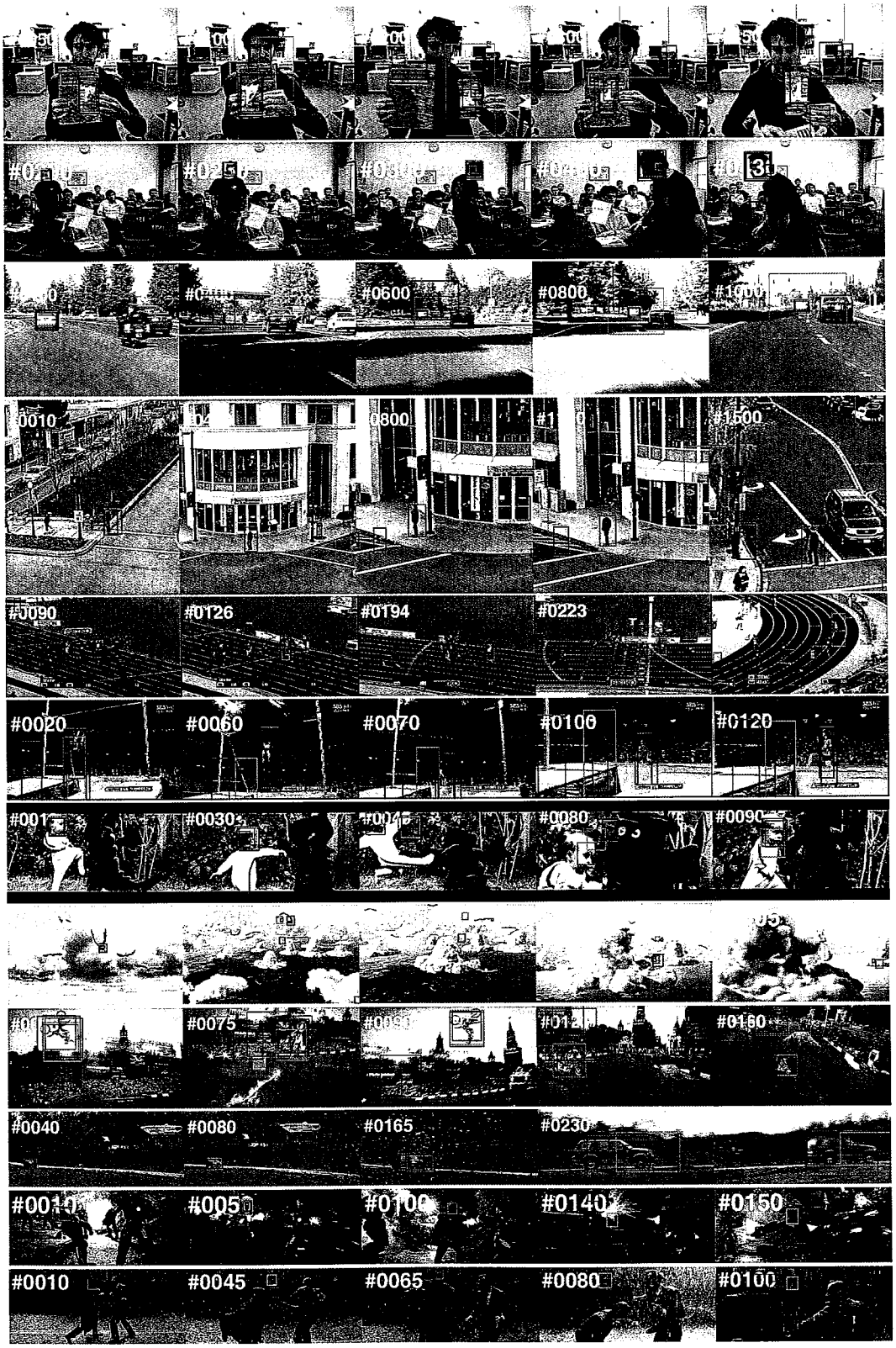


图 3.15 基于数据集 OTB-2015[1] 的 11 种属性标注的算法跟踪性能对比展示。



—RASNet —CREST —SINT —SiamFC_3s —CFNet

图 3.16 基于数据集 OTB-2015[1] 的视频序列(*clifBar*、*freeman3*、*car1*、*bolt*、*jump*、*dragonBaby*、*bird1*、*motorRolling*、*carScale*、*ironman* 和 *matrix*) 可视化定性比较 RASNet、CREST[9]、SINT[10]、SiamFC[7] 和 CFNet[11] 的跟踪性能。

3.5　本章小结

本章提出了基于残差注意力机制的孪生网络高效目标跟踪算法。算法通过将判别跟踪网络框架解耦为目标特征的表示网络以及用于判别学习的判别网络，重构了孪生网络架构组成。然后，本章提出了带有加权的交叉相关操作算子，可以对目标的不同空间位置的相关操作赋以自适应调整的权重。模型通过联合学习判别相关损失以及目标区域的判别系数，具有较强的表观形态的适应性。本章提出将用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行调整的通道注意力机制相结合的方式，共同学习模板图像判别信息。算法通过注意力机制的引入，减少了网络在训练过程中的过拟合，同时通过轻量化的网络设计，保证了良好的跟踪效率。最后，我们在四个跟踪数据集中的全面实验比较，验证了本章所提出的残差注意力机制在视觉目标跟踪中的有效性。

第 4 章　基于孪生网络的视觉目标跟踪与分割一体化高效处理框架研究

4.1　引言

在上一章中，我们提出使用注意力机制结合带加权的交叉相关操作帮助孪生网络进行自适应判别学习，利用孪生网络结构的共享特性加速特征提取与滑动窗口比对。然而由于算法并没有改变视觉目标跟踪对象的采样过程以及输出描述，使得模型仅构建了目标图像在二维平面上的平移运动，因而无法适应目标的长宽比以及旋转变化。基于此，本章着重于从目标跟踪问题的输出定义出发，重新提炼视觉目标跟踪对象的表述形式。通过孪生网络直接预测目标对象的分割结果，扩展了跟踪模型输出的表示方法，弥补了滑动窗口采样方式对于目标形态变化建模不足的缺陷。

通用深度学习网络主要用于分类、回归以及聚类等基本模式识别任务的学习，不同计算机视觉应用的本质差异通常由其具体定义的输入与输出所体现。例如，基于分类学习的图像分类任务 [39, 61]、基于分类与回归学习的图像目标检测任务 [28, 62] 以及对图像像素点进行分类的图像分割任务 [63, 64] 等。本文所关注的视觉目标跟踪任务通常需要算法使用简单的轴向对齐的矩形框 [3, 16, 17, 125, 132] 或旋转包围框 [8, 13] 表示目标对象。这种简单表述形式有助于降低数据标注的成本，同时方便用户快速执行目标初始化。但随着目标的形态多样性的提升，矩形框对于目标的表述较为粗糙，难以精确地描述非规则轮廓的目标对象。当目标对象发生非刚体形变或旋转时，矩形框表述通常会引入大量无关的背景信息，影响算法后续的学习与更新。

在本文第 §1.2.5小节中，我们详细地阐述了基于得分预测、动作决策、矩形框回归以及分割预测这四种输出模式对于视觉目标跟踪精度与速度的影响。当前视觉目标跟踪最主流的方法是：在线利用从视频第一帧图像所提供的真实标签来训练判别分类器，然后不断更新该分类器以适应目标表观变化。在过去的几年中，相关滤波器由于其简单可靠的表现获得越来越多研究人员的关注 [12, 21, 29, 54, 89]。相关滤波方法通过区分目标模板和背景物体以及它们的 2 维平移变换进行回归学习。Bolme 等 [21] 提出将相关滤波器学习引入到目标跟踪中，通过快速傅里叶变化实现了超高速的目标跟踪算法。此后，基于相关滤波器的

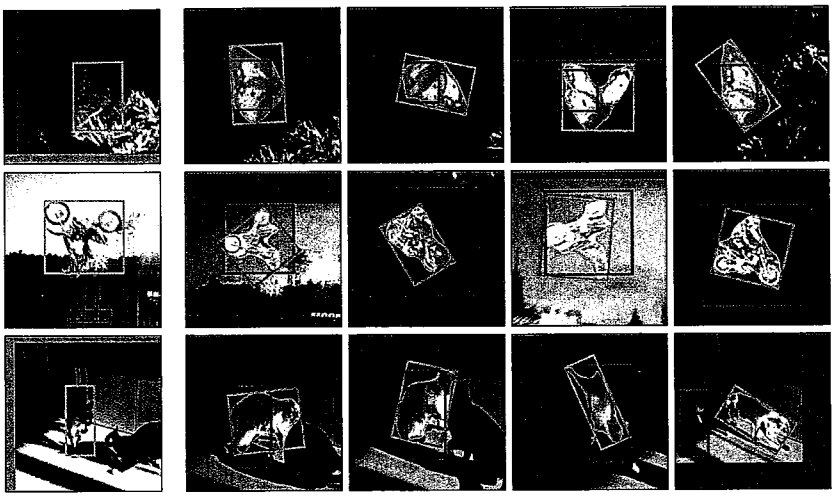


图 4.1 本章提出的孪生分割网络 SiamMask（绿色）和相关滤波算法 ECO[12]（红色）在数据集 VOT-2016[13] 视频序列 *butterfly*、*motocross1*、*fernando* 中的跟踪结果对比展示。其中，初始框为蓝色。

跟踪算法通过采用多通道特征 [29, 44]、正则空间约束 [53–55] 和深度学习特征 [11, 12, 30, 67] 等方式显著提升了精度。在第 §2章中，我们通过对相关滤波网络进行端到端学习，优化了特征表示网络的判别性能。但该方法表观模型同样只进行了线性回归分析，通过滑动窗口以及多尺度目标采样来估计目标的平移与尺度变化。

基于孪生卷积网络的算法研究 [7, 10, 35] 也在视觉目标跟踪中占据主流地位。研究人员 [10] 通过离线训练视频帧之间的相似性估计函数进行度量学习，去除在线训练判别分类器所带来的时间消耗。在跟踪测试过程时，直接利用孪生网络提取特征进行目标比对，通过相似性排序实现目标对象的定位，有效提升了跟踪效率。但从输出模式的角度而言，该类方法依然采用了和相关滤波器类似的候选区域滑动窗口采样过程，因而仍然难以建模目标的长宽比以及旋转变化。

上述目标跟踪算法都采用矩形边框来初始化目标，并且利用滑动窗口采样估计目标在后续帧中的位置。尽管矩形框表述较为简单方便，但它常常不能正确地描述目标对象。图4.1中展示了当前最高性能的判别相关滤波器 ECO[12] 在三个具有挑战性的视频序列中的跟踪结果。基于相关滤波器的跟踪算法的表观学习过程中仅对平面位移样本进行在线训练，采样测试过程中同样只包含平移样本以及尺度放缩样本，因而该类方法通常无法适应目标的长宽比以及旋转变化。在物体发生快速形变时，由于算法无法调整目标矩形框的长宽比，其预测结果

通常无法产生边缘紧致的目标表述。此外，由于跟踪器输出表述的局限，在目标长宽比发生快速变换的视频中，算法预测结果通常只包含目标对象的一个局部；在目标发生平面旋转时，其预测的矩形框会发生尺度适配问题；在目标周围存在干扰对象时，由于候选采样无法匹配目标形态变化，会造成错误跟踪。

上述由于目标表述局限造成的错误跟踪现象难以通过表观模型判别力的提升来改善。算法需要引入大量的位移、尺度、旋转以及长宽比变化样本联合学习与测试才能解决对于表观形态变化的适应，但枚举所有的形态变化会极大增加特征提取以及表观模型的计算负担，难以实现高效率的目标跟踪。因而跟踪算法的输出模式成为现阶段视觉跟踪算法的主要瓶颈。

文献 [118] 提出增加旋转测试样本来进行目标的旋转估计，但是该方法在增加旋转适应性的同时成倍降低了跟踪速度。为了增加孪生网络输出的长宽比适应性，文献 [35] 将目标图像与搜索区域图像堆叠输入到神经网络中进行目标矩形框回归预测。直接预测目标矩形框的优势是不用穷举评估就可以处理目标尺度以及长宽比的变化，因而算法取得了极高的跟踪速度。但该方法离线训练过程中难以生成所有位移样本，实际跟踪精度并不理想。研究人员 [70] 尝试将图像检测的建模方法迁移到孪生网络跟踪学习过程中，通过将图片检测中基于锚点（anchor）分类以及矩形框回归的网络框架迁移到视觉目标跟踪中的相似性度量以及矩形框回归学习。该类方法从本质上将孪生网络的单一得分输出转变为同时进行得分与矩形框回归输出，有效降低了对于采样数量的需求。由于采用了矩形框回归的输出模式，该算法实现了对于目标对象的长宽比变化的适应。随后，全卷积孪生网络方法通过困难样本挖掘 [72]、深化网络架构 [73] 和多阶段级联回归 [33, 34] 等改进，显著提高了跟踪精度性能，证明了输出模式对于视觉目标跟踪的重要性。在 VOT-2018[2] 和 VOT-2019[77] 中，研究人员通过采用基于矩形框回归作为输出的孪生网络架构，取得了实时跟踪挑战赛的冠军。

在本章中，我们充分挖掘神经网络的建模潜力，进一步提出使用孪生神经网络直接预测得到目标的分割表示，通过对于目标形态的精细描述来提升目标跟踪的精度。图4.1同时展示了本章所提出的孪生网络分割架构的输出结果，在每一帧中通过对目标对象的分割输出，生成目标对象的旋转矩形表述。从图中可以看到，这种表述形式显著提升了跟踪算法在复杂场景的表示精度，从目标表述形式上区别于传统的目标跟踪算法。

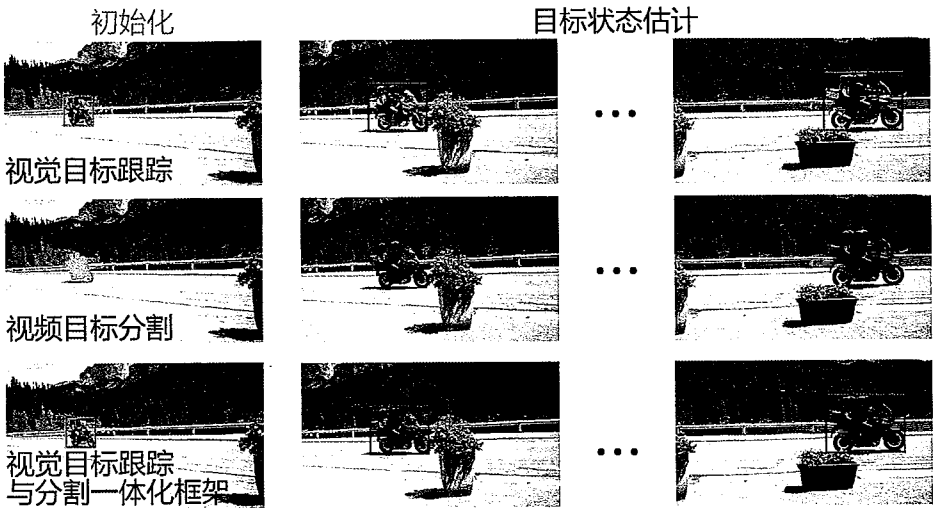


图 4.2 视觉目标跟踪、视频目标分割、视觉目标跟踪与分割一体化框架下的视频目标跟踪输入输出结果对比展示。

与视觉目标跟踪任务类似，半监督视频目标分割（Semi-supervised Video Object Segmentation，简称 VOS）[14] 任务同样用来构建目标对象在不同帧之间的关联关系。但该任务将初始帧的标注信息与所需要的输出同时扩展为分割表述。图4.2对比了两个任务的输入和输出形态，视觉目标跟踪任务的输入输出较为简洁，视频目标跟踪的输入输出结果较为精确。视频目标分割任务对于目标对象进行像素级跟踪估计，这种输出表述有助于跟踪过程的背景去除以及前景对象提取，被广泛应用于视频编辑领域。

视频目标分割方法的研究主要从图像分割方法 [63, 64] 进行演变，测试过程中算法利用首帧的图像与分割标签在线训练图像分割网络，然后对后续帧图像进行像素级前景背景分类。因此，视频目标分割中的经典算法 [122, 151, 152] 主要专注于目标的分割精度提升，运行速度较低，通常每帧需要几秒钟的运行时间。例如，在利用视频帧之间的一致性进行分割传播的视频目标分割方法中，研究人员 [122, 153–156] 使用图学习方法将第一帧的标注分割掩模传播到时间上相邻的图像帧。Bao 等 [156] 提出了利用时空马尔科夫随机场（Markov Random Filed，简称 MRF）进行分割结果传播。在该方法中，时间维度的传播依赖关系由光流模型建模，而空间依赖关系由卷积神经网络特征表述建模。文献 [151, 152, 157] 提出消除视频帧间的时序依赖关系，将视频目标分割任务转换为独立帧的目标分割检测任务。在 Maninis 等人提出的 OSVOS-S[157] 算法中，模型依赖于一个经过预先训练的全卷积网络进行分类，在测试时使用第一帧中提供的标注掩码对其进行

微调，在后续测试过程中不使用任何时序信息。此外，为了获得尽可能高的分割精度，视频目标分割方法通常具有计算密集的技术特点。研究人员通常采用在线微调 [151, 152, 156, 157]、数据增强 [158, 159] 和光流预测 [124, 151, 154, 156, 159] 等多种方法对分割结果进行提升。这些方法通常具有低运算速度的特点。对于视频目标分割领域最广泛使用的 DAVIS 数据集 [14] 中仅 10 秒钟的视频片段，分割算法 [152, 156] 需要数分钟的运行时间。尽管这种输出模式具有更为精确的目标表示特性，然而生成像素级的目标估计相比于简单的边界框估计过度消耗了计算资源。算法如果仅追求单一精度指标提升，将会限制其实际应用范围。

最近，研究人员 [124, 156, 160, 161] 加强了对高效率视频目标分割算法的研究。当前运行速度最快的方法是 Yang 等提出的 OSMN[161] 以及 Wug 等提出的 RGMP[162]。OSMN 使用元学习（meta-learning）网络构建的“调制器”，在测试过程中快速调整分割网络的参数，避免了使用随机梯度下降的优化方式对网络进行在线微调。RGMP 更进一步提出不使用任何在线微调，利用多个阶段训练的编解码器孪生网络架构进行分割。然而这两种方法的运行速度都低于每秒 10 帧，无法满足实时跟踪的需求。

对于现实中的许多计算机视觉应用场景而言，其接入的视频流需要实时处理并产生相应的行为决策，因而可以进行**实时**（real-time）以及**在线**（online）的视觉目标跟踪是其得以应用的必要条件。视觉目标跟踪任务中的在线跟踪是指不利用未来的图像帧来推断目标在当前帧的位置 [13]，这种设定也同样符合跟踪应用的现实需求。通用目标跟踪的基准数据集（如 [3, 125, 127]）假设跟踪器以连续的方式接收输入帧。这种属性通常用“在线”或“因果”来表示 [127]。同时，在视觉目标跟踪算法设计中，更高的算法效率也就意味着更少的硬件资源消耗以及更低的执行延迟（latency）。

为了解决上述算法对于目标表述以及运行速度方面的不足，本章提出孪生分割网络 SiamMask 作为视觉目标跟踪与视频目标分割任务的一体化处理框架（如图4.2所示）。该框架通过矩形框的方式对目标对象进行标注初始化，在后续帧中算法预测目标的分割表述。同时在算法设计过程中，我们借鉴了全卷积孪生网络的设计框架，提出了高效的跟踪与分割算法。在视觉目标跟踪领域，Yeo 等 [123] 的提出基于超像素表示的马尔科夫方法生成目标的分割表述，然而该方法的最快版本运行在 4fps 的速度下。当使用卷积神经网络特征时，其速度会下降

60 倍，下降到 0.1 fps 以下。在视频目标分割领域，Perazzi 等 [151] 和 Ci 等 [163] 提出的视频分割方法也可以采用矩形框进行初始化，并每帧输出目标分割掩码。但是，它们需要在测试时进行在线微调，这使它们无法满足实时应用需求。

在本章中，我们的目标是通过提出一种简单的多任务学习方法，来缩小视觉目标跟踪和视频目标分割之间的表示差距，利用统一的网络框架同时高效实现这两个任务。本章所提出的算法基于全卷积孪生网络架构 [7, 70]，在离线训练过程中利用视频目标检测数据集进行实例（instance）级别相似性度量学习，同时算法使用最近提出的具有大规模目标分割标注的 YouTube-VOS 数据集 [19] 进行分割学习。算法设计在保持在线实时速度的同时，将跟踪算法输出的简单轴向对齐的矩形框表述扩展为精确的分割表述。

为了实现这个目标，本章提出同时训练一个孪生网络完成三个任务的联合学习，每个任务对应一个不同的分支以及训练策略。我们首先通过孪生网络建立目标对象和候选区域之间的对应关系。如同 Bertinetto 等 [7] 提出的全卷积孪生网络方法一样，其中一个任务是用滑动窗口的方式学习目标对象和多个候选对象之间的相似性度量。该分支的输出是滑动窗口在每个候选位置的相关响应，它用于指示模板图像与候选图像的相似度，但该分支并不提供关于目标对象长宽大小的任何形状信息。为了改善孪生网络对于目标形态的表述，我们提出同时学习两个更深入的预测任务：使用区域建议网络实现目标边界矩形框回归 [70] 和使用分割网络实现目标类不可知的二值化目标分割 [164]。为了简化网络的初始化过程，我们对于图像的输入并不叠加分割信息，在离线训练时通过使用分割标签来计算搜索图像分割损失，这确保了在线过程中仅使用矩形框标注即可完成算法初始化。在本章提出的一体化跟踪分割框架中，每个任务都由一个不同的分支来完成。不同分支共享孪生网络特征表述，并分别学习分类、矩形框回归以及分割损失，将三个输出损失汇总在一起得到多任务学习架构。在网络完成离线训练后，SiamMask 仅依赖于简单的矩形边界框进行算法初始化，在线以约 55 帧每秒的运行速度生成目标对象的分割掩码以及相应的旋转边界框。由于目标分割表示的使用，SiamMask 在 VOT-2018[2] 数据集上建立了当时的最高精度指标。此外，与最近在 DAVIS-2016[14] 和 DAVIS-2017[18] 上进行半监督视频目标分割的算法相比，本章方法在精度方面具有竞争力，同时实现了当时最快的视频目标分割方法。

本章余下的部分做如下安排：在第 §4.2小节描述了我们所提出的孪生网络目标跟踪与分割一体化高效处理框架；在第 §4.3小节，我们对本章提出的算法进行了评测，并通过与现有的视觉目标跟踪算法以及视频目标分割算法进行对比，证明了所提出的分割表述对于目标跟踪精度的提升作用。通过模型各模块的对照实验，分析得出本章提出的分割分支对孪生网络特征学习具有促进作用。在第 §4.4小节中，我们将孪生分割网络算法进一步扩展到无监督多目标的视频分割领域，进一步证明本章所提出的一体化框架对于跟踪领域的重要意义。最后第 §4.5小节对本章的内容进行了概括总结。

我们将评测结果和统一框架的代码分享在项目网站，可以参见http://www.robots.ox.ac.uk/~qwang/SiamMask。

4.2　基于孪生网络的视觉目标跟踪与分割一体化高效处理框架

为了实现高效的在线视觉目标跟踪，我们采用了 Bertinetto 等人 [7] 提出的全卷积孪生网络框架作为基础框架。此外，为了说明本章所提出的方法不局限于特定的全卷积方法，我们同时采用 SiamFC[7] 和 SiamRPN[70] 两个代表性算法作为基础实例。我们在下一小节中首先简要介绍这两种孪生网络跟踪方法。

4.2.1　全卷积孪生网络

全卷积孪生网络 SiamFC[7]: Bertinetto 等 [7] 提出使用离线训练的全卷积孪生网络作为跟踪系统的基本构建模块，该网络将模板图像 $\mathbf{x}$ 与较大的搜索图像 $\mathbf{z}$ 进行相似度比较，通过滑动窗口采样以获得密集的相关响应图。$\mathbf{x}$ 和 $\mathbf{z}$ 分别是以目标对象为中心大小为 $w \times h$ 的图像区域和以目标的最后估计位置为中心的较大搜索区域。两个图像输入经过相同的卷积神经网络 f_θ 提取特征，两者特征由交叉相关操作得到响应图：

$$g_\theta(\mathbf{x}, \mathbf{z}) = f_\theta(\mathbf{x}) \star f_\theta(\mathbf{z}). \tag{4.1}$$

在本章中，我们将公式（4.1）左侧响应图中的每个空间位置称为候选窗口区域（Region of Window，简称 RoW）。具体地，$g^n_\theta(\mathbf{x}, \mathbf{z})$ 编码了模版图像 $\mathbf{x}$ 和在搜索窗口 $\mathbf{z}$ 中的第 n 个候选窗口之间的相似性。对于 SiamFC 的测试过程，响应图中的最大值的目标位置所对应的搜索区域就是目标在搜索窗口中的位置。算法通过交叉相关操作度量模板特征与候选图像特征，得到单一数值的相似度响

应。在本章中，算法充分利用孪生网络学习到的特征表述，预测更加丰富的相关信息。因而，为了让每一通道编码更丰富的目标对象信息，我们将公式（4.1）中的简单的交叉相关操作替换为分层（depth-wise）交叉相关操作 $\star_d$ 同时产生多通道响应图 [73]。离线过程中，SiamFC 采用 Logistic 损失在数百万个视频帧中进行训练，本章把这个相似性度量损失函数记为 $\mathcal{L}_{sim}$。

孪生区域提议网络 SiamRPN[70]：Li 等人 [70] 利用区域建议网络（Region Proposal Network，简称 RPN）显著提高了全卷积孪生网络 SiamFC 的跟踪精度，该孪生区域建议网络允许使用可变长宽比的矩形框来估计目标位置。特别地，在 SiamRPN 中，网络输出每一通道编码一组 k 个预设锚点（anchor）的目标 / 背景分数和相应的矩形框回归结果。因此，SiamRPN 并行输出分类分数与回归预测的矩形框。利用交叉熵损失和 soft-ℓ_1 损失 [62] 对两个输出支路进行训练。在本章中，我们将它们分别称为 $\mathcal{L}_{cls}$ 和 $\mathcal{L}_{box}$。

4.2.2 孪生分割网络

本章提出的孪生分割网络 SiamMask 不同于上述依赖于低精度目标表示的跟踪方法，算法主要突出生成分割掩码（segmentation mask）的重要性。孪生网络算法本质上通过共享的特征提取器对目标图像以及候选图像进行特征提取，通过交叉相关或神经网络预测所需的候选目标表示。除了相似性度量预测和回归矩形边界框坐标输出外，全卷积孪生网络的通道向量具有较大的模型容量，可以被用来生成目标像素级分割所需的编码信息。本章提出通过使用额外的分支和损失来扩展现有的孪生网络跟踪器。最为直接的方式：孪生网络相关操作中的每一个候选区域 RoW 对应于搜索图像中分辨率大小为 $w \times h$ 的图像块。我们为每一个候选窗口预测一个分辨率为 $w \times h$ 二值分割掩码，通过将分割结果向量化的表示形式进行编码。算法使用一个简单的轻量化神经网络 h_ϕ 来学习分割参数，其中的网络参数记作 ϕ。假设 m^n 表示第 n 个候选窗口的预测分割掩码，

$$m^n = h_\phi \left(g_\theta^n(\mathbf{x}, \mathbf{z}) \right), \tag{4.2}$$

从公式（4.2）中，我们可以得出候选窗口分割的预测函数是由搜索区域 $\mathbf{z}$ 与跟踪目标 $\mathbf{x}$ 共同决定的函数。通过这种样本对输入设计，神经网络可以使用 $\mathbf{x}$ 作为参考以及条件先验指导分割过程。该结构设计同时使得算法可以跟踪任意类的目标对象：给定不同的参考图像 $\mathbf{x}$，网络将为 $\mathbf{z}$ 生成不同的分割掩码。基于图像分

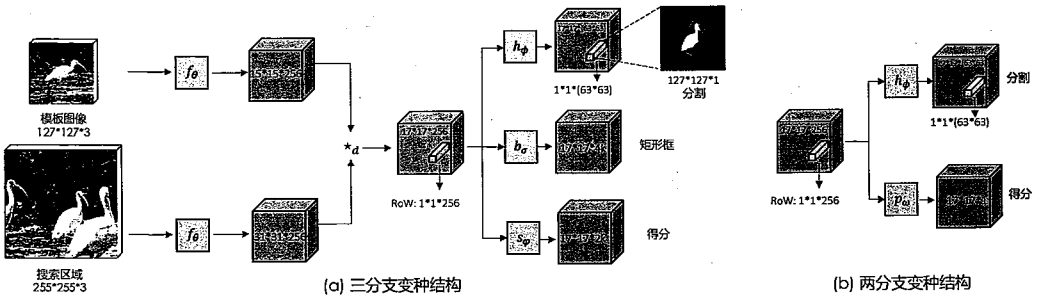


图 4.3 孪生分割网络的两种跟踪变体结构示意图：(a) 三分支（完整）架构；(b) 两分支架构。其中，$\star_d$ 表示逐通道相关操作。

割网络的视频目标分割算法通常只包含单一输入 $h_\phi\left(f_\theta^n(\mathbf{z})\right)$，因而无法根据目标样本作为条件控制输出结果，必须经过在线学习训练以吸收模板图像标签信息。

4.2.2.1　损失函数设计

在离线训练过程中，每一个候选窗口首先被标注为基于前景背景分类的正负样本标签 $\mathbf{y}^n \in \{\pm 1\}$，并与大小为 $w \times h$ 的基于实值的二值分割掩码标签 c^n 相关联。我们假定 $\mathbf{c}_{ij}^n \in \{\pm 1\}$ 为像素 (i, j) 在第 n 个候选窗中的对象分割掩码。分割预测任务的损失函数 $\mathcal{L}_{mask}$ 为所有位置的二分类 Logistic 回归损失均值的总和:

$$\mathcal{L}_{mask}(\theta, \phi) = \sum_n \left(\frac{1 + \mathbf{y}^n}{2wh} \sum_{ij} \log\left(1 + e^{-\mathbf{c}_{ij}^n \mathbf{m}_{ij}^n}\right) \right). \tag{4.3}$$

因此，分割分支 h_ϕ 实际上由 $w \cdot h$ 个二值分类器组成，每个分类器用来指示给定的像素是否属于候选窗口中的目标。注意，$\mathcal{L}_{mask}$ 只考虑 RoW 中的正样本，即 $\mathbf{y}^n = +1$ 的样本。候选窗口中存在大量的无关负样本，其分割输出理论上应完全为 -1。但如果对于负样本的分割预测进行训练，会造成分割学习的类别不平衡，影响最终的分割结果。

4.2.2.2　两种跟踪变体结构

我们将 SiamFC[7] 和 SiamRPN[70] 的架构分别增加了本章提出的分割分支以及分割损失 $\mathcal{L}_{mask}$，得到了 SiamMask 的两分支与三分支变体。图4.3展示了 SiamMask 的两种跟踪变体形式，它们分别优化了多任务损失 $\mathcal{L}_{2B}$ 和 $\mathcal{L}_{3B}$:

$$\mathcal{L}_{2B} = \lambda_1 \cdot \mathcal{L}_{\text{mask}} + \lambda_2 \cdot \mathcal{L}_{sim}, \tag{4.4}$$

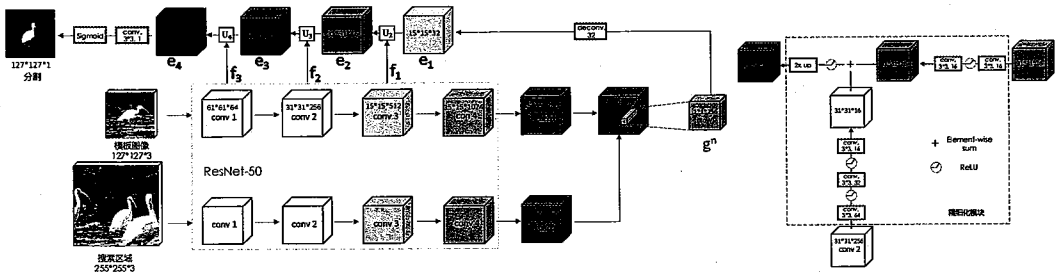
$$\mathcal{L}_{3B} = \lambda_1 \cdot \mathcal{L}_{mask} + \lambda_2 \cdot \mathcal{L}_{cls} + \lambda_3 \cdot \mathcal{L}_{box}. \tag{4.5}$$

本章第 §4.2.1小节具体描述了全卷积孪生网络相似性度量损失 $\mathcal{L}_{sim}$ 以及孪生区域提议网络的分类损失 $\mathcal{L}_{cls}$ 和矩形框回归 $\mathcal{L}_{box}$ 的构成。下面我们将按照分支组成具体定义目标的正负样本设定。对于三分支结构损失 $\mathcal{L}_{3B}$，候选窗口 RoW 的锚点框与目标矩形框标签的重叠率大于等于 0.6 时，则该窗口位置被认为是正样本，即 $\mathbf{y}^n = 1$；否则该窗口位置被认为是负样本 $\mathbf{y}^n = -1$。对于两分支结构损失 $\mathcal{L}_{2B}$，我们采用与 SiamFC 中相同的策略来定义正样本和负样本，即距离目标中心位置小于等于 16 个像素的候选窗口定义为正样本，超过 16 个像素以外的候选窗口为负样本。公式（4.4）和公式（4.5）中的超参数简单地设置为 $\lambda_1 = 32$ 和 $\lambda_2 = \lambda_3 = 1$。矩形框回归和相似性得分输出的任务相关分支由两个卷积核大小为 1×1 的卷积层组成。

4.2.2.3 孪生网络分割输出编码方法

本章提出对于视觉跟踪中的目标物体进行分割表述，因而在具体实现中如何编码分割结果具有重要意义。在矩形框表述中，算法通常采用 4 个元素的向量分别编码矩形框的左上角与右下角坐标位置。而对于目标分割而言，较为直观的编码方式是算法直接输出与原始图像一样大小的二维数组，然而该表述具有较高的参数量以及运算量。图像语义分割方法中的代表作 FCN[63] 和 Mask R-CNN[28] 采用这种类似的编码方式，在整个网络中显式地保持空间分辨率，但该类方法通常无法取得实时分割速度。为了降低计算消耗，算法通常采用低分辨率预测进行上采样输出。

在本节中我们提出使用一个高维向量来表述目标的分割掩码。具体地，在本章所提出的算法中，目标的隐含表述对应于 17×17 个候选区域 RoW 特征 $f_\theta(\mathbf{z})$ 和目标特征 $f_\theta(\mathbf{x})$ 逐通道的互相关中。网络 h_ϕ 由两个 1×1 的卷积层级联构成，两个卷积层分别包含 256 个通道和 63^2 个通道（如图4.3）。这使得每个像素分类器的输入信息包含了候选窗口中的全部信息，从而在每个像素分类时具有对应候选窗口图像 $\mathbf{z}$ 的完整感受野。这种全局信息的获取能力是消除相似表观对象干扰的关键，而通用的图像分割网络在保持分辨率的同时，只具有局部感受野。此外，由于本章提出的算法网络不需要在平面空间维度上保持图像分辨率，相比于传统的分割网络 [64] 具有更快的运行效率。为了生成更精确的分割掩模，我们采用了自顶向下的精细化分割策略，提出使用由上采样层和跨层连接组成的多个精细化模块来合并低分辨率分割表述和高分辨率目标特征。



(a) 自顶向下的分割精细化结构　　(b) 精细化融合模块

图 4.4 自顶向下的分割精细化模块结构示意图：（a）通过将目标候选区域 RoW 响应依次与底层特征融合，得到更高分辨率的特征表述；（b）精细化融合模块。

4.2.2.4　分割精细化模块

由于目标空间信息的丢失，采用向量化的网络分割预测方式会损失细节信息以及边缘信息。文献 [148] 提出使用编解码的网络结构提升目标表示精度，通过结合网络底层信息与上采样的高层信息逐步精细化目标分割预测。受该方法启发，为了生成更精确的目标分割边缘，我们采用了自顶向下的多层特征融合机制。通过使用由上采样层和跨层连接组成的多个精细化模块合并低分辨率的相关响应和高分辨率底层特征。图4.4a显示了使用堆叠的精细化模块生成最终分割掩码连接过程。模板图像和搜索图像通过共享的特征提取网络得到交叉相关特征 $g(\mathbf{x},\mathbf{z})$，其中目标所在的候选窗口的相关特征表示记为 $g^n(\mathbf{x},\mathbf{z}) \in \mathbb{R}^{1\times1\times d}$，该目标窗口在网络的不同层级特征分别记作 $\mathbf{f}_1$、$\mathbf{f}_2$ 和 $\mathbf{f}_3$。我们首先通过反卷积操作将目标窗口的相关表示编码为具有一定分辨率的多通道分割编码 $\mathbf{e}_1 \in \mathbb{R}^{m_1\times m_1\times k_1}$，然后通过精细化模块 U_2 将顶层的分割编码 $\mathbf{e}_1$ 与高层特征 $\mathbf{f}_1$ 结合进行上采样得到较高分辨率的分割编码 $\mathbf{e}_2 = U_2(\mathbf{e}_1,\mathbf{f}_1)$，其中 $\mathbf{e}_2 \in \mathbb{R}^{m_2\times m_2\times k_2}, k_2 > k_1$。此后目标分割编码 $\mathbf{e}_2$ 依次迭代经过精细化模块 U_3 和 U_4 直到得到目标的高精度分割编码 $\mathbf{e}_4$。在对相关特征表示 g^n 进行上采样的过程中，算法融合了搜索图像的不同层特征表示进行信息互补。图4.4b给出了精细化模块的设计结构，通过底层特征的联合学习增强了最终的分割输出精度。

4.2.2.5　矩形框生成策略

尽管视频目标分割任务只需要得到分割输出，但典型的跟踪基准 [1, 4] 都需要算法输出矩形框作为目标对象描述。为了兼容视觉目标跟踪任务，我们提出利用分割结果 $\mathbf{m}^n$ 生成目标的矩形框表述。如图4.5所示，本章算法考虑了三种不同

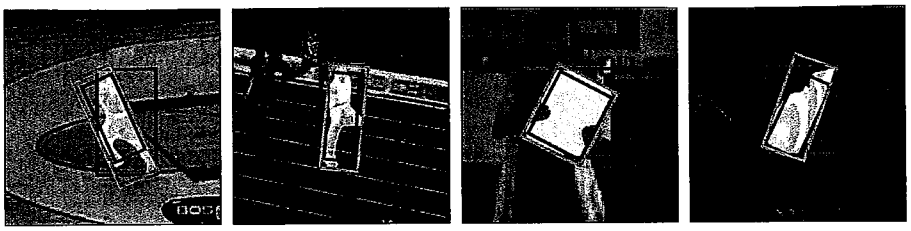


图 4.5 通过二值分割掩码（　　）生成矩形边界框的图像示例。其中，*Min-max*：轴向对齐的最小外包矩形框（红色）；*MBR*：旋转最小外接矩形框（绿色）；*Opt*：使用 VOT-2016[13] 中提出的优化策略得到的矩形（蓝色）。

的从二值分割掩码生成矩形边界框的策略:（1）通过分割掩码生成轴向对齐的最小外接边界矩形（记作 Min-max），（2）通过分割掩码生成旋转的最小外接矩形（记作 MBR）以及（3）VOT-2016[13] 中提出的通过优化策略生成最优表述边界框（记作 Opt）。我们在第 §4.3.1.1小节中对这三种策略进行了实验评估。

4.2.3　网络参数设置

表 4.1 孪生分割网络的主干网络架构参数展示。

模块	模板区域	搜索区域	结构参数
输入	127×127	255×255	7×7, 64, 步长 2
conv1	61×61	125×125	7×7, 64, 步长 2
conv2_x	31×31	63×63	3×3 池化, 步长 2 [1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	15×15	31×31	[1×1, 128 3×3, 128 1×1, 512] ×4
conv4_x	15×15	31×31	[1×1, 256 3×3, 256 1×1, 1024] ×6
调整层	15×15	31×31	1×1, 256
相关响应	17 × 17		逐通道交叉相关

网络架构设置：表4.1显示了孪生分割网络主干架构 f_θ 的详细结构参数。对于本章所提出的两种跟踪变体，算法都采用 ResNet-50[61] 网络结构第 4 卷积层的最后一个卷积输出作为共享特征提取器。为了使深层卷积网络输出获得更高的空间分辨率，我们使用步幅为 1 的卷积层替换原有的步幅为 2 的卷积层，将网络输出的有效步长从 32 减小到 8。此外，算法通过使用膨胀卷积来增加感受野。具体来说，我们将 conv4 的 3×3 卷积层的步长设为 1，膨胀率设为 2。与原

表 4.2 孪生分割网络的两分支变种输出网络架构参数展示。

模块	score 分支	mask 分支
conv5	$1\times1, 256$	$1\times1, 256$
conv6	$1\times1, 1$	$1\times1, (63\times63)$

表 4.3 孪生分割网络的三分支变种输出网络架构参数展示。k 表示锚点数量。

模块	score 分支	box 分支	mask 分支
conv5	$1\times1, 256$	$1\times1, 256$	$1\times1, 256$
conv6	$1\times1, 2k$	$1\times1, 4k$	$1\times1, (63\times63)$

始 ResNet-50 不同的是，conv4 中去除了下采样操作。此外，我们还在主干网络中增加了一个调整层，该层包含一个具有 256 个输出通道 1×1 的卷积层。模板图像和搜索区域图像共享由 conv1 到 conv4 的网络参数，而调整层的网络参数不进行共享。将调整层的输出特征进行逐通道交叉相关，得到大小为 $17\times17\times256$ 的相关特征表示。表4.2和表4.3分别介绍了孪生分割网络的两分支变种以及三分支变种的输出分支网络架构，三分支网络架构的锚点数量 k 设置为 5。

离线训练设置：和通用孪生网络 [7, 70] 训练相一致，我们使用模板图像和搜索区域图像大小分别为 127×127 和 255×255 个像素。在训练过程中，我们通过对模板和搜索区域随机移动（random shift）以及随机缩放（random scale）实现数据增强。具体来说，对于搜索区域图像随机平移范围设置在 ±64 个像素以内；对于模板图像随机平移范围设置在 ±4 个像素以内；对于模板图像和搜索区域图像分别采用 [0.95, 1.05] 和 [0.82, 1.18] 范围内的随机缩放。在跟踪任务训练之前，我们将网络主干在 ImageNet-1k 分类任务 [6] 中进行预训练。随后在跟踪任务训练过程中，我们使用随机梯度下降（SGD）进行训练，学习速率在最开始的 5 个周期内从 10^{-3} 线性增加到 5×10^{-3}。然后再训练 15 个周期，学习率从 5×10^{-3} 对数递减到 5×10^{-4}。在本章中，网络训练数据使用 COCO[116]、ImageNet-VID[6] 和 YouTube-VOS[19] 三个数据集进行混合。

在线测试设置：在跟踪过程中，孪生分割网络 SiamMask 在每一帧中只进行网络前向计算，而不进行任何在线训练调整。在孪生分割网络的两分支变体中，我们使用在分类分支中获得最大分数的候选窗口位置来选择输出分割掩码。然后，对每个像素前景概率估计 $\mathbf{m}^n$ 进行 Sigmod 函数归一化，并采用阈值 0.5 对分割分支的输出进行二值化。在两分支变体中，对于第一个视频帧之后的每个视频帧，我们将输出分割结果与 Min-max 框匹配，并将其作为参考来裁剪下一个帧搜索区域。在三分支变体中，我们发现利用矩形框回归分支输出作为位置参考可以取得最好的跟踪性能。

4.3 实验评估与分析

在本节中，我们分别评估孪生分割网络方法在视觉目标跟踪任务和视频目标分割上的性能指标。具体地，我们在 VOT-2016[13] 和 VOT-2018[2] 数据集中评价视觉目标跟踪性能，并在大规模视频目标跟踪数据集 GOT-10k[16] 和 TrackingNet[17] 中验证算法的鲁棒性。然后，我们在 DAVIS-2016[14]、DAVIS-2017[18] 以及 YouTube-VOS[19] 数据集中评估算法在半监督视频目标分割任务上的性能。本节分别采用 SiamMask-2B 和 SiamMask 表示本章提出的两分支变体和完整的三分支变体。

4.3.1 视觉目标跟踪任务性能评估

数据集和设置：我们采用广泛使用的 VOT-2016[13] 和 VOT-2018[2] 基准来评估算法在视觉目标跟踪任务上的性能。这两个数据集都使用旋转矩形边框对目标对象进行标注。我们首先在 VOT-2016 数据集中进行对照实验，通过实验结果对比不同类型的目标输出表述对跟踪性能的影响。在第一个实验中，我们使用了平均重叠率（mIoU）以及在 0.5 和 0.7 作为 IoU 阈值下的平均精度（mAP）来评估算法性能。然后，我们使用官方提供的工具包在 VOT-2018 数据集中与当前最好跟踪算法进行比较，分别通过精度（Accuracy）、鲁棒性（Robustness）和期望平均重叠率（EAO）对算法进行排序。EAO 指标同时考虑了跟踪器的准确性和鲁棒性。最后，我们在大规模视觉目标跟踪数据集 GOT-10k[16] 和 TrackingNet[17] 中测试了算法的泛化性能。

4.3.1.1 分割输出的重要性验证

现有的视觉目标跟踪方法通常使用固定长宽比 [7, 29] 或可变长宽比 [25, 70] 的坐标轴对齐（axis-aligned）矩形框来表述目标位置。在本章中，我们提出使用分割结果表述目标对象形态，通过分割结果快速生成旋转矩形框表述用来测评视觉目标跟踪任务性能。为了定量分析输出二值分割掩码对于跟踪性能的影响，我们设置了两组对照实验。在本实验中主要关注算法的输出模式对跟踪精度的影响，因而对视频帧进行随机采样生成样本对，消除时序因素对性能评估的影响。下表描述的结果是在 VOT-2016 数据集上随机裁剪的搜索图像块上进行测试评估，随机移动范围设定在 ± 16 像素内，缩放变形尺度设定在 $2^{\pm 0.25}$。

表4.4中，我们首先对比了固定长宽比的矩形框、轴向对齐最小外接矩形框

表 4.4 基于数据集 VOT-2016[13] 不同矩形框输出策略的跟踪性能对比展示。

Method	mIoU (%)	mAP@0.5 IoU	mAP@0.7 IoU
Fixed Oracle	73.43	90.15	62.52
Aligned Oracle	77.70	88.84	65.16
MBR Oracle	84.07	97.77	80.68
SiamFC[7]	50.48	56.42	9.28
SiamRPN[72]	60.02	76.20	32.47
SiamMask-*Aligned*	65.05	82.99	43.09
SiamMask-*MBR*	67.15	85.42	50.86
SiamMask-*Opt*	**71.68**	**90.77**	**60.47**

以及旋转最小外接矩形框表述性能的差异。对应的三种理论实验设定如下：（1）固定长宽比理论实验（Fixed Oracle）在每帧的输出结果使用真实目标的面积和中心位置，但将长宽比调整为第一帧中目标物体的长宽比，并生成轴向对齐的矩形框作为预测输出。（2）轴向对齐最小外界矩形框理论实验（Aligned Oracle）在每帧中利用目标真实分割标签生成轴向对齐的最小外接矩形框作为预测输出。（3）最后，旋转最小外接矩形理想实验（MBR Oracle）在每帧中利用目标真实分割标签生成旋转的最小外接矩形框作为预测输出。然后，我们对比了以 SiamFC[7] 和 SiamRPN[70] 为代表的固定长宽比与可变长宽比矩形框表示方法和本章提出的三分支变体分别使用 Aligned，MBR 和 Opt 方法的真实性能。同时，注意理想实验设定（1）、（2）和（3）可以分别认为是 SiamFC、SiamRPN 和本章所提出的 SiamMask 表示策略的性能上界。

通过表4.4中对比理想实验所代表的上限性能，我们可以发现固定长宽比矩形框和旋转最小外接矩形框之间的 mIoU 精度改进幅度为 +10.64%。这说明只要简单地改变矩形框的表示形式，跟踪器就可以获得很大的改进空间。在真实算法性能比较中，本章提出的孪生分割网络 SiamMask 无论使用何种矩形框生成策略（详见第 §4.2.2.5小节）都能获得最佳的平均重叠率（mIoU）。其中 SiamMask-Opt 提供了最高的平均重叠率和平均精度性能，但该方法通过最大化矩形框中的前景分割结果以及最小化矩形框内的背景迭代求解，需要消耗大量的计算资源。SiamMask-MBR 实现了 85.4% 的 mAP@0.5 IoU，相对于 SiamFC 和 SiamRPN 分别提高了 +29% 和 +9.2%，这一显著提升证明了分割输出对于矩形框估计的提升。当考虑到 0.7 IoU 阈值的精度时，SiamMask-MBR 相比于 SiamFC 与 SiamRPN

表 4.5 基于数据集 VOT-2016[13]、VOT-2018[2] 本章提出的 SiamMask 算法采用不同矩形框输出策略的跟踪性能对比展示。

Method	VOT-2016[13]			VOT-2018[2]			Speed
	EAO ↑	A ↑	R ↓	EAO ↑	A ↑	R ↓	
SiamMask-box	0.412	0.623	0.233	0.363	0.584	0.300	**76**
SiamMask	0.433	0.639	**0.214**	0.380	0.609	**0.276**	55
SiamMask-*Opt*	**0.442**	**0.670**	0.233	**0.387**	**0.642**	0.295	5

精度提升分别达到 +41.6% 和 +18.4%，性能差距显著扩大。因而随着跟踪算法的判别性能逐步提升，低阈值平均精度会很快饱和，目标输出模式将成为视觉目标跟踪算法的主要瓶颈。此外，本文所提出孪生分割网络算法的精度结果与固定长宽比的理想实验相差较小，这表明采用固定长宽比的算法优化上限可能与本章所提出的算法接近。

总的来说，这组对照实验显示了从目标对象的二值分割掩码中获得旋转矩形框策略比简单输出轴向对齐的边界框策略具有显著的精度优势。

4.3.1.2 基于数据集 VOT-2016 和 VOT-2018 的评测结果分析

表4.5显示了本章算法 SiamMask 使用不同矩形框生成策略在 VOT-2016[13] 和 VOT-2018[2] 数据集的性能对比。SiamMask-box 是指使用 SiamMask 的矩形框回归分支进行推理的跟踪结果。在较为简单的 VOT-2016 数据集中，相比于直接输出轴向对齐矩形框表示的 SiamMask-box，本章所提出的生成分割掩码方式 SiamMask-*Opt* 在整体指标 EAO 中提升 3%，在精度指标下提升达到 4.7%。该结果充分证明了特征表示对于目标跟踪的重要性。在更具挑战的 VOT-2018 中，通过分割分支生成的矩形框输出 SiamMask-*Opt* 相较于只使用矩形框回归分支的 SiamMask-box 的精度提升达到 5.8%，整体性能指标 EAO 提升达到 2.4%，算法的鲁棒性也有所提升。此外，采用最小外接矩形的 SiamMask 算法同样在各项指标下都优于 SiamMask-box，算法在实现分割输出的基础上可以保持超实时的运算速度。

在表4.6中，我们将 SiamMask 最小旋转外接矩形策略，SiamMask-opt 以及两分支变体与 VOT-2018 上与最近发布的五个跟踪器进行了比较。SiamMask 指代使用旋转最小外接矩形策略的三分支变体。SiamMask-2B 指代使用旋转最小

表 4.6 基于数据集 VOT-2018[2] 本章提出的 SiamMask 算法与其他领先算法的跟踪结果对比展示：比较指标包含期望平均重叠率（EAO）、跟踪精（Accuracy）、跟踪鲁棒性（Robustness）以及平均跟踪速度。

Metric	SiamMask *-Opt*	SiamMask	SiamMask *-2B*	DaSiamRPN [72]	SiamRPN [70]	SA_Siam_R [109]	CSRDCF [55]	STRCF [76]
EAO ↑	**0.387**	**0.380**	0.334	0.326	0.244	0.337	0.263	0.345
Accuracy ↑	**0.642**	**0.609**	0.575	0.569	0.490	0.566	0.466	0.523
Robustness ↓	0.295	0.276	0.304	0.337	0.460	0.258	0.318	**0.215**
Speed (fps) ↑	5	55	60	160	**200**	32.4	48.9	2.9

外接矩形策略的两分支变体。首先，SiamMask 的两种变体都实现了出色的跟踪精度和实时运行速度。特别地，三分支变体 SiamMask 显著优于最近提出的性能最优的孪生网络算法 DaSiamRPN[72]，实现了 0.380 的 EAO，并以 55fps 速度运行。DaSiamRPN 通过优化困难样本学习提升孪生区域提议网络 SiamRPN[70] 的判别性能，算法同时引入了大规模视频检测数据集 YouTube-BoundingBoxes[117] 辅助跟踪训练，本章提出的算法所需的训练数据量要显著低于 DaSiamRPN。即使去除矩形框回归分支，本章所提出的两分支变体（SiamMask-2B）也可以实现 0.334 的高 EAO 性能，与 SA-Siam-R[109] 取得类似的精度，并且优于已发表文献中的任何其他实时方法。算法 SA-Siam-R 采用了与本章同样的主干网络，并结合多层特征进行相似性度量学习，而本章所提出的孪生分割网络只采用单层网络进行相关分析。此外，SiamMask-Opt 方法取得了 0.387 的最优 EAO 指标性能，但运行速度只有 5fps。这是因为矩形框优化策略需要更多的计算来提供更高的精度。

本章所提出的模型在精度指标（Accuracy）表现较优，与基于相关滤波器的跟踪算法 CSRDCF[55]、STRCF[76] 相比具有显著优势。如表4.6所示，相比于使用深度学习特征的在线学习相关滤波器 STRCF，SiamMask 取得了 11.9% 的精度提升。尽管本章算法去除了在线学习过程，SiamMask 依赖于更丰富的目标表示输出带来显著提升。He 等人 [109] 通过采样多个旋转角度下的搜索区域进行旋转矩形边界框估计。但是，该算法输出表示仍然受到固定的长宽比限制，因而本章所提出的算法相比于 SA_Siam_R 取得了 7.6% 的精度提升以及更快的跟踪速度。

为了对比不同算法在实时跟踪特性，我们在 VOT-2018 数据集中进行了实

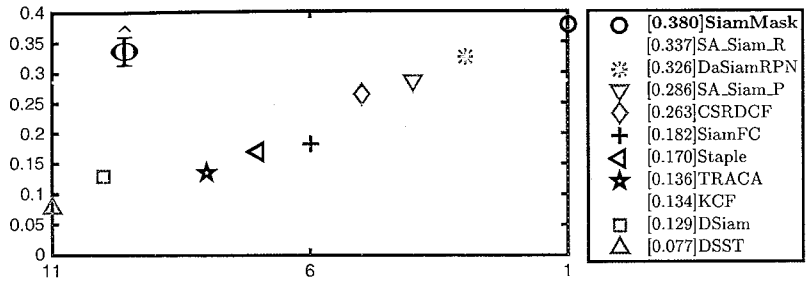
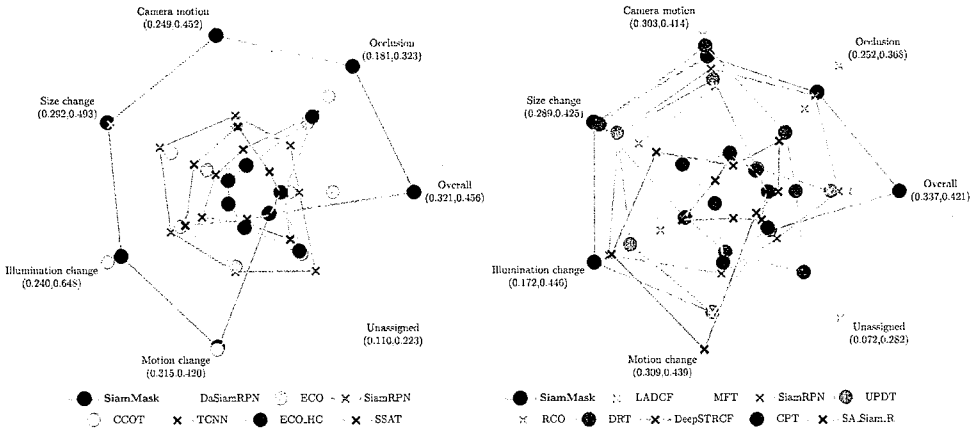


图 4.6 基于数据集 VOT-2018[2] 本章提出的 SiamMask 算法与其他领先算法进行实时跟踪的期望平均重叠率（EAO）排序图。



(a) VOT-2016 **(b) VOT-2018**

图 4.7 本章提出的 SiamMask 算法与其他领先算法在数据集 VOT-2016[13]、VOT-2018[2] 的不同视觉场景的跟踪结果对比展示。

时跟踪实验。图4.6展示了本章所提出的 SiamMask 与 VOT-2018 数据集实时跟踪性能前十名的算法的对比。基于孪生网络学习的方法在跟踪速度方面具有较大优势，本章所提出的 SiamMask 与基于孪生网络的跟踪算法 SA_Siam_R 、DaSiamRPN、SA_Siam_P 取得了前四名的跟踪性能。相比全卷积孪生网络 SiamFC[7]，本章所提出的孪生分割网络算法性能提升达到 19.8%，该结果充分说明了输出表示模式对于跟踪性能的影响。

除了对算法整体性能的评价分析，VOT 数据集精细地标注了每一帧图像的视觉场景属性，通过在每个场景属性进行分别测评以便分析算法的优缺点。具体而言，VOT 数据集中的视觉属性主要包含目标遮挡、光照变化、运动变化、尺度变化以及摄像机运动。本节分别在 VOT-2016 与 VOT-2018 的 5 种视觉属性中比较了本章提出的孪生分割网络与其他领先算法 [12, 31, 70, 72, 76, 96, 103, 109]，并将结果展示在图4.7中。本章提出的算法在 VOT-2016 数据集中的大多场景属

表 4.7 基于数据集 GOT-10k[16] 本章提出的 SiamMask 算法与其他领先算法的跟踪结果对比展示：比较指标包括平均重叠率（AO）、0.75 重叠阈值下的成功率（$SR_{0.75}$）以及 0.5 重叠阈值下的成功率（$SR_{0.5}$）。

Metric	SiamMask	CFNet[11]	SiamFC[7]	GOTURN[35]	CCOT[31]	MDNet[25]
AO ↑	**51.4**	37.4	34.8	34.2	32.5	29.9
$SR_{0.75}$ ↑	**36.6**	14.4	9.8	12.4	10.7	9.9
$SR_{0.5}$ ↑	**58.7**	40.4	35.3	37.5	32.8	30.3

性中取得最优结果。在目标尺度变化场景中，相比于连续相关滤波算法 ECO[12]、CCOT 在 EAO 指标下提升 10.6% 和 16.5%。基于相关滤波器的跟踪算法缺乏窗宽比调整能力，无法适应目标的快速形变。在 VOT-2018 数据集的目标尺度变化场景中，算法依然保持最高性能，相比于使用多种数据增强的相关滤波算法 UDPT[96] 提升达到 2.6%。通过各属性场景下的性能分析，可以得出本章提出的算法性能较为均衡，同时由于使用分割预测方式进行目标跟踪，在目标尺度估计方面取得较为优异的性能。

4.3.1.3　基于大规模视觉目标跟踪数据集的评测结果分析

GOT-10k[16] 是近期由中科院提出的大规模多类别跟踪数据集，其中包含一万个矩形框标注的视频序列。跟踪算法在 180 个测试视频上进行评估，其中包含 84 个不同的类别目标以及 32 种运动模式。跟踪算法根据平均重叠率（Average Overlap，简称 AO）进行排序，本节同时报告了算法在 0.5 和 0.75 重叠阈值处的成功率。在该数据集中，我们与代表当前最高性能的 CFNet[11]、SiamFC[7]、GOTURN[35]、CCOT[31] 以及 MDNet[25] 进行对比。表4.7报告了各算法在 GOT-10k 中的性能，与在该基准上表现最好的 CFNet[11] 相比，本章提出的孪生分割网络 SiamMask 在所有性能指标上都取得显著优势，并且在平均重叠率指标下实现了 37% 的相对提升，在以 0.75 重叠率为阈值的成功率指标中相对提升达到 154%。这一结果充分说明了基于分割的输出表示对于跟踪算法实现高精度跟踪的重要性。同时，算法在具有广泛目标类别的 GOT-10k 数据集中取得一致提升，这表明本章提出的算法在不同类别对象中具有较强的泛化性能。

TrackingNet[17] 是另一个用于训练和测试视觉目标跟踪算法的大型数据集，其中训练集包含超过三万段视频序列，测试集包含 511 个视频序列。跟踪算法根据成功率曲线下面积（AUC）、跟踪精度（Prec.）以及归一化精度（$Prec._N$）这三个

表 4.8 基于数据集 TrackingNet[17] 本章提出的 SiamMask 算法与其他领先算法的跟踪结果对比展示：比较指标包括成功率曲线下面积（AUC）、跟踪精度（Prec.）以及归一化精度（Prec._N）。

Metric	SiamMask	ATOM[101]	MDNet[25]	CFNet[11]	SiamFC[7]	ECO[12]
AUC ↑	**72.5**	70.3	60.6	57.8	57.1	55.4
Prec. ↑	**66.4**	64.8	56.5	53.3	53.3	49.2
Prec._N ↑	**77.8**	77.1	70.5	65.4	66.3	61.8

指标进行排序。本节将 SiamMask 的性能与表现最好的领先跟踪器 ATOM[101]、MDNet[25]、CFNet[11]、SiamFC[7] 以及 ECO[12] 进行比较。表4.8中可以发现 SiamMask 在各项指标中均取得领先。相比于使用深度卷积网络进行在线优化学习的 ATOM，本章提出的孪生分割网络取得了 2.1% 的提升。这证明了本章提出的直接预测目标分割的策略要优于 ATOM 算法中所采用的深度回归网络迭代优化求解矩形框策略。算法在该数据集中的结果进一步支持了本章提出的 SiamMask 具有训练优异的分割分支以及较好的泛化能力。

4.3.2 半监督视频目标分割任务性能评估

本章所提出的孪生分割网络模型 SiamMask 不仅适用于视觉目标跟踪任务，同样可以应用于视频目标分割任务（Video Object Segmenation，简称 VOS）。算法在实现具有竞争力的精度的同时，不需要在测试时进行任何调整或在线训练。此外，与传统的 VOS 方法不同，本章方法可以实时运行，并且只需要简单的矩形边界框对跟踪器进行初始化。

数据集和设置。本小节报告了 SiamMask 在 DAVIS-2016[14]、DAVIS-2017[18] 和 YouTube-VOS[19] 基准上的表现。对于 DAVIS 数据集的评测，我们使用官方的精度度量：Jaccard 指数（$\mathcal{J}$）用来表示区域重叠相似性，F 度量（$\mathcal{F}$）用来表示算法轮廓精度。对于每个评价指标 $\mathcal{C} \in \{\mathcal{J}, \mathcal{F}\}$，实验分析同时考虑三个统计数据：均值指标 $\mathcal{C}_{\mathcal{M}}$、召回指标 $\mathcal{C}_{\mathcal{O}}$ 和衰减指标 $\mathcal{C}_{\mathcal{D}}$，最后一项指标表明算法性能随时间的变化梯度。对于 YouTube-VOS[19] 数据集，实验采用四个指标的平均值作为最终结果：$\mathcal{J}_{\mathcal{S}}$ 代表可见的类别重叠精度，$\mathcal{J}_{\mathcal{U}}$ 代表不可见的类别重叠精度，$\mathcal{F}_{\mathcal{S}}$ 代表可见的类别边缘精度，$\mathcal{F}_{\mathcal{U}}$ 代表不可见的类别边缘精度。

为了在视频目标分割场景中初始化 SiamMask，算法从第一帧提供的分割掩码中提取轴向对齐包围框（采用图4.5中的轴向对齐最小外接矩形框策略）。与大

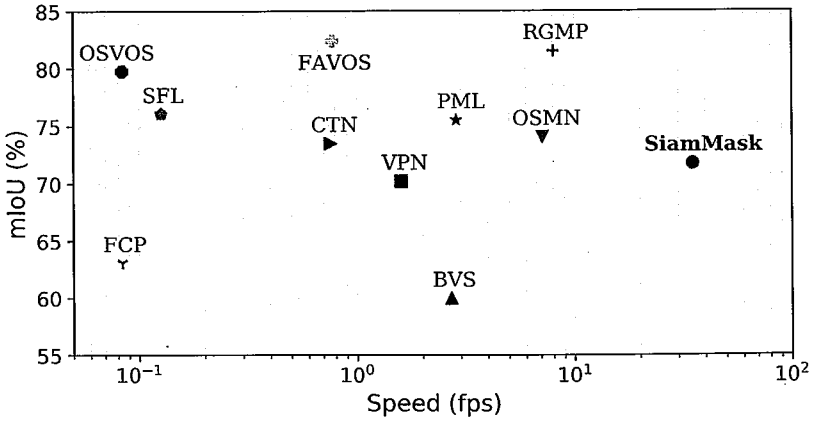


图 4.8 基于数据集 DAVIS-2016[14] 本章提出的 SiamMask 算法与典型视频目标分割算法的分割质量（平均重叠精度 mIoU）、运算速度（fps）对比展示。其中，x 轴采用对数刻度。

多数 VOS 方法类似，在同一个视频中有出现多个目标对象的场景（DAVIS-2017 和 Youtube-VOS），算法通过执行多个跟踪器分别进行跟踪分割。

视频目标分割算法通过第一帧的分割标签进行初始化，需要整合多种计算密集的技术进行预测。例如，算法使用在线精细训练 [151, 152, 156, 157, 165]、训练样本数据增强 [158, 159]、马尔科夫随机场或条件随机场 [122, 154–156] 以及光流 [124, 151, 154, 156, 159] 等后处理操作。因此，视频目标分割算法通常需要几分钟来处理一个短序列。显然，这些策略使得在线实用性受到影响。因此，在 DAVIS 和 YouTube-VOS 数据集进行结果对比的算法中，我们主要关注于最先进的实时运行算法。

图4.8展示了本章提出的孪生分割网络算法 SiamMask 与典型视频目标分割算法在 DAVIS-2016[14] 上的速度与精度对比。可以发现 SiamMask 取得与其他视频分割算法类似的精度，但是运行速度要快一个数量级，是首个可以达到实时运行（≥30fps）的视频目标分割算法。相比于采用元学习进行快速更新的 OSMN[161]，SiamMask 在线过程中不采用任何在线更新策略，通过目标跟踪进行分割，显著提升了分割速度。此外，与 OSMN 不同，我们通过向量化的分割编码同样加速了分割算法。

表4.9全面展示了各算法在DAVIS-2016中的性能。基于在线更新的OnAVOS[152]取得了最好的重叠精度以及轮廓精度。但是由于该算法使用在线模型更新，无法实时运行。本章所提出的算法相对其在速度性能上提升了 600 倍，同时仅需使

表 4.9 基于数据集 DAVIS-2016[14] 本章提出的 SiamMask 算法与其他领先算法的跟踪性能对比展示。其中，FT 和 M 分别表示是否在线训练以及使用分割标注结果（✔）进行初始化还是使用矩形框（✘）进行初始化；速度通过帧率（fps）进行度量。

Method	FT	M	$\mathcal{J}_{\mathcal{M}\uparrow}$	$\mathcal{J}_{\mathcal{O}\uparrow}$	$\mathcal{J}_{\mathcal{D}\downarrow}$	$\mathcal{F}_{\mathcal{M}\uparrow}$	$\mathcal{F}_{\mathcal{O}\uparrow}$	$\mathcal{F}_{\mathcal{D}\downarrow}$	Speed
OnAVOS[152]	✔	✔	**86.1**	**96.1**	5.2	**84.9**	**89.7**	5.8	0.08
MSK[151]	✔	✔	79.7	93.1	8.9	75.4	87.1	9.0	0.1
MSK_b[151]	✔	✘	69.6	-	-	-	-	-	0.1
SFL[165]	✔	✔	76.1	90.6	12.1	76.0	85.5	10.4	0.1
FAVOS[124]	✘	✔	82.4	96.5	4.5	79.5	89.4	5.5	0.8
RGMP[162]	✘	✔	81.5	91.7	10.9	82.0	90.8	10.1	8
SFL-ol[165]	✘	✔	67.4	81.4	6.2	66.7	77.1	5.1	3.3
PML[166]	✘	✔	75.5	89.6	8.5	79.3	93.4	7.8	3.6
OSMN[161]	✘	✔	74.0	87.6	9.0	72.9	84.0	10.6	8.0
PLM[160]	✘	✔	70.2	86.3	11.2	62.5	73.2	14.7	6.7
VPN[167]	✘	✔	70.2	82.3	12.4	65.5	69.0	14.4	1.6
SiamMask	✘	✘	71.7	86.8	**3.0**	67.8	79.8	**2.1**	**55**

用矩形框进行初始化。MSK_b[151] 同样采用了矩形框初始化并且进行在线网络精调，本章所提出的 SiamMask 由于构建了模板图像与搜索图像的关联学习，取得了 2.1% 的重叠精度提升。相比于近期提出的不使用在线更新的视频分割算法 FAVOS[124]、RGMP[162]、SFL-ol[165]、PML[166]、OSMN[161]、PLM[160] 和 VPN[167]，本章提出的方法初始化较为简单，具有较强的速度优势。SFL-ol 同时学习视频目标分割任务以及光流估计任务，利用光流信息聚合具有相同运动向量的像素点。相比于该算法，本章所提出的孪生分割网络关注于目标整体模板进行跟踪，利用图像的全局信息进行相似性度量与分割，在 DAVIS-2016 中的各项指标均优于 SFL-ol。此外，类似于孪生网络跟踪方法，PML[166] 算法同样采用特征网络进行相似性度量，但是由于其比较对象为图像像素，运行过程中需要构建超大规模的细粒度样本对，本章所提出的算法进行实例级别比对，取得了超过 10 倍的速度提升。除了精度的均值与召回率指标，我们注意到 SiamMask 在区域重叠率衰减（$\mathcal{J}_{\mathcal{D}}$）与轮廓精度衰减（$\mathcal{F}_{\mathcal{D}}$）两个指标方面都达到了最低数值，这表明我们的方法随着时间的推移是更加稳健的，因此适用于长序列跟踪分割。

表 4.10 基于数据集 DAVIS-2017[18] 本章提出的 SiamMask 算法与其他领先算法的跟踪性能对比展示。

Method	FT	M	$\mathcal{J}_{\mathcal{M}\uparrow}$	$\mathcal{J}_{\mathcal{O}\uparrow}$	$\mathcal{J}_{\mathcal{D}\downarrow}$	$\mathcal{F}_{\mathcal{M}\uparrow}$	$\mathcal{F}_{\mathcal{O}\uparrow}$	$\mathcal{F}_{\mathcal{D}\downarrow}$	Speed
OnAVOS[152]	✔	✔	**61.6**	**67.4**	27.9	**69.1**	**75.4**	26.6	0.1
OSVOS[168]	✔	✔	56.6	63.8	26.1	63.9	73.8	27.0	0.1
FAVOS[124]	✘	✔	54.6	61.1	**14.1**	61.8	72.3	**18.0**	0.8
OSMN[161]	✘	✔	52.5	60.9	21.5	57.1	66.1	24.3	8.0
SiamMask	✘	✘	54.3	62.8	19.3	58.5	67.5	20.9	**55**

表 4.11 基于数据集 YouTube-VOS[19] 本章提出的 SiamMask 算法与其他领先算法的跟踪性能对比展示。

Method	FT	M	$\mathcal{J}_{\mathcal{S}\uparrow}$	$\mathcal{J}_{\mathcal{U}\uparrow}$	$\mathcal{F}_{\mathcal{S}\uparrow}$	$\mathcal{F}_{\mathcal{U}\uparrow}$	$\mathcal{O}\uparrow$	Speed
OnAVOS[152]	✔	✔	60.1	46.6	**62.7**	51.4	55.2	0.1
OSVOS[168]	✔	✔	59.8	**54.2**	60.5	**60.7**	**58.8**	0.1
OSMN[161]	✘	✔	60.0	40.6	60.1	44.0	51.2	8.0
SiamMask	✘	✘	**60.2**	45.1	58.2	47.7	52.8	**55**

在 DAVIS-2017[18] 数据集中，我们取得了与 FAVOS[124] 类似的精度指标，同时带来了 60 倍的速度提升。FAVOS 为了适应目标的快速形变，采用多个全卷积孪生网络进行目标部件跟踪，利用多个跟踪结果分别分割前景，然后融合输出。本章提出的方法通过孪生网络直接输出分割预测，显著节约了跟踪时间。相比于使用元学习（meta-learning）进行在线快速更新的 OSMN 算法，本章算法在相关分析模块构建时增加了目标图像作为先验信息，具有更强的适应性。而 OSMN 的网络只包含单一输入，缺乏对于场景的适应性。在 DAVIS-2017 的重叠精度指标 $\mathcal{J}_{\mathcal{M}}$ 中，本章提出的 SiamMask 相比于 OSMN 提升了 1.8%。

在大规模视频分割数据集 Youtube-VOS[19] 中，本章所提出的 SiamMask 在可见类别场景中取得了最高分割精度，并且算法保持了当前最快的运行速度。

表4.9、表4.10和表4.11共同表明 SiamMask 可以视为视频目标分割任务的强大基线。首先，算法相比于 OnAVOS[152]、SFL[165] 等精确方法的速度提升了两个数量级。其次，SiamMask 与目前不采用在线微调的算法在精度方面具有竞争力，同时比最快的方法（即 OSMN[161] 和 RGMP[162]）的速度高出四倍。

表 4.12 本章提出的孪生分割网络算法部件在数据集 VOT-2018[2]、DAVIS-2016[14] 上的对照实验分析展示。

Method	AlexNet	ResNet-50	EAO ↑	$\mathcal{J}_{\mathcal{M}}\uparrow$	$\mathcal{F}_{\mathcal{M}}\uparrow$	Speed
SiamFC	✔		0.188	-	-	86
SiamFC		✔	0.251	-	-	40
SiamRPN	✔		0.243	-	-	**200**
SiamRPN		✔	0.359	-	-	76
SiamMask-2B w/o R		✔	0.326	62.3	55.6	43
SiamMask w/o R		✔	0.375	68.6	57.8	58
SiamMask-2B-score		✔	0.265	-	-	40
SiamMask-box		✔	0.363	-	-	76
SiamMask-2B		✔	0.334	67.4	63.5	60
SiamMask		✔	**0.380**	**71.7**	**67.8**	55

4.3.3 算法部件对照实验分析

在这一小节中我们对提出的孪生分割网络部件进行详细分析，同时对于网络无法适应的失败案例以及耗时分析进行全面总结。

网络架构分析：在表4.12中，算法分别采用 AlexNet[39] 和 ResNet-50[61] 作为共享特征主干 f_θ（参考图4.3）。同时，实验使用 "w/o R" 表示去除精细化分割输出模块（详见第 §4.2.2.4小节）。从表4.12我们可以得出以下结论：

- 表格的第一栏区域表明，通过将共享网络主干 f_θ 替换成更深的网络结构带来了巨大的性能提升。同时更换网络主干使得算法整体速度显著下降。
- 本章所提出的 SiamMask-2B 和 SiamMask 相对于它们各自的基线算法 SiamFC 和 SiamRPN 都取得了显著提升，这证明了分割表述对跟踪精度性能提升的有效性。
- 对于视频目标分割任务中轮廓精度指标 $\mathcal{F}_{\mathcal{M}}$ 而言，精细化分割模块至关重要。但精细化分割模块对于跟踪指标的提升并不明显。视觉目标跟踪目前的评价指标由于采用旋转矩形框表述进行比较，无法正确度量分割输出对于跟踪结果的提升。

多任务学习分析：我们通过进一步的实验来分析多任务训练的效果，结果在表4.12中报告。对照实验在跟踪推理阶段修改了 SiamMask 的两种变体。算法

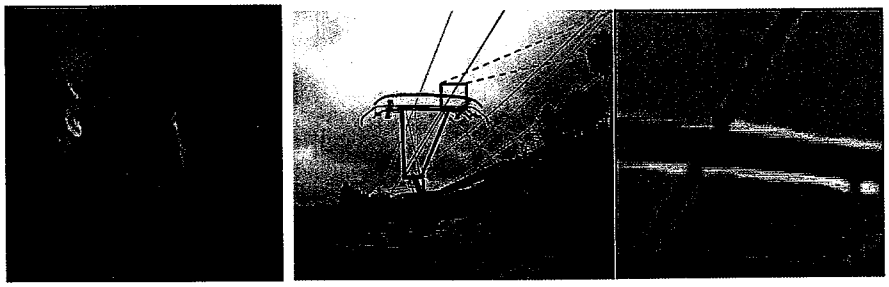


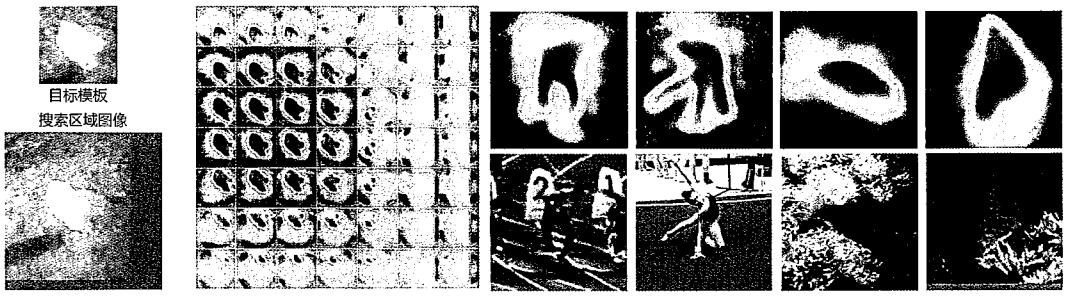
图 4.9 孪生分割网络跟踪失效的场景示例：运动模糊或者不具有语义的物体。

首先通过公式（4.4）和公式（4.5）进行多任务学习得到两分支孪生分割网络 SiamMask-2B 以及三分支孪生分割网络 SiamMask。在线跟踪过程中，去除算法的分割分支，得到仅包含分类分支的 SiamMask-2B-score 以及分类与矩形框回归分支的 SiamMask-box。因此，尽管这两个对照算法使用分割分支进行联合训练，但分割分支在推理过程中并没有被使用。我们可以观察到这两种变体相对于同样网络架构的 SiamFC 和 SiamRPN 取得了适当地提升。两分支变种在 VOT-2018 中的 EAO 指标从 0.251 提升到 0.265，三分支变种从 0.359 提升到 0.363。实验结果表明算法通过多任务学习吸收了精确分割表述信息，增强了矩形框回归分支与判别分支的训练。

网络耗时分析：在单个 NVIDIA RTX 2080 GPU 上，两个分支变体 SiamMask-2B 和三个分支变体 SiamMask 的平均运行速度分别为每秒 60 帧和每秒 55 帧。本章提出的 SiamMask 在线运行中不调整网络参数，其最大的计算负担来自特征提取器 f_θ。通过精巧的向量表示以及轻量化的网络设计，分割预测只增加了非常微小的耗时。该结果证明了本章所提出的分割分支的高效性。

4.3.4　算法失效场景分析

本节主要讨论孪生分割网络 SiamMask 跟踪失效的两种场景，当跟踪目标运动模糊或目标为不具有语义的物体（如图4.9）时，分割分支的预测结果会产生极大的不确定性。由于本章所提出的孪生分割网络架构采用离线训练，因而训练样本的分布会影响跟踪性能。这两种失效情况都是由于训练集中缺乏相似的训练样本而造成。对于运动模糊场景通常难以生成准确的分割标注，因而算法应加强离线训练过程中的数据增强来克服这种困难；对于无特殊语义的目标对象，基于离线学习的方法很难涵盖这种类别目标，因而需要在一定程度上与在线学习方法相结合来解决特殊表观目标的判别学习。



(a) 分割分支输出　　(b) 不同目标物体分割概率图

图 4.10 孪生分割网络的分割分支在不同位置的分割结果（a）以及孪生分割网络对不同目标对象的分割预测前景概率（b）示意图。

4.3.5 算法通用性验证

在跟踪推理过程中，算法依靠分类分支来判断目标所在的候选窗口，通过候选窗口对应的分割分支结果进行输出。为了更加清晰地了解算法的分割分支预测内容，我们首先可视化显示了不同候选窗口所预测的分割结果 $\mathbf{m}^n$。在图4.10a中显示了不同候选区域 RoW 所对应的分割结果，分割分支在多个窗口中对目标对象具有精确的分割预测。同时，通过不同候选窗口进行分割预测的方式有效降低了只采用单一位置预测的风险。图4.10b显示了孪生分割网络对于不同形态的目标对象的分割预测，我们可以看出本章算法对于目标对象具有较强的形态适应性。在嘈杂背景环境中，目标对象的分割结果也较为稳定。

为了定性分析算法在实际序列中的跟踪表示精度，我们使用本章提出的孪生分割网络对 VOT-2018[2] 和 DAVIS-2016[14] 数据集中具有挑战性的序列进行了可视化结果分析。算法分割结果分别如图4.11和图4.12所示。

对于视觉目标跟踪任务，在具有较大目标形变的 *butterfly*、*iceskater1* 视频序列中，本章所提出的方法可以正确、完整地分割出目标对象。该结果表明本章所提出的分割分支表述具有高度形态变化适应性。通过观察 *iceskater1* 视频序列的跟踪结果，我们可以发现在目标发生较大形变时，矩形框表示中会包含大量的背景信息，而分割表述可以精确地描述所要跟踪的目标对象。在 *crabs1* 与 *iceskater2* 视频序列中都存在较为邻近的干扰对象，算法的分割结果有效区分了不同目标物体。这一结果的实现来自于分割网络设计过程中的全局信息获取能力，使得算法有效区分了实例（instance）特征而非语义类别特征。此外，图像分割算法通常对于光照变化、运动模糊较为敏感，而本章所提出的孪生分割网络在

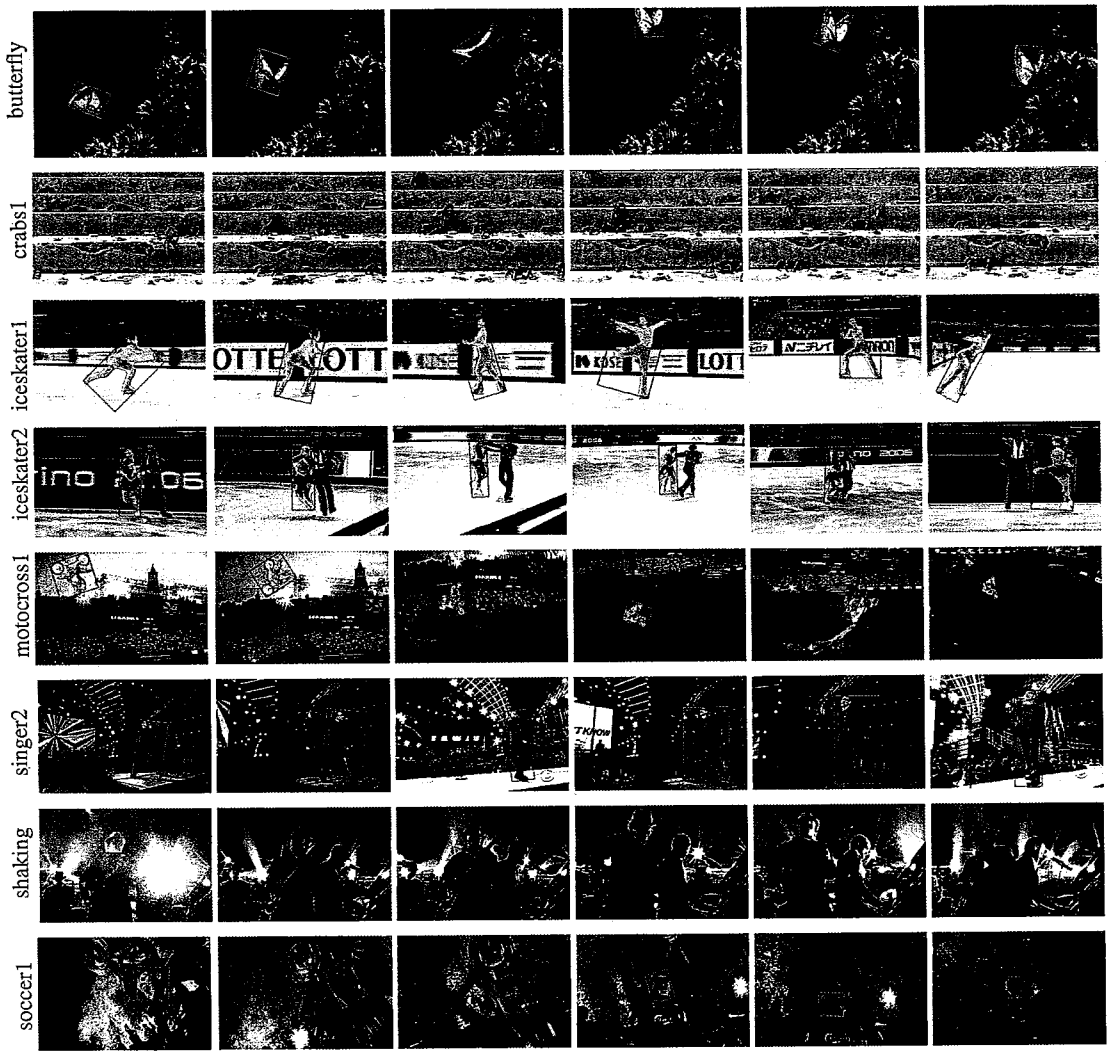


图 4.11 孪生分割网络 SiamMask 算法在数据集 VOT-2018[2] 的视频序列 *butterfly*、*crabs1*、*iceskater1*、*iceskater2*、*motocross1*、*singer2*、*shaking* 和 *soccer1* 的跟踪分割结果展示。

singer2、*shaking* 和 *soccer1* 均取得精确分割结果，证明了算法的鲁棒性。

在视频目标分割任务中，本章提出的孪生分割算法同样有效适应了目标的尺度、视角、形状等多种挑战。在 *Bmx-Trees* 以及 *libby* 视频序列中，目标发生了局部的遮挡，算法可以精确区分目标与障碍物的边缘。在 *drift-straight* 以及 *motorcross-jump* 中，目标对象的尺度以及视角均发生较大变化，跟踪结果表明孪生网络算法学习的相似性度量具有较好的旋转适应性。

通过这些实验结果的对比，说明了孪生分割网络的有效性与鲁棒性，也验证了本章提出的分割输出对于视觉目标跟踪任务的重要意义。

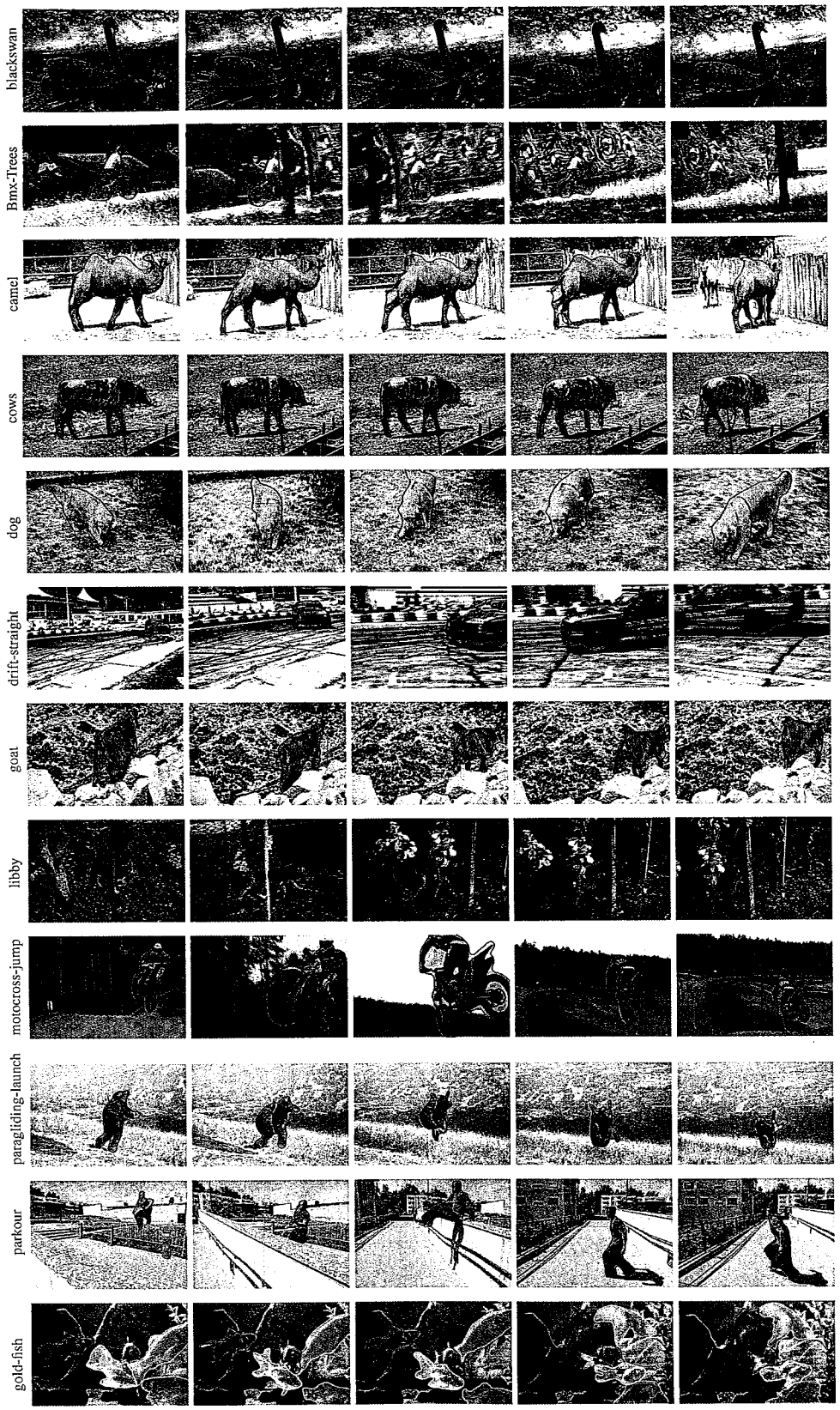


图 **4.12** 孪生分割网络 **SiamMask** 算法在数据集 **DAVIS-2016[14]** 的多个视频的跟踪分割结果展示。

4.4　孪生分割网络在视频实例分割领域的应用扩展

上文主要介绍了孪生分割网络跟踪算法在视觉目标跟踪以及视频目标分割领域的应用，并证明该算法的有效性。本小节进一步将其扩展至视频实例分割（Video Instance Segmentation，简称 VIS）任务，该任务主要特点是不提供标签信息用于初始化跟踪器，同时需要算法输出多目标的跟踪分割轨迹。任务要求在整个视频中分割定位任意数量的目标对象，并区分多个目标物体，为每个对象分配一个唯一的标识号，该标识号在不同帧之间保持一致。视频实例分割 [15] 与视频目标分割具有紧密联系，可以通过图像实例检测方法初始化视频目标分割算法进行统一。同时，视频实例分割相对于单目标视觉目标跟踪算法又极具挑战，需要算法同时跟踪多个目标对象。由于具有多目标跟踪的属性，我们同时借鉴多目标跟踪（MOT）[126] 的算法思路。算法框架采用基于检测的跟踪机制（tracking-by-detection），利用检测器结果构建网络节点，利用二分图匹配方式关联检测结果与跟踪轨迹，此外，多目标跟踪算法通常采用运动模型 [37] 估计目标轨迹在当前帧的位置，如果目标轨迹没有关联到检测结果时，需要使用运动模型估计的结果当作当前帧的目标位置。但是视频实例分割不同于通用的多目标跟踪任务，算法需要得到目标对象的分割输出，因而对检测算法与跟踪算法的输出均需要进行输出表示提升。具体而言，在检测过程中将通用的行人检测器转换为图像实例分割检测器；在多目标跟踪过程中，我们提出使用孪生分割网络 SiamMask 进行目标跟踪与分割。

4.4.1　基于孪生分割网络的视频实例分割框架

我们的方法借鉴了多目标跟踪（MOT[126]）中的基于匈牙利算法的帧间数据关联框架 [169]，将其迁移到视频实例检测问题中。图4.13概要地说明了我们

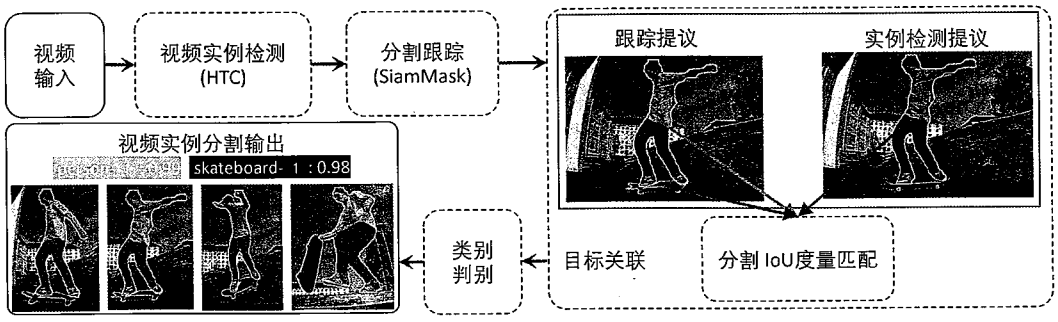


图 4.13 基于孪生分割网络 SiamMask 算法的视频实例分割流程。

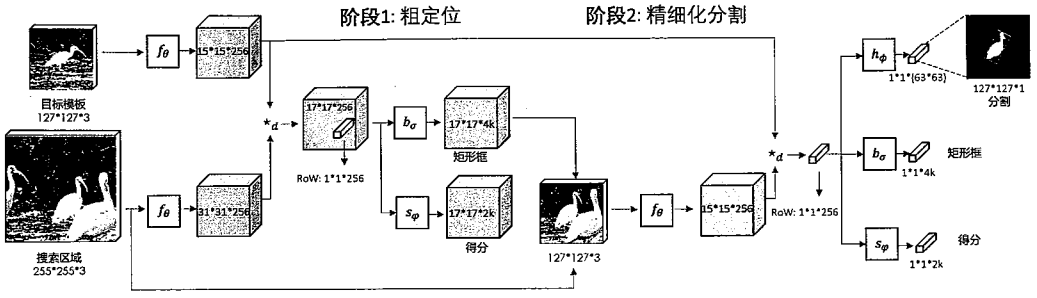


图 4.14 改进的两阶段孪生网络结构示意图：以级联的方式叠加两个原始的 SiamMask 模型组成，可提供更准确的定位和分割。

的算法流程。对于每一帧图像，我们首先使用图像实例检测器 HTC[170] 得到候选分割结果 $\mathcal{D} = \{\mathbf{d}_i\}_{i=1}^{N}$。对于现有的跟踪轨迹 $\mathcal{T} = \{\mathbf{t}_j\}_{j=1}^{M}$，我们通过孪生分割网络 SiamMask 估计轨迹段 $\mathcal{T}$ 在当前帧的分割结果 $\mathcal{P} = \{\mathbf{p}_j\}_{j=1}^{M}$。算法通过计算当前帧检测器得到的分割结果 $\mathcal{D}$ 与跟踪器预测的分割结果 $\mathcal{P}$ 的分割重叠比例距离计算分配代价矩阵。分配问题通过匈牙利算法 [169] 进行优化求解。

4.4.1.1 图像实例检测

文献 [171] 结果表明提高检测精度可以显著提升多目标跟踪算法的性能。在本节中我们使用 ResNeXt101-DCN[172] 作为检测网络主干，通过实例分割算法 HTC[170] 进行图像实例检测。在训练数据方面，为了增加训练数据多样性，我们扩展了 Youtube-VIS 训练数据集，采用 COCO[116] 和 OpenImage[173] 数据集作为辅助训练数据。在训练策略方面，除了采用随机缩放和裁剪进行数据增强外，我们还采用了 InstaBoost[174] 来进一步提高数据生成效率。

4.4.1.2 两阶段孪生分割网络

利用图像实例检测器生成候选分割结果 $\mathcal{D}$ 后，算法需要将目标对象的分割结果跨帧关联到跟踪轨迹集合 $\mathcal{T}$ 中。在多目标跟踪领域 [126]，算法通常使用卡尔曼滤波器 [37] 或视觉目标跟踪算法构建运动模型。但是由于视频实例分割在每帧中都需要输出分割表述，常规的跟踪算法无法满足分割输出要求，而本章提出的孪生分割网络可以高效运行并得到分割结果。此外，与基于运动模型（如 kalman filter）的跟踪器相比，SiamMask 可以提供更精确的目标状态估计。孪生网络算法分类分支预测的跟踪分数可以指示目标的遮挡或消失信息，分割分支预测的分割掩码可以降低检测器的漏检（missed detection）影响。

表 4.13 基于数据集 YouTube-VIS[15] 本节提出的两阶段孪生分割网络与单阶段孪生分割网络对照实验展示。其中，$\triangle_{mAP}$ 和 $\triangle_{AR10}$ 分别代表 mAP 和 AR@10 的性能提升百分比。

Model	mAP	$\triangle_{mAP}$	AR@10	$\triangle_{AR10}$
HTC+SiamMask	0.366	-	0.423	-
HTC+ 两阶段 SiamMask	0.381	+0.015	0.439	+0.016

在本节中，我们对第 §4.2.2小节提出的单阶段多分支的 SiamMask 进行级联改进，网络结构如图4.14所示。两阶段的孪生分割网络由两个以级联方式堆叠的单阶段 SiamMask 模型组成。在第一个阶段中，SiamMask 的矩形框回归分支生成了目标对象位置的初始位置估计 $\mathbf{p}^1$，在第二个阶段中利用 $\mathbf{p}^1$ 重新截取搜索区域用于预测对象的细化分割掩码。第二阶段的分割输出用来构建轨迹预测 $\mathcal{P}$，进行二分图匹配。对于匹配过程，我们设置了最低匹配阈值 IoU_{min}，当检测器与轨迹预测的类别相同时阈值为 0.7，否则为 0.4。

4.4.2　基于数据集 YouTube-VIS 的评测结果分析

4.4.2.1　数据集以及度量指标

本节实验主要是在最近发布的大规模视频实例分割 YouTube-VIS 数据库 [15] 的验证集与测试集进行评估，该数据集包含 2883 段高分辨率视频、40 种目标类别以及 13 万个高质量的实例分割标签。我们使用平均精度（mAP）和平均召回率（AR）这两个度量指标进行评估。

4.4.2.2　两阶段孪生分割网络性能提升分析

我们首先在 YouTube-VIS 的验证集上对比了采用单阶段的孪生分割网络以及本节提出的两阶段孪生分割网络框架，结果如表4.13所示。两阶段的跟踪算法首先利用检测分支进行目标的矩形框回归，降低了第二个阶段的分割难度。因此两阶段的孪生分割网络算法平均精度性能提升了 1.5%，目标召回率也提升了 1.6%。

4.4.2.3　在无监督视频目标分割中的性能评估分析

表4.14显示了本节所提出的算法框架在 YouTube-VIS 测试集 [15] 与其他参赛队伍的性能比较。与 YouTube-VIS[15] 官方提出的基线算法相比，本节所提出

表 4.14 基于数据集 YouTube-VIS[15] 本章提出的孪生分割网络算法框架和其他领先方法的跟踪性能展示。

队伍	mAP	AP50	AP75	AR1	AR10
Jono[175]	0.467	0.697	0.509	0.462	0.537
Our	**0.457**	**0.674**	**0.490**	**0.435**	**0.507**
bellejuillet[176]	0.450	0.636	0.502	0.447	0.503
mingmingdiii[177]	0.444	0.684	0.487	0.436	0.508
xiAaonice[178]	0.400	0.578	0.449	0.396	0.452
baseline[15]	0.313	0.503	0.338	0.335	0.369

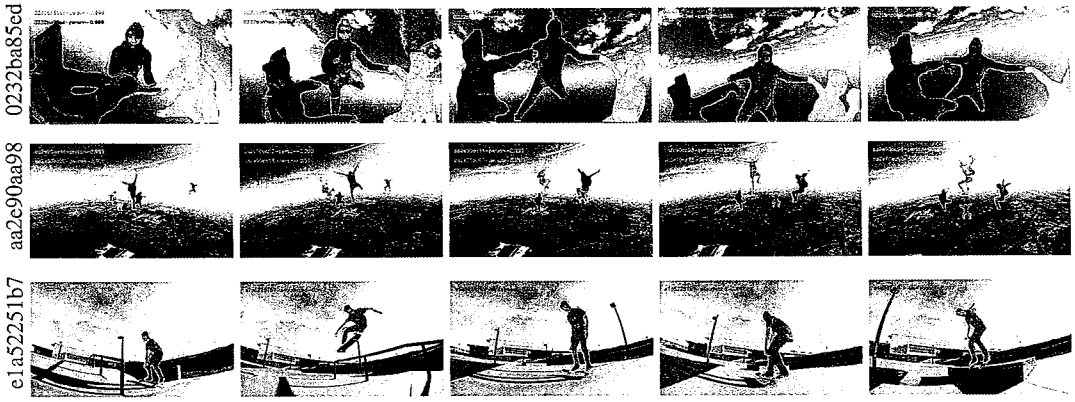


图 4.15 孪生分割网络 SiamMask 算法在数据集 YouTube-VIS[15] 的多个视频的跟踪分割结果展示。

的孪生网络分割架构在 mAP 指标下相对提升达到 46%。值得注意的是，我们的方法在大多数评价指标下都能获得第二好的性能。该结果证明了孪生网络分割框架在视频实例分割任务上具有良好的适应性。

4.4.3 孪生分割网络在视频实例分割领域的通用性验证

在图4.15中，我们可视化了本章提出的孪生分割网络 SiamMask 算法在 YouTube-VIS 具有挑战性的视频中的分割结果。这些视频场景覆盖了不同尺度的对象，并且目标经历了较大变形、快速运动以及物体之间存在相互遮挡。可视化结果表明，本章提出的孪生分割网络能够准确地分割多尺度目标以及快速运动的目标，具有较强的鲁棒性。

4.5　本章小结

在本章中，我们介绍了孪生网络分割框架 SiamMask，通过将分割分支引入到孪生网络架构，算法实现了对于视觉目标跟踪的输出模式扩展。对于分割分支的架构设计，算法采用向量化的分割表述方式获取目标全局信息，并提出自顶向下的堆叠精细化模块来增强分割细节。然后本章展示了将孪生网络分割框架应用于视觉目标跟踪和半监督视频目标分割任务，算法在视觉目标跟踪领域取得了当前最高的跟踪精度，同时，实现了当前最快的视频目标分割运行速度。最后，我们将孪生分割网络框架扩展到视频实例分割应用中，整体框架在具有挑战性的 YouTube-VIS 数据集上实现了较为领先的性能。

第5章 总结与展望

5.1 全文工作总结

本文主要围绕视觉目标跟踪问题进行了广泛而深入的研究，致力于解决孪生网络跟踪算法在特征提取、表观模型构建以及输出模式等方面的不足，通过对上述跟踪环节的持续提升改善了视觉目标跟踪算法对于光照变化、背景嘈杂、非刚体形变等挑战的适应性，同时严格约束算法在实时场景中的运行速度。本文贯彻使用数据驱动的方式进行视觉跟踪模型训练，通过推导传统相关滤波器的反向传播过程，实现端到端的特征学习，有效提升了深度特征表示的判别力；在多个视觉跟踪数据集上的测试结果表明，该模型显著优于采用手工设计特征以及预训练神经网络特征的跟踪算法，证明了端到端的特征学习的有效性。此外，针对孪生网络的表观模型，本文设计了融合多种注意力机制的自适应加权网络，通过解耦通用特征表示与判别学习的网络设计提升了算法的泛化性能以及目标判别适应性。同时本文将图像分割思想引入到目标跟踪对象的状态表述中，拓展了目标跟踪的表述形式，并首次构建了视觉目标跟踪与视频目标分割的一体化处理框架，建立了目标跟踪新范式。具体说来，本文主要完成的工作以及贡献点有以下三个方面：

第一，本文首先对判别相关滤波操作的反向传播过程进行详细推导，提出了端到端学习的判别相关滤波器跟踪算法，创新性地实现了深度特征自动提取与相关滤波判别模型的联合优化。由于特征表示的质量直接影响了跟踪器的性能，本文将视觉目标跟踪中常用的手工设计特征以及直接迁移预训练网络模型的特征表示方式改善为端到端学习的特征表示方式，有效提升了深度学习特征表示的判别能力，增加了特征表示网络设计的自由度，同时显著降低了算法的计算存储消耗。在端到端学习过程中，通过尺度-位移空间的联合学习，算法引入了尺度空间样本，进而可以提供更准确的目标尺度估计。在此基础上，本文又探索了基于深度特征表示的语义嵌入模型，并通过编解码孪生网络进行结构感知的自监督学习，提升了模型的泛化性能与细粒度特征表示能力。最后本文通过分别构建具有上下文感知能力的判别相关滤波器和自监督学习语义嵌入模型实现了具有互补性的跨层级特征融合表示与学习，显著提升了跟踪器的鲁棒性。

第二，本文重新形式化了判别式目标跟踪算法框架，将整体网络解耦为通用目标特征的表示网络以及用于判别学习的判别网络。本文通过理论分析证明了孪生网络算法的相关操作缺乏对不同位置的度量权重调整能力，因而提出了带有加权的交叉相关操作算子，利用该算子对目标不同空间位置的相关操作赋以自适应调整的权重。在模型离线训练阶段利用大规模视频数据联合学习判别相关损失以及目标区域的判别系数，实现了较强的表观形态适应性。本文通过注意力机制实现判别权重表述，并将其分解为用于统计整体样本分布的先验注意力机制、具有个体自适应性的残差注意力机制以及对于网络的不同语义层进行调整的通道注意力机制。算法通过注意力机制的引入，减少了训练过程中的过拟合，同时通过轻量化的网络设计，保证了良好的跟踪效率。最后，我们在多个具有挑战性的数据集中验证了本文提出的注意力机制对于视觉目标跟踪的有效性。

第三，本文提出孪生分割网络用于构建视觉目标跟踪与分割的一体化高效处理框架。本文从目标跟踪的表述形式出发，结合当前主流的视觉检测技术对于物体形态的表示方法，提出了对于目标状态的精确分割描述。本文通过引入一个独立的分割分支到全卷积孪生网络框架，使得孪生网络架构同时预测目标的矩形框位置以及精细的目标分割表述。在算法输出表述形式上，本文将孪生网络输出从低维度的相似性表述以及矩形框回归表述推广至高维度的分割向量表述，通过多任务联合学习提升了特征表示对于复杂形态的目标的感知能力。本文通过分析比较图像分割表述形式，提出使用具有全局信息的向量化表述方法进行分割结果估计。随后为了进一步精细化分割结果，本文提出自顶向下地堆叠精细化模块来增强分割细节，利用多层次特征不断修正目标分割边缘。在线跟踪过程中，算法只需要矩形框标注进行初始化，即可同时实现实时视觉目标跟踪与分割任务，具有较强的任务适应性。整个框架在完成多个任务的基础上，具有较高的分割效率，运行速度接近 55 帧每秒。最后，本文将上述一体化跟踪框架扩展到多目标跟踪场景，实现了无输入标签监督的多目标视频实例分割。我们在多个视频目标跟踪数据集、视频分割数据集以及视频实例分割挑战赛中对上述方法进行评测，实验结果一致证明了本文提出的孪生分割网络在精度性能上有较大幅度提升，同时算法取得了实时的跟踪速度。

整体来说，不管是第一项工作中采用端到端学习进行深度网络特征训练，还是第二项工作中利用注意力机制增强网络对目标对象的适应性。它们本质都是

利用深度学习方法改善目标的表观判别能力，因而针对于光照变化、平面旋转、运动模糊等跟踪挑战场景的跟踪稳定性有大幅度提升。但是由于这两种方法都采用了滑动窗口采样结合相似性度量输出的跟踪模式，缺乏对于目标复杂形变的适应性。为了增强对于目标跟踪问题的定义探索，第三项工作充分利用了深度神经网络具有的高模型容量特性，直接学习预测目标对象的分割表示。算法通过对于输出模式的升级，不仅带来了跟踪精度的提升，同时避免了大量的尺度空间采样带来的时间消耗，促进了视觉跟踪领域对于目标定义的研究。在最新的 VOT-2020 挑战赛[1]中，视觉目标跟踪的评价指标从矩形框的相似度转换为分割结果的相似性度量。

5.2　工作展望

尽管近几年来，视觉目标跟踪的研究取得了长足的发展，涌现出大量令人印象深刻的跟踪算法，但是这些研究主要围绕基于深度特征表示进行提升。基于深度学习的跟踪算法广泛借鉴了图像检测领域的先进结构设计 [28, 62]，而缺乏针对于视觉目标跟踪问题的领域先验研究。基于表观匹配的孪生网络算法对于非遮挡场景中的光照变化、尺度变化、平面内旋转、快速运动等场景具有较强的适应性，但对于遮挡、干扰对象交错的复杂场景缺乏足够的判别稳定性。本文的研究内容围绕孪生网络方法进行了比较深入的研究，提出了几种较为精确的跟踪估计算法。但本文的研究仍然以表观匹配为基础，缺乏时序鲁棒性。同时视觉跟踪对象通常存在于三维空间中，目前的研究主要考虑图像坐标系中的匹配判别分析，缺乏对立体空间信息的挖掘。在本文第二章中我们通过注意力机制改善目标的不同空间位置对于最终匹配的权重，但基于整体表观的模板匹配仍然难以适应目标对象的非刚性形变，算法对于目标的立体建模不够灵活。此外，深度神经网络也常常被当做黑箱使用，在实际应用中的决策形式通常缺乏可解释性。基于上述考虑，我们认为未来的研究工作可以从如下两个大的方面展开：

- 随着深度学习的发展，图像检测与视觉目标跟踪的技术边界越来越近，许多基于检测的跟踪算法（tracking-by-detection）取得了良好的跟踪性能。然而视觉目标跟踪较图像检测任务的核心区别在于其研究对象随着时间不断变化，时序信息是其重要的研究对象。在本文中的孪生网络算法通常采用图像样本对

[1]https://www.votchallenge.net/vot2020

(pairwise) 的方式进行度量学习，这种图像对的训练方式忽视了时序变化的建模。一方面是由于卷积神经网络在时序建模方向的研究较为初步，而视觉目标跟踪算法大都需要深度学习底层部件的支持；另一方面，时序模型训练相对于表观模型训练需要更大量的数据支撑，现有视觉目标跟踪数据集的数据量不足，直到近期才出现了针对视觉目标跟踪的大规模数据集 [16, 17, 133]。本文提出的第一项工作中试图利用增量方式更新滤波器学习，但这种线性加权的更新方式无法对于场景中的上下文环境、以及目标的运动状态进行动态时序调整。研究人员 [113] 尝试使用长短时记忆网络对时序模板进行更新，但其时间依赖关系容量较低。针对本文所提出的孪生网络学习方法，研究人员应该尝试通过自监督学习的方式进行时序关系学习。

- 本文所提出并使用的孪生网络算法主要以挖掘图像表观信息进行度量学习，而该表观建模过程中缺乏对于物体的立体空间环境感知。视觉目标跟踪对象通常在现实的立体空间中运动，仅利用物体在图片坐标系的表观信息难以实现在障碍物遮挡等困难场景中的稳定跟踪。我们可以增加对立体空间中的深度信息建模来改善目标表观信息不足。除了物体本身的深度信息，不同物体在立体空间中的几何位置关系以及关联关系同样可以被视觉目标跟踪器所利用。文献 [179] 利用图模型构建不同物体之间的关联关系，在多目标跟踪中取得了有效提升。此外，当前视觉目标跟踪中物体的表观特征表示形式过于单一，通常采用向量化表示对目标构建整体模型表示，通过欧氏距离或余弦距离进行度量。这种整体模板匹配方式本质上难以有效建模目标对象在立体空间中的旋转以及非刚性形变。在固定类别的行人或车辆跟踪领域，对于目标对象可以采用立体模型进行表述。尽管现阶段对于通用物体的立体表述较难实现，但目标在时序运动过程中不断提供了多视角下的视图，该信息可以用作对目标物体进行重建。通过对于目标对象的立体感知，算法可以有效解决跟踪中因视角变化或目标旋转变化带来的错误估计，通过图形学方法对目标对象建模是一个值得关注的研究方向。