



中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

复杂场景下基于深度特征匹配的目标跟踪算法研究

作者姓名: 李荅群

指导教师: 乔彦峰 研究员

余毅 研究员

中国科学院长春光学精密机械与物理研究所

学位类别: 工学博士

学科专业: 光学工程

培养单位: 中国科学院长春光学精密机械与物理研究所

2021 年 6 月

Research on Object Tracking Based on
Deep Feature Matching in Complex Scenes

A dissertation submitted to
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Engineering
in Optical Engineering

By

Li Daqun

Supervisor : Professor Qiao Yanfeng

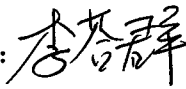
Professor Yu Yi

Changchun Institute of Optics, Fine Mechanics and Physics,
Chinese Academy of Sciences

June 2021

中国科学院大学
研究生学位论文原创性声明

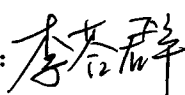
本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

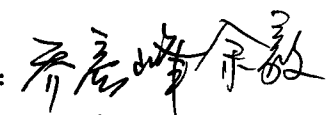
作者签名：
日期：2021.5.25

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有关保存和使用学位论文的规定，即中国科学院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：
日期：2021.5.25

导师签名：
日期：2021.5.25

摘 要

作为图像处理、计算机视觉领域中重要的研究方向之一，目标跟踪在诸如智能交通、视频监控、视觉导航、国防侦查和军事观测等应用领域具有很高的应用价值。尽管在之前几十年的研究中目标跟踪已经取得较好的发展，但是其仍然面临很多难题。一方面，在跟踪过程中目标经常会发生较为复杂的变化，比如光照变化、形变、遮挡等问题，这使得稳定且准确地跟踪目标变得尤为困难。另一方面，如何全面且有效地对目标外观表征模型、目标定位方法及模板更新策略等重要环节进行提升，也时刻困扰着国内外众多学者。

近年来，基于深度学习的目标跟踪算法相比于其它目标跟踪算法展现出更加优异的性能，其在大数据的驱动下利用端到端学习不仅能够高效、便捷地训练模型，还极大地提升了算法的鲁棒性与准确性。其中，基于深度特征匹配的目标跟踪算法以其均衡的准确性、鲁棒性和实时性获得了较多的关注。虽然这类算法在很多方面表现优异，但由于网络结构中缺少层级间的自适应特征融合、模板更新策略不够完善、缺少再检测机制等原因，其在复杂场景下的鲁棒性与准确性都会明显降低。因此，本论文在研究大量国内外目标跟踪算法的基础上，针对复杂场景下基于深度特征匹配的目标跟踪算法的局限性进行一定的改进。本论文主要开展可见光单目标跟踪算法的研究，主要研究内容和创新性成果如下：

- 1、针对基于深度特征匹配的目标跟踪算法在复杂场景下准确性及鲁棒性降低且模板更新策略不够完善的问题，提出一种结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。在跟踪过程中，当背景中出现相似干扰时，基于深度特征匹配的目标跟踪算法生成的响应图中易出现多个波峰，从而造成跟踪框不稳定甚至跟踪目标丢失的情况。本文利用一种高效且准确的再检测孪生卷积神经网络作为再检测机制来应对响应图中的多波峰。再检测网络在出现的各个波峰周围采样并进行实时、准确的匹配计算，从而实现对于干扰的筛选与排除。另外，为了应对跟踪过程中遮挡、形变等复杂因素带来的影响，本文提出一种基于生成式模型的自适应模板更新方法。该方法将评估过的可靠跟踪结果进行特征提

取与概率统计，利用统计结果与已有的目标模板特征进行自适应融合，从而实现模板的自适应更新。仿真实验结果表明，该算法能较好地应对相似背景干扰、遮挡、目标形变等难题，有效提升复杂场景下的跟踪精度与鲁棒性。

2、针对基于深度特征匹配的目标跟踪算法网络结构较浅且缺少层级间的自适应特征融合的问题，提出一种结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。虽然，基于深度特征匹配的目标跟踪算法利用浅层卷积神经网络和离线训练能够在一些低复杂度的场景取得不错的跟踪效果。但是，由于缺少丰富的语义特征，其在应对复杂场景下一些具有挑战性的难题时，跟踪效果将会明显下降。本文利用一种改进的残差网络来替代传统的浅层网络，从而提取出具有丰富语义信息的深层特征。与此同时，本文提出一种高效的自适应多层特征融合策略，有效提升了响应图的质量，减少了在复杂场景下相似背景干扰所带来的影响。另外，本文利用一种自适应特征信息融合方法来实现模板的更新，从而有效克服遮挡、目标形变等难题。仿真实验结果表明，该算法能够有效提升跟踪效果，并且在准确性与鲁棒性等方面表现良好。

3、针对基于深度特征匹配的目标跟踪算法前景信息利用不充分，跟踪过程中在语义背景下易丢失目标的问题，提出一种前景信息引导的孪生卷积神经网络目标跟踪算法。考虑到现有的基于深度特征匹配的目标跟踪算法在离线训练时正样本对包含的挑战因素较少，在目标遇到遮挡或较严重形变时跟踪效果下降的问题，本文采用一种相对简单的采样策略来扩充离线训练时正样本对中的挑战因素，从而有效增强算法的鲁棒性。并且，为了进一步提升算法在语义背景下的识别能力，本文将背景信息进行填充遮挡，从而增强前景信息的显著性。与此同时，将处理后的数据输入到基于卷积神经网络的引导分支中，并运用一种填充损失计算方式来改进算法的损失函数，有效提升算法在语义背景下的识别能力。此外，本文采用一种改进的特征信息融合方法来实现模板的更新，进一步应对目标的外观变化。仿真实验结果表明，该算法在多种干扰因素下取得了较好的跟踪效果，优于所对比的其它多个先进的跟踪算法。

关键词：目标跟踪，深度特征匹配，孪生卷积神经网络，多层特征融合，前景信息引导

Abstract

As one of the important research directions in the field of image processing and computer vision, visual object tracking has high application value in the fields such as intelligent transportation, video surveillance, visual navigation, national defense investigation and military observation. Although object tracking has achieved obvious development in the previous decades of research, it still faces many difficulties. On the one hand, in the tracking process, the object often undergoes various complicated changes, such as illumination variation, deformation, occlusion, etc., which make it particularly difficult to track the object stably and accurately. On the other hand, how to comprehensively and effectively improve the appearance model, the object location method and the template update strategy has always troubled many scholars at home and abroad.

In recent years, object tracking based on deep learning shows better performance than other algorithms. Driven by the big data and the end-to-end learning, it can not only train the model efficiently and conveniently, but also greatly improve the robustness and accuracy of the algorithm. As one of the most representative directions, object tracking based on deep feature matching has attracted much attention for its balanced accuracy, robustness and real-time performance. Although this type of algorithm performs well in many aspects, its robustness and accuracy in complex scenes will be significantly reduced due to the relatively simple network structure, the insufficient template update strategy, and the lack of re-detection mechanism. Therefore, based on the research of a large number of object tracking algorithms at home and abroad, the dissertation makes certain improvements to the limitations of the object tracking based on deep feature matching in complex scenes. The dissertation focuses on the research of visible light single object tracking algorithm. The main contributions of the dissertation are listed as follows:

- 1.To improve the accuracy and the robustness of the object tracking based on

deep feature matching in complex scenes, a Siamese network combining the re-detection mechanism and the adaptive template updating method is proposed. In the tracking process, when similar interference appears in the background, the response map generated by the Siamese network is prone to appear multiple peaks, which may lead to the tracking failure. In this dissertation, an efficient and accurate Siamese convolutional neural network is utilized as the re-detection mechanism to deal with the multiple peaks. The re-detection network can sample around each peak in the response map, and perform accurate matching calculations. Thus, the elimination of the interference can be realized. In addition, in order to deal with the influence of occlusion, deformation and other complex factors in the tracking process, this dissertation proposes an adaptive template updating method based on the generative model. In this method, the evaluated reliable tracking result is utilized to extract the features and perform the probability statistics. The statistical results are combined with the existing template features adaptively, so as to realize the adaptive updating of the template. The simulation results demonstrate that the proposed algorithm can effectively deal with the challenging problems and improve the tracking performance in complex scenes.

2. In most object tracking algorithms based on deep feature matching, the structure of the network is shallow and the convolutional features are not fully utilized. Although most trackers can achieve excellent tracking performance in some simple scenes, they can not perform well in some complex scenes, which have many challenging problems (Occlusion, Deformation, etc.). In this dissertation, an improved ResNet network is used to replace the traditional shallow network, and the deep features with more semantic information can be extracted. At the same time, an adaptive multi-layer feature fusion strategy is adopted to improve the quality of the response map. Thus, the algorithm can significantly reduce the impact of the distractors in complex scenes. In addition, an adaptive feature information fusion is adopted to update the template, so that the algorithm can adapt to various changes of

the target appearance. The simulation results show that the proposed algorithm can effectively improve the tracking performance and perform favorably in both robustness and accuracy.

3. In most object tracking algorithms based on deep feature matching, the foreground information is not fully utilized and the tracking performance decreases significantly in the semantic backgrounds. For effectively discriminating the object from the semantic backgrounds, a Siamese tracker combining foreground information guidance is proposed in this dissertation. Considering the tracking algorithms based on deep feature matching have limited challenging factors in the positive pairs, an effective sampling strategy is utilized to enrich the factors in the offline training phase. In addition, in order to further improve the discriminating ability of the tracker, the foreground information is highlighted by padding the background. At the same time, a guidance branch is combined to extract the features of foreground and a novel padding loss is adopted to guide the tracker. Furthermore, an improved feature information fusion is used to update the template, so that the algorithm can adapt to various changes of the target appearance. The simulation results demonstrate that the proposed algorithm can effectively deal with the challenging problems and improve the discriminating ability in semantic backgrounds.

Keywords: object tracking, deep feature matching, Siamese convolutional neural network, multi-layer feature fusion, foreground information guidance

目 录

摘 要.....	I
Abstract	III
目 录.....	VII
第 1 章 绪 论.....	1
1.1 课题研究背景及意义.....	1
1.2 目标跟踪算法研究现状.....	4
1.2.1 传统目标跟踪算法.....	4
1.2.2 基于深度学习的目标跟踪算法.....	8
1.3 复杂场景下目标跟踪面临的挑战.....	12
1.4 论文主要内容及结构安排.....	15
第 2 章 目标跟踪与深度学习基础理论	21
2.1 目标跟踪概述.....	21
2.2 深度学习基本原理.....	26
2.2.1 深度学习基本结构.....	26
2.2.2 卷积神经网络基本原理.....	31
2.2.3 几种典型的卷积神经网络模型.....	34
2.3 孪生卷积神经网络.....	39
2.3.1 孪生卷积神经网络概述.....	40
2.3.2 SiamFC 目标跟踪算法概述	41
2.4 测试数据集及评价指标.....	44
2.4.1 OTB 数据集及评价指标.....	44
2.4.2 VOT 数据集及评价指标	46
2.5 本章小结.....	47
第 3 章 结合再检测机制与自适应模板更新的孪生卷积神经网络	
目标跟踪算法.....	49

3.1 引言.....	49
3.2 结合再检测机制与自适应模板更新的孪生卷积神经网络.....	49
3.2.1 孪生卷积神经网络.....	50
3.2.2 再检测机制.....	51
3.2.3 自适应模板更新.....	55
3.2.4 算法描述.....	57
3.3 实验结果与分析.....	58
3.3.1 OTB 数据集实验分析.....	60
3.3.2 VOT 数据集实验分析	66
3.3.3 消融实验.....	67
3.4 本章小结.....	68
第 4 章 结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法.....	69
4.1 引言.....	69
4.2 结合深层特征与鲁棒特征融合的孪生卷积神经网络.....	69
4.2.1 改进结构的残差网络.....	70
4.2.2 多层特征融合.....	72
4.2.3 基于特征信息融合的模板更新.....	74
4.2.4 算法描述.....	76
4.3 实验结果与分析.....	77
4.3.1 OTB 数据集实验分析.....	78
4.3.2 VOT 数据集实验分析	84
4.3.3 消融实验.....	85
4.4 本章小结.....	87
第 5 章 前景信息引导的孪生卷积神经网络目标跟踪算法	89
5.1 引言.....	89
5.2 前景信息引导的孪生卷积神经网络.....	90
5.2.1 正样本对的扩充.....	90
5.2.2 前景信息引导.....	92
5.2.3 基于改进特征信息融合的模板更新.....	94

5.2.4 算法描述.....	96
5.3 实验结果与分析.....	97
5.3.1 OTB 数据集实验分析.....	98
5.3.2 VOT 数据集实验分析	102
5.3.3 消融实验.....	104
5.4 本章小结.....	105
第 6 章 本文算法对比实验分析.....	107
6.1 OTB 数据集对比实验分析.....	107
6.2 VOT 数据集对比实验分析	110
6.3 实际应用场景对比实验分析.....	112
6.4 运行速度分析.....	115
6.5 本章小结.....	116
第 7 章 总结与展望.....	117
7.1 论文工作总结.....	117
7.2 研究展望.....	118
参考文献.....	121
致 谢.....	133
作者简历及攻读学位期间发表的学术论文与研究成果	135

第1章 绪论

1.1 课题研究背景及意义

作为获取和理解外界信息的重要途径,视觉在人类的生存和发展过程中扮演着非常重要的角色。与人类一样,计算机视觉在人类设计的机器中也占据着非常重要的地位。计算机视觉旨在研究如何运用计算机和摄像机对目标进行处理、跟踪、分析、识别等,从而在一定程度上代替人眼。它不仅巧妙融合计算机科学、信号处理及应用数学等领域的知识,并且随着深度学习的迅猛发展,其已经在人工智能领域中成为一个极具应用价值与发展前景的分支,引来大量研究人员及资本的关注。根据 Tractica 预测,在 2022 年,与计算机视觉产品相关的市场将会达到前所未有的新高度,同时将有大量的资金投入相关技术与产品的研发中。目前,我国已经有 100 多家先进的计算机视觉企业,且拥有全球前三的计算机视觉市场,这不仅有效促使着社会经济的发展,还为新一轮的科技革命和产业变革积蓄着巨大能量。

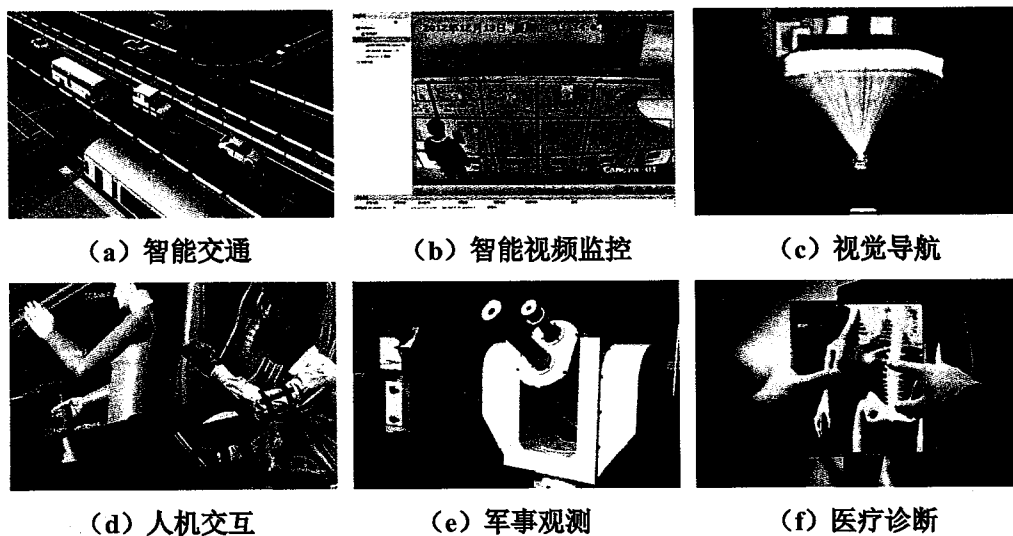


图 1.1 目标跟踪技术在部分领域的应用

Figure 1.1 The application of the object tracking in some fields

作为计算机视觉领域非常重要的组成部分和关键技术之一,目标跟踪在近些年成为了一个备受关注的研究课题。给定目标在视频或图像序列第一帧中的状态,

在后续过程中逐帧处理并预测目标的大小、位置及运动轨迹等信息。这些信息不仅可以用作场景理解、精确定位和行为识别等需求的基础元素，还能实现对视频或图像序列的高层次理解^[1,2]。近些年，随着大数据和人工智能技术的发展，目标跟踪已经在智能交通、智能视频监控、视觉导航、人机交互、军事观测和医疗诊断等领域得到了广泛的应用。图 1.1 简要展示了目标跟踪技术在部分领域的应用。

智能交通^[3-5]：智能交通主要是将信息技术、传感器技术、自动控制理论和人工智能等科学技术综合运用于服务控制和交通运输等方面，通过车、人、路的和谐配合提高交通运输效率，有效缓解交通阻塞并提高路网的通过能力，减少环境污染和交通事故。其中，目标跟踪在智能交通系统中主要用于对车辆和行人等进行精确定位，从而为后续的行为识别、智能管控和车辆缉查提供有效信息。

智能视频监控^[6-9]：近些年，随着经济和社会的快速增长与进步，交通、银行、安检以及军事等领域对安全防范和监控报警的需求与日俱增，功能要求也是越来越高，传统的完全依靠人力的视频监控技术费时费力，已经无法完成对海量数据的分析工作并满足其智能化的需求。作为智能安防的主要组成部分，智能视频监控系统在传统的视频监控技术的基础上融入了计算机视觉，从而智能地分析、跟踪和判别监控对象。其中，目标跟踪在智能视频监控系统中主要用于监控对象的精确定位，为安防、缉凶和侦查等警戒工作提供可靠且高效的帮助。并且，目标跟踪技术还能够实现运动轨迹提取，从而对目标的动作和行为进行智能分析，实现对异常行为或其它危险状况的预警。

视觉导航^[10-14]：视觉导航主要是将雷达、高清摄像头和激光等各类传感器收集的信息综合运用，结合计算机视觉技术实现精确导航与智能避障。其中，目标跟踪在视觉导航系统中主要用于实时计算周围物体的大小、位置、方向和移动速度等，从而实现快速分析并做出合理响应。目前，融合多种信息的视觉导航系统已经在武器精确制导、智能机器人、无人驾驶等领域广泛应用。

人机交互^[15-18]：在传统的人机交互中，人们通常依靠鼠标与键盘等简单的外接设备实现交互。近些年，随着计算机技术与网络的发展，计算机与人们的交互变得更加直观和自然，不仅能够利用摄像头等传感器实时地捕获使用者的手势、

动作和表情,还能根据这些关键信息做出及时的反馈。作为人机交互系统的核心技术之一,目标跟踪主要用来跟踪使用者的特定部位,从而对手势、姿态和行为进行识别和准确分析并实现智能交互。

军事观测^[19-22]:目前在军事观测领域,目标跟踪主要用于靶场观测、战地侦查和精确制导等。根据外引导数据、理论弹道数据和先验参数精确计算出目标位置,并对运动轨迹、速度和姿态进行分析,从而实现智能观测与精确制导。

医疗诊断^[23-25]:随着医学领域的发展,医疗数据的数量也在急剧增长,单纯依靠人工分析的方法来进行诊断已经无法满足现实的需求。医疗诊断可以辅助医生从海量的医疗图像数据中定位具有特定病理特征的病例,不仅可以有效提高诊断效率,还能增加诊断的准确性。其中,目标跟踪在医疗诊断中主要用于病理特征的精确定位,从而更加准确、高效地进行病理分析与诊断。

综上所述,无论是在民用领域还是在军事领域,目标跟踪都是一项不可或缺的关键技术。尤其是近些年,随着大数据和人工智能的快速发展,目标跟踪的应用价值与发展前景更加凸显。

一项技术的价值不仅体现在其应用前景上,还能通过学术上的研究热度直接进行考量。近些年,目标跟踪技术作为学术研究领域热度较高的方向之一,每年都会有与其相关的大量学术文献在各大知名期刊或会议上进行发表。其中,一些著名的期刊包括 Elsevier 旗下的《Pattern Recognition》、《Image and Vision Computing》和《Neurocomputing》等,IEEE 旗下的《IEEE Transactions on Pattern Analysis and Machine Intelligence》、《IEEE Transactions on Signal Processing》、《IEEE Signal Processing Letters》和《IEEE Transactions on Image Processing》等,以及 Springer 旗下的《International Journal of Computer Vision》等。著名的三大顶级会议包括 CVPR (Conference on Computer Vision and Pattern Recognition)、ICCV (International Conference on Computer Vision) 和 ECCV (European Conference on Computer Vision)。此外,从 2013 年开始,由 ECCV 和 ICCV 轮流举办的目标跟踪大赛 VOT (Visual Object Tracking Challenge) 受到了国内外学者的广泛关注。其不仅提供了一个用于讨论全新目标跟踪技术的平台,还评选出优秀的目标跟踪算法供研究人员参考。由此可见,目标跟踪在世界范围内都具有十分重要的

研究价值。

1.2 目标跟踪算法研究现状

通过对目标跟踪算法应用价值和研究价值的分析可以发现,计算机视觉的急速发展为目标跟踪技术带来了前所未有的机遇。人工智能、大数据、高性能处理芯片、物联网的应用,不仅为目标跟踪提供了可靠的支持,同时也促进其向更高、更智能、更准确的方向迈进。但是,在目标跟踪的发展道路上依然面临很多难题。一方面,在跟踪过程中目标经常会发生较为复杂的变化,比如光照变化、形变、遮挡、运动模糊等问题,这使得稳定且准确地跟踪目标变得尤为困难。另一方面,如何在复杂场景下全面且有效地提升跟踪器的性能也困扰着国内外众多学者。为此,国内外研究人员自主提出许多先进的目标跟踪算法。这些算法按照不同的规则可以分为不同的类别:按照目标大小的不同,可以分为面目标跟踪算法、线目标跟踪算法和点目标跟踪算法;按照跟踪目标数量的不同,可以分为多目标跟踪算法和单目标跟踪算法;按照跟踪时长的不同,可以分为长时目标跟踪算法和短时目标跟踪算法;按照目标外观表征模型的不同,可以分为基于判别式模型的目标跟踪算法和基于生成式模型的目标跟踪算法;按照特征提取与描述方式的不同,可以分为传统目标跟踪算法和基于深度学习的目标跟踪算法。接下来,本节将分别从传统目标跟踪算法和基于深度学习的目标跟踪算法两大类,进行可见光单目标跟踪算法研究现状的总结。

1.2.1 传统目标跟踪算法

传统的目标跟踪算法广义上可以分为生成式模型和判别式模型两大类。其中,生成式模型旨在学习目标的外观变化,通过计算目标模板与候选样本之间的相似度将目标从背景中识别出来。常见的生成式模型有:子空间建模^[26-34]、稀疏表示^[35-42]和核密度估计^[43-45]等。

子空间建模是将目标看作一个整体而不是单类特征的集合,通过将目标信息映射到低维子空间来构建一个低复杂度、低冗余性、高判别力的数据表述空间。1996年,Black等人^[26]首次将子空间建模应用到目标跟踪算法中。通过在前期采集大量的目标样本来线下学习能够表示目标的特征子空间,从而运用这些子空间

实现自动跟踪。然而在算法运行过程中,这些子空间并没有进行实时的在线更新,当运用到现实场景时,算法的跟踪效果并不理想。在此之后,一些学者提出基于多增量主成分分析的算法^[27-29],而这些算法在学习子空间时并不能对样本的均值进行更新。为了克服这一问题,Ross 等人^[30]提出了 IVT (Incremental Visual Tracking) 算法。该算法在每一帧中都对特征向量进行增量地更新,并且基于 R-SVD (Rank-R Singular Value Decomposition) 构建出能够进行样本均值在线更新的子空间模型,从而在复杂场景下提升算法的鲁棒性。但是,前面描述的算法都是在一维基础上的子空间建模,当目标特征的维度较高且样本数量较少时,需要采用高维子空间建模来提升跟踪效果。因此,Wang 等人^[31]通过分析二维图像矩阵来构建基于二维主成分分析(2DPCA)的外观模型,Lu 和 Wang^[32]将 2DPCA 重构与 L1 正则化约束结合,有效地提升了跟踪效果。随着研究的深入,研究人员们发现基于子空间建模的算法都是假设目标外观数据处于线性空间中的,而实际跟踪过程中目标外观多处于非线性空间,并不能满足该假设。因此,一些学者通过构建非线性子空间模型来应对更复杂的场景。Lim^[33]等利用非线性降维成功将非线性子空间模型应用到行人跟踪中。Chin 和 Suter^[34]利用 KPCA (Kernel Principal Component Analysis) 构建一个更通用的非线性子空间模型,一定程度上提升了算法在部分遮挡、形变及相似背景干扰下的跟踪性能。

稀疏表示旨在远小于奈奎斯特采样频率的条件下,运用随机采样来获取图像信号的离散样本,然后通过非线性重建算法对信号进行重建。Mei 和 Ling^[35]首次将稀疏表示模型引入到跟踪算法中,并运用一组单位基向量将遮挡等干扰进行线性表示。该算法利用目标模板和单位基向量构建一个过完备字典,从而可以利用该字典对目标进行线性表示。为了实现稀疏表示,在求解系数时运用 L1 正则项使系数稀疏化并运用 LASSO 算法^[36]求解。这样,只要目标存在于候选样本中,无论是被干扰或是遮挡,都能运用模板和单位基向量进行线性的表示。此后,众多学者以此稀疏表示模型为基础进行不同的改进,不断提升跟踪算法的准确性与鲁棒性。Bao 等人^[37]在研究中发现,单位基向量不仅可以表示遮挡等干扰因素,还可以用来表示目标。该算法在单位基向量的系数中引入一个 L2 正则化项,并在目标没有遮挡的时候开启,从而使单位基向量系数的能量得到抑制。此外,为

了提升求解最优化模型的速度,该算法还提出一种近端梯度加速方法来进行快速求解。虽然,基于稀疏表示的算法能够很好地应对遮挡,但在目标形变、旋转等方面表现较差。为此,Wang 等人^[38]将稀疏表示与 PCA 子空间建模的优点相结合,提出一种新模式下的稀疏表示方法。Jia 等人^[39]将目标模板和候选样本进行分块,并采用模板块构成的字典稀疏表示候选样本块,从而有效利用目标本身的空间结构信息和局部信息来克服目标形变等难题。另外,基于稀疏表示模型的跟踪算法往往因为计算量过大而很难达到实时。为了提高其实时性,Liu 等人^[40]采用一种新的特征选择方法来提取具有更低维度、更具区分力的特征。Yan 和 Tong^[41]采用加权的 LASSO 算法来代替单位基向量,从而大大减少计算量。Mei 等人^[42]提出一种有界的粒子重采样算法,该算法能够在最小化求解之前有效剔除大量无效粒子,大大减少求解数量并提升实时性。

核密度估计是一种非参数估计方法,其无需利用数据分布的先验知识,能够根据数据本身的性质、特点来拟合分布。Comaniciu 等人^[43]将核密度估计与梯度信息相结合,成功提出一种基于核函数的目标跟踪算法。该算法通过对核密度进行估计,利用空间平滑相似函数对颜色直方图进行正则化,并为特征向量分配不同的权重。同时,算法通过迭代计算均值漂移向量来求解梯度优化问题,从而准确推断出目标的位置。在此算法的基础上,一些学者提出了改进的方法。Leichter 等人^[44]扩展了基于核密度估计的跟踪算法,将目标边界信息和颜色信息融入到跟踪过程中,并允许核进行仿射变换,从而使目标定位更加精确。另外,为了克服迭代求解过程中易收敛于局部极值的问题,Shen 等人^[45]提出一种多带宽的均值漂移算法来可靠、快速地获得目标的真实位置。

与生成式模型不同,判别式模型旨在利用样本训练一个鲁棒的分类器,通过分类器将图像分为背景与目标两类,从而实现目标的精确跟踪。常见的判别式模型有:支持向量机^[46-51]、集成学习^[52-56]和相关滤波^[57-70]等。

支持向量机是一种应用广泛的二分类模型。它在统计学习理论的基础上,根据有限的样本信息在模型学习能力与模型复杂性之间寻求最佳平衡,从而达到较好的分类效果。Shai Avidan^[46]首次将支持向量机引入到目标跟踪算法中,利用光流法以及离线训练过的支持向量机分类器进行跟踪。虽然在跟踪性能上表现良好,

但由于不能在线更新以及目标类别的局限性,在跟踪过程中会出现跟踪不稳定甚至是跟踪丢失的现象。Li 等人^[47]采用基于图模型的核方法来学习支持向量机分类器。与之前采用二值化样本标签的算法不同, Hare 等人^[48]在考虑样本空间分布的同时,成功训练了一个结构化的支持向量机,从而能够准确地预测目标的位置。Zhang 等人^[49]将多个具有不同学习率的支持向量机分类器进行组合,并利用信息熵对跟踪结果进行有效地结合。Bai 等人^[50]在总结先前算法的基础上,提出一种基于增广核矩阵学习框架的跟踪算法。该算法有效结合 LBP 和像素强度两种特征来增强表征能力,提升算法对目标形变等因素的适应能力。Sharma 等人^[51]考虑到部分图像中提取目标的不准确性,成功地将一种在线顺序学习算法结合到支持向量机的参数学习中,有效提升了算法在复杂场景中的鲁棒性。

集成学习旨在将多种方法进行组合,从而有效增强跟踪算法的性能。Helmut^[52]利用 AdaBoost 集成学习算法^[53]得到一个自适应强分类器。该算法在跟踪过程中维持一组选择器和一个弱分类器集合。当跟踪到新的目标时,算法会对集合中的所有弱分类器进行在线更新。与此同时,选择器会依次选取最优的弱分类器,并对样本权重进行更新。Avidan^[54]同样利用 AdaBoost 集成学习算法将众多弱分类器组合得到一个强分类器。与之前不同的是,该算法对所选区域中的像素进行分类并得到相应的置信图,然后利用均值漂移算法确定目标的位置。同时,该算法会在每一帧处理后对各个弱分类器进行更新。虽然,在线更新能够很好地应对外观的变化,但同时也会带来跟踪漂移的问题。为此,Grabner 等人^[55]结合初始帧中的先验知识来对在线更新进行优化。Babenko 等人^[56]将在线 AdaBoost 与多示例学习进行结合,将跟踪到的目标与其附近的一些样本纳入到一个集合中,并用该集合的标签来代替集合中所有样本的标签,从而在跟踪位置稍有偏差的情况下依然能够选择出正确的弱分类器,有效解决了在线更新中跟踪漂移的问题。

相关滤波旨在计算候选样本与相关滤波器模板之间的相关程度,是一个多元二次多项式回归模型。其巧妙地利用卷积定理,将候选区域与相关滤波器通过快速傅里叶变换映射到频域中进行点乘运算,再利用逆变换映射回空域便可得到所有候选样本的响应。2010 年, Bolme 等人^[57]首次将相关滤波应用到跟踪中,并提出著名的 MOSSE (Minimum Output Sum of Squared Error) 算法。该算法不仅

在运算速度上能够超过 700fps，而且其卓越的跟踪能力也得到了学者们的认可。因此，众多学者在此基础上继续研究并提出若干基于相关滤波的跟踪算法。Henriques 等人^[58]将循环矩阵与岭回归理论相集合，并在核空间中采用相关滤波器进行跟踪，成功提出十分经典的 CSK (Circulant Structure of Tracking-by-Detection with Kernels) 跟踪算法。并且，在后续的工作中 Henriques 等人^[59]借助 HoG 特征将 CSK 算法提升为 KCF 算法，取得了十分理想的跟踪效果。Danelljan^[60]等人将颜色属性与 CSK 算法相结合，并将高维特征映射到高斯核空间，不仅提升了滤波器的鲁棒性与区分能力，还有效减少了计算压力。同时，Danelljan^[61]等人在同年成功提出 DSST (Discriminative Scale Space Tracker) 跟踪算法。该算法借助 HoG 特征学习一种自适应多尺度相关滤波器，能够对目标的尺寸做出快速且有效的估计。一些学者在后续研究中发现，循环采样带来的边界效应时刻影响着相关滤波器的精度与鲁棒性。为此，Danelljan^[65]等人将空间上的正则化权重图结合到算法中，有效削弱了相关滤波器的边界效应，极大地提升了跟踪算法的性能。Liu 等人^[70]将分块思想与相关滤波做结合，将目标分成多块并分别对应一个相关滤波器。与此同时，算法将各个滤波器得到的响应图进行拼接，并在拼接后的响应图中利用贝叶斯定理确定目标的最终位置。

虽然，传统目标跟踪算法在过去的发展中取得了显著的成果，并提供了许多非常宝贵的理论知识。但是，其较低的鲁棒性与准确性都无法满足现阶段的要求。近些年，随着深度学习的不断发展，学者们已经逐渐将研究重心转移到基于深度学习的目标跟踪算法上，并进一步全面提升了跟踪算法的性能与应用价值。

1.2.2 基于深度学习的目标跟踪算法

传统的目标跟踪算法多是以浅层特征进行描述的。所谓浅层特征是指研究人员手工进行设计的表观特征，这些特征一般在特定的任务中跟踪效果较好，但是当应用在通用场景或较为复杂的场景时，跟踪效果就会出现明显下降的趋势。例如，RGB 颜色特征在亮度变化明显的场景就很容易受到干扰；颜色直方图特征一般用来应对目标形变所产生的外观变化，但是当背景具有相同颜色直方图时就会受到干扰；HoG 特征通过对图像不同方向的梯度进行统计来获取稳定且具有较高识别性的特征，但是当图像较为平滑的时候，HoG 特征就会出现失效的情

况。近年来,随着深度学习的不断发展,其高效的特征表征能力及强大的端到端学习机制得到了学者们的广泛关注。作为深度学习中最具代表性的一种网络模型,卷积神经网络(Convolutional Neural Network, CNN)被广泛应用在目标检测、目标跟踪、语义分割等计算机视觉领域,并取得了前所未有的突破。与传统算法相比, CNN 提取的特征会包含更加丰富的信息。从 CNN 较浅层中提取的特征保留了图像的细节信息,并具有传统算法中浅层特征的特性,在进行精确分割、精确定位和精确匹配等环节时具有很理想的效果。从 CNN 较深层中提取的特征包含了许多语义信息,这些信息不仅具有较高的识别性,还具有一定的平移不变性等优良特性,被广泛应用在目标识别、实例分割和图片标注等方面。同时,随着大数据时代的来临,训练数据也较比之前有了大幅度的增加。CNN 的训练过程采用端到端学习的模式,不仅能够在大数据的驱动下实现最优特征的自动选取,还极大减少了模型设计及训练时的工作量。接下来,本小节将以 CNN 为主要研究对象,介绍基于 CNN 的目标跟踪算法的研究现状。

与传统跟踪算法相同,基于 CNN 的目标跟踪算法也可以分为判别式算法^[71-85]和生成式算法^[86-100]两大类。其中,最常用的判别式算法是利用 CNN 模型去提取深度特征,并用该特征取代传统判别式模型中(如相关滤波等)的浅层特征。Ma 等人^[71]将 VGGNet^[116]中三种不同的卷积层(Conv3-4, Conv4-4, Conv5-4)提取的特征分别输入到三个对应的相关滤波器中,最后将滤波器输出的三个权重图进行结合,从而得到目标的精确位置。Danelljan^[72]等人利用 VGGNet 提取特征,并提出著名的 C-COT (Continuous Convolution Operators Tracker) 算法。该算法通过立方插值成功将不同分辨率的特征图映射到连续空域,从而运用 Hessian 矩阵求得具有亚像素精度的目标位置。在后续研究过程中, Danellja 等人^[73]还将 CNN 输出的深度特征与传统的 SRDCF 算法^[65]相结合,提出了 DeepSRDCF 跟踪算法。在该算法中,作者指出卷积层的浅层特征比深层特征在精确跟踪方面表现得更好,并且该浅层特征要完全优于传统算法中的浅层特征。之后, Danellja 等人^[74]在前作 C-COT 的基础上提出 ECO (Efficient Convolution Operators) 跟踪算法。该算法从模型大小(特征复杂度)、训练集大小和模型更新三个方面进行改进,在保证跟踪性能的同时大幅度提升了算法的效率。

与前述取代传统浅层特征的判别式算法不同，一些基于 CNN 的判别式算法直接利用端到端学习来训练 CNN，并将其作为一个高精度的分类器。Nam 等人^[78]将网络结构分成两部分，并提出著名的 MDNet (Multi-domain Network) 跟踪算法。该算法采用共享层部分来提取通用的特征，采用视频专用层部分对目标的专用特征进行提取，并且该部分会根据目标样本进行在线微调，从而实现准确、鲁棒的在线跟踪。之后，Nam 等人^[82]利用多个 CNN 对目标外观进行表示，并采用树形结构来对这些外观模型进行管理，从而获得一个鲁棒的分类器。Han 等人^[85]借鉴集成学习的方法，将卷积层和具有多分支的全连接层相结合，运用一种简单有效的正则化技术来减少模型多样化的限制，显著提高了算法的跟踪性能。

虽然，基于 CNN 的判别式跟踪算法在性能上已远远超过传统的跟踪算法，但其仍面临着一些问题。当采用 CNN 提取的深度特征取代传统的浅层特征时，CNN 并没有针对当前的跟踪任务进行适当调整，从而无法在跟踪时达到最优化，容易造成跟踪不稳定甚至是跟踪丢失。当采用端到端学习去训练 CNN 时，虽然得到的分类器能够在线自适应的进行调整，但模型的在线更新依然会带来几个问题。一方面，CNN 需要采用反向传播算法进行网络的在线更新，当网络较深时，反向传播算法的时耗会急剧增加，极大地影响了算法的实时性。另一方面，在线跟踪过程中采集的新样本会非常少，在这种情况下更新的模型很容易过拟合，从而使算法的准确性和鲁棒性明显下降。为了克服这些问题，基于 CNN 的生成式算法逐渐成为了研究的重点。

基于 CNN 的生成式算法采用 CNN 提取候选样本与目标模板的深度特征，通过学习一个精确、鲁棒的相似度量函数，实现深度特征之间的精确匹配。所以，这种算法也可以称为基于深度特征匹配的跟踪算法。最近几年，凭借在学习相似度量函数上的极大优势，孪生卷积神经网络已经被广泛应用在基于深度特征匹配的跟踪算法中，并以其均衡的准确性、鲁棒性和实时性获得了较多的关注。Tao 等人^[86]使用孪生卷积神经网络分别对模板模块和搜索模块进行特征提取。为了提高跟踪精度，算法减少了最大池化层的使用，并采用感兴趣区域池化层 (ROI Pooling) 来减少特征提取的复杂度。同时，为了提高算法的性能，采用光流法对需要搜索的候选样本进行滤除，保证采样机制的自适应性。Held 等人^[87]将孪

生卷积神经网络作为特征提取器,并运用全连接层融合张量。在跟踪过程中,算法采用一种回归的方式进行处理,将上一帧与当前帧的图像剪裁后输入到离线训练好的网络中,继而计算出当前帧目标的位置。Bertinetto 等人^[90]提出的 SiamFC (Fully-convolutional Siamese Networks) 跟踪算法尤为引人注目。其首次利用全卷积网络来提取模板和搜索区域的深度特征,并将输出的深度特征运用交叉相关计算一次性得到匹配后的响应图。该算法不仅在跟踪性能上表现优异,在运行速度上也完美达到了实时性的要求。之后,许多学者都以此算法为基础进行改进和创新,为基于深度特征匹配的生成式目标跟踪算法开辟了新的分支。Valmadre 等人^[91]在算法中引入相关滤波器,并在 SiamFC^[90]的相似性度量计算中引入一个可微分的层。Guo 等人^[92]在算法中引入了模板更新机制,通过学习背景及目标外观的变化对模板特征进行更新,从而提高算法的鲁棒性和准确性。Li 等人^[95]在算法中引入了目标检测算法中常用的 RPN (Region Proposal Network) 模型,不仅能够有效应对目标的多尺度变化,还极大提高了算法的精度和鲁棒性。Zhu 等人^[96]在 Li 等人的研究基础上引入干扰感知模块,并对训练用的样本集进行有效扩充,成功提升了算法在复杂场景下的鲁棒性。在之前的研究基础上, Li 等人^[97]成功将深层网络架构引入到算法中,不仅超越了之前的深度特征匹配算法,还打破了浅层网络架构的限制,开辟了新的研究趋势。

虽然,基于深度特征匹配的生成式跟踪算法已经有了极大的进步,并以其均衡的准确性、鲁棒性和实时性获得了较多的关注。但是,由于网络结构中缺少层级间的自适应特征融合、缺少再检测机制、自适应模板更新和特征融合策略不够完善等原因,其在复杂场景下的鲁棒性与准确性都会明显降低,并且较比性能突出的判别式算法仍然存在着一定的差距。所以,如何提升基于深度特征匹配的目标跟踪算法在复杂场景下的跟踪性能,依旧是一个值得研究且富有挑战的课题。

基于深度学习的目标跟踪算法在大数据及人工智能的浪潮下取得了令人瞩目的成就,但其仍然存在着一些局限性。第一,深度学习模型拥有着十分巨大的参数量,无论是离线还是在线训练都需要足够的样本数据支撑。然而,有效的样本数据较为缺乏,尤其是在跟踪过程中,这使得深度学习模型有时很难达到理想状态。第二,目标跟踪算法在大多数场景下需要达到实时性要求,而跟踪精度高、

鲁棒性强的算法需要对网络模型进行在线更新，这个过程往往非常耗时。第三，大多数优秀的深度学习模型仍处于算法仿真状态，如何结合现有的硬件资源高效地应用于工程实践中，仍需要进行不断地探索与验证。

1.3 复杂场景下目标跟踪面临的挑战

目标跟踪算法不仅需要精确计算出各时刻跟踪目标的状态信息，还需要分析目标在视频或图像序列中的运动轨迹。然而在复杂场景下进行跟踪，不仅目标的外观会因为光照、运动、遮挡等因素发生各种各样的变化，而且背景中的相似干扰更会提升跟踪的难度。如果跟踪算法不进行适当的更新或调整，算法的跟踪性能就会受到影响，准确性和鲁棒性就会急剧下降。下面，本节将对复杂场景下目标跟踪过程中所面临的挑战做简要阐述。

(1) 光照变化及尺度变化

光照变化及尺度变化是跟踪过程中经常面临的难题。光照变化是指在跟踪过程中，目标外观上的光谱分布或光强度产生明显变化。通常情况下，光照变化可以分为非均匀光照变化和均匀光照变化两种。非均匀光照变化是指目标外观的各部位光照不均匀，局部光照变化不一致的情况。均匀光照变化是指目标外观各部位光照变化一致的情况。其中，非均匀光照变化是跟踪过程中最常见的，如图 1.2 (a) 所示。尺度变化是指在跟踪过程中，目标在每帧图像中所占的像素比例发生变化，如图 1.2 (b) 所示。目标跟踪算法往往需要对目标的尺度变化进行估计，从而调整跟踪框大小并防止背景噪声的引入。



(a) 非均匀光照变化

(b) 尺度变化

图 1.2 光照变化及尺度变化

Figure 1.2 The illumination variation and the scale variation

(2) 目标形变

目标形变是指非刚性物体在跟踪过程中出现的形状变化。如图 1.3 所示，目

标在运动过程中出现了严重的形变，空间结构信息也发生了改变。通常情况下，跟踪算法经常采用直方图特征来应对目标形变。但是，当背景具有与目标相同颜色的直方图特征时，跟踪效果就会受到干扰。一些跟踪算法还采用局部分块处理的方法来应对目标形变。但是，如何协调并融合各块的跟踪结果也是一个非常棘手的问题。随着深度学习的发展，CNN 提取的深层特征具有丰富的语义信息，可以有效地应对目标形变带来的干扰。但是，仅仅依靠深层特征又无法完全实现目标的精确定位。所以，如何设计一个鲁棒、准确的跟踪模型来处理目标形变是一个具有挑战的研究方向。

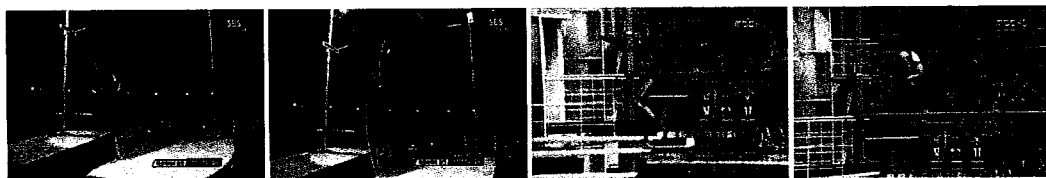


图 1.3 目标形变

Figure 1.3 The deformation of the target

(3) 目标旋转

目标旋转在跟踪过程中一般可分为平面外旋转和平面内旋转两种情况。如图 1.4 (a) 所示，平面内旋转是指目标在相对于摄像头的二维平面内发生旋转，一般通过多角度采样等策略来解决。然而，平面外旋转相对来说比较复杂。如图 1.4 (b) 所示，目标在发生平面外旋转前和平面外旋转后，目标的外观已经发生了巨大的变化。如果没有适当的更新策略或自适应算法，针对旋转前的目标表观模型或跟踪策略会无法适应旋转后的目标，从而造成跟踪失败。目前为止，学者们还没有提出能够检测出平面外旋转的算法，这无疑给跟踪算法的设计带来不小的困扰。



(a) 平面内旋转

(b) 平面外旋转

图 1.4 目标旋转

Figure 1.4 The rotation of the target

(4) 目标遮挡

目标遮挡是被学者们投入最多精力,也是目标跟踪领域内最为棘手的问题之一。目标遮挡在跟踪过程中可以分为部分遮挡和完全遮挡两种情况,如图 1.5 所示。目标遮挡之所以成为跟踪领域的难题,其主要原因有以下两点:第一,遮挡跟踪目标的并不是目标本身的一部分,而是背景中的干扰物。与其它跟踪难题不同,被遮挡目标的空间结构信息已被完全破坏,大量的背景信息被引入到了候选样本中。这些引入的背景信息会容易使候选样本被分类为背景,或者降低与目标模板的匹配程度。第二,被遮挡的目标会影响模型的更新。一些跟踪算法往往会设计相应的模块来检测目标是否发生遮挡,并根据遮挡程度进行更新。然而,如何保证设计的模块能够对遮挡做出正确的检测,如何有效排除更新时背景干扰的影响,依旧是跟踪领域一个极具挑战的难题。



图 1.5 目标遮挡

Figure 1.5 The occlusion of the target

(5) 相似背景干扰

相似背景干扰是指跟踪目标周围存在与目标表观相似的背景干扰,该干扰的纹理、边缘及其它特征信息与目标本身的特征十分相似,是跟踪领域另一种十分棘手的问题。如图 1.6 (a) 所示,跟踪的目标周围出现多个与目标纹理、亮暗程度相似的干扰目标。如图 1.6 (b) 所示,跟踪的目标周围出现多个与目标形状、大小相似的干扰目标。在设计跟踪模型时,只利用单一的信息去解决相似背景干扰的问题往往会遇到困难。如果训练过程中让模型专注于区分极其相似的目标,模型就会因为缺少泛化能力而陷入过拟合。如果在训练过程中专注于提升模型的泛化能力,那么模型就会在处理相似背景干扰时陷入误区。所以,在解决相似背景干扰的问题上,往往需要跟踪模型结合多种信息进行训练,如上下文信息、背景信息、前景信息和多层特征信息等。如何利用其它有效信息提升跟踪模型处理

相似背景干扰问题的能力，仍然是一个极具研究意义的课题。



(a) 纹理与亮暗程度相似

(b) 形状与大小相似

图 1.6 相似背景干扰

Figure 1.6 The similar background interference of the target

(6) 快速运动及运动模糊

快速运动是指在跟踪过程中目标在连续几帧之间的位移较大。运动模糊是指跟踪目标在相对摄像机运动时，相机内的单个像元对目标的多个区域进行积分而造成的模糊，如图 1.7 所示。在跟踪时，运动模糊会造成目标的外观特征偏离真实的特征，从而影响跟踪模型的跟踪性能。一些算法在每帧定位到目标后，会在目标周围开设一定范围的搜索区域，从而减少背景信息的引入并提高算法的实时性。然而，快速运动会使目标脱离搜索区域的范围，导致候选样本中充满背景信息。这样不仅容易造成跟踪失败，还会为模型的更新带来一定的风险。所以，如何改进跟踪策略使算法能鲁棒、准确地跟踪快速运动及运动模糊的目标，仍然是一个极具研究意义的研究方向。



图 1.7 快速运动及运动模糊

Figure 1.7 The fast motion and the motion blur of the target

1.4 论文主要内容及结构安排

目标跟踪算法不仅需要对任意指定的目标进行精确定位，还需要应对跟踪过程中各种复杂的场景。传统的跟踪算法虽然在过去的发展中取得了显著的成果，但是，其较低的鲁棒性与准确性都无法满足现阶段对跟踪算法的要求。基于深度学习的判别式算法虽然极大地提升了跟踪算法的精度与鲁棒性，但是，其较复杂

的网络模型需要采用反向传播算法进行网络的在线更新。当网络较深时，反向传播算法的时耗会急剧增加，极大地影响了算法的实时性。并且，在线跟踪过程中采集的新样本会非常少，在这种情况下更新的模型很容易过拟合，从而使算法的准确性和鲁棒性明显下降。因此，本论文在研究大量国内外目标跟踪算法的基础上，对基于深度特征匹配的生成式目标跟踪算法进行研究，并针对复杂场景下基于深度特征匹配的目标跟踪算法的局限性进行改进。本论文主要开展可见光单目标跟踪算法的研究，主要研究内容如下：

(1) 针对基于深度特征匹配的目标跟踪算法在复杂场景下准确性及鲁棒性降低且模板更新策略不够完善的问题，提出一种结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。在跟踪过程中，当背景中出现相似干扰时，基于深度特征匹配的目标跟踪算法生成的响应图中易出现多波峰的状况，从而造成跟踪框不稳定甚至跟踪目标丢失的情况。为此，本算法利用一种高效且准确的再检测孪生卷积神经网络作为再检测机制来应对响应图中的多波峰。同时，算法中还提出一种基于生成式模型的自适应模板更新方法来应对跟踪过程中遮挡、形变等复杂因素带来的影响。

(2) 针对基于深度特征匹配的目标跟踪算法网络结构较浅且缺少层级间的自适应特征融合的问题，提出一种结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。虽然，基于深度特征匹配的目标跟踪算法利用浅层卷积神经网络和离线训练能够在一些低复杂度的场景取得不错的跟踪效果。但是，由于缺少丰富的语义特征，其在应对复杂场景下一些具有挑战性的难题时，跟踪效果将会明显下降。本算法利用一种改进的残差网络来替代传统的浅层网络，从而提取出具有丰富语义信息的深层特征。与此同时，算法中提出一种高效的自适应多层特征融合策略，有效提升了响应图的质量，减少了在复杂场景下相似背景干扰所带来的影响。另外，该算法利用一种自适应特征信息融合方法来实现模板的更新，从而有效克服遮挡、目标形变等难题。

(3) 针对基于深度特征匹配的目标跟踪算法前景信息利用不充分，跟踪过程中在语义背景下易丢失目标的问题，提出一种前景信息引导的孪生卷积神经网络目标跟踪算法。考虑到现有的基于深度特征匹配的目标跟踪算法在离线训练时

正样本对包含的挑战因素太少,在目标遇到遮挡或较严重形变时跟踪效果下降的问题。本算法采用一种相对简单的采样策略来扩充离线训练时正样本对中的挑战因素,从而有效增强算法的鲁棒性。并且,为了进一步提升在语义背景下算法的识别能力,算法将背景信息进行填充遮挡,从而增强前景信息的显著性。与此同时,将处理后的数据输入到基于卷积神经网络的引导分支中,并运用一种填充损失计算方式来改进算法的损失函数,有效提升算法在语义背景下的识别能力。此外,本算法采用一种改进的特征信息融合方法来实现模板的更新,进一步应对目标的外观变化。

本论文详细的章节结构安排如下所示:

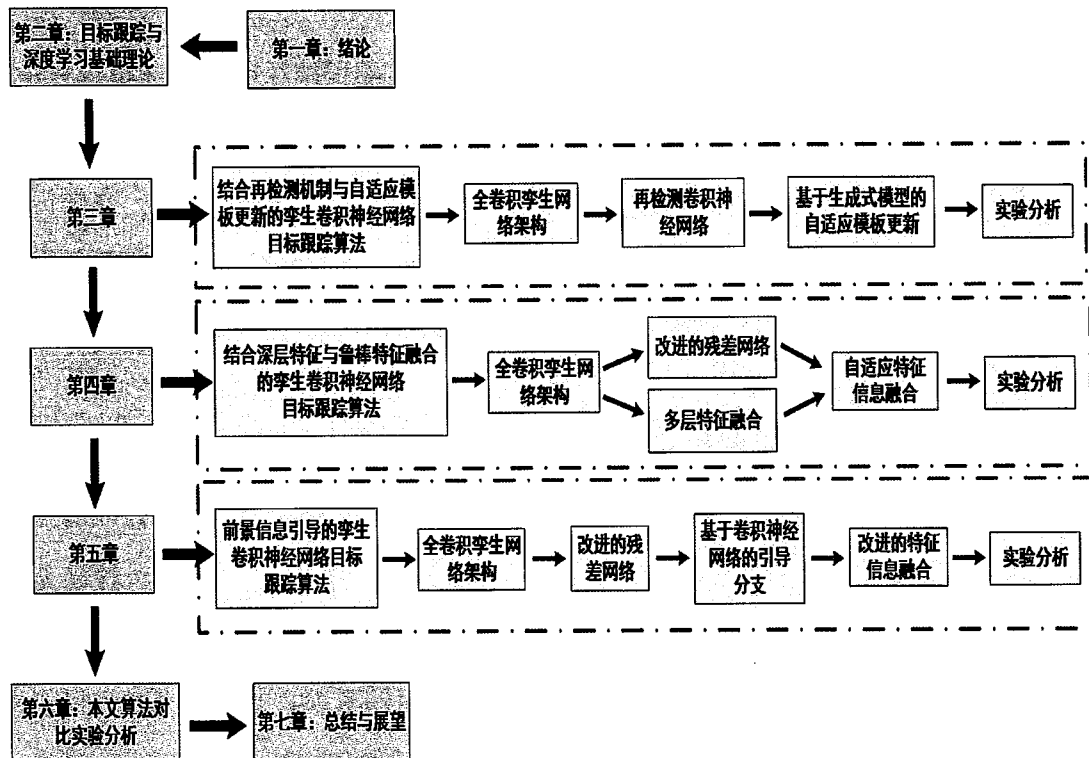


图 1.8 论文章节结构安排

Figure 1.8 The framework of this dissertation

第一章, 绪论。主要阐述了本课题的研究背景及研究意义, 分别从传统目标跟踪算法和基于深度学习的目标跟踪算法两大类, 进行可见光单目标跟踪算法研究现状的总结。接着, 对目标跟踪领域所面临的挑战进行了简要分析, 并对一些常见且较难解决的问题进行了举例。最后, 列出本论文主要的研究内容及详细的章节结构安排。

第二章，目标跟踪与深度学习基础理论。首先，对单目标跟踪算法的基本框架和关键部分进行简要介绍。接着，对深度学习基本原理以及几种典型的卷积神经网络模型进行简述。然后，对基于深度特征匹配的目标跟踪算法中常见的孪生网络架构进行简要描述，并对经典的 SiamFC 算法原理进行概述。最后，对本论文中用于实验分析的测试数据集及评价指标进行介绍。

第三章，结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。首先，提出一种高效且准确的再检测孪生卷积神经网络作为再检测机制来应对响应图中的多波峰。接着，提出一种基于生成式模型的自适应模板更新方法来应对跟踪过程中遮挡、形变等复杂因素带来的影响。最后，通过实验分析对跟踪算法的跟踪性能进行验证。

第四章，结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。首先，提出一种改进的残差网络来替代传统的浅层网络，从而提取出具有丰富语义信息的深层特征。接着，提出一种高效的自适应多层特征融合策略，有效提升了响应图的质量，减少了在复杂场景下相似背景干扰所带来的影响。然后，提出一种自适应特征信息融合方法来实现模板的更新，从而有效克服遮挡、目标形变等难题。最后，通过实验分析对跟踪算法的跟踪性能进行验证。

第五章，前景信息引导的孪生卷积神经网络目标跟踪算法。首先，提出一种相对简单的采样策略来扩充离线训练时正样本对中的挑战因素，从而有效增强算法的鲁棒性。接着，引入一种基于卷积神经网络的引导分支，并运用一种填充损失计算方式来改进算法的损失函数，有效提升算法在语义背景下的识别能力。然后，提出一种改进的特征信息融合方法来实现模板的更新，进一步应对目标的外观变化。最后，通过实验分析对跟踪算法的跟踪性能进行验证。

第六章，本文算法对比实验分析。利用标准数据集对本文提出的三种算法进行对比实验，从而能够对三种算法的准确性、鲁棒性及有效性进行全面分析。利用实际应用场景下采集的视频数据进行实验分析，有效验证本文提出的三种算法在实际应用场景中的跟踪性能，进一步明确基于深度特征匹配的目标跟踪算法适用的场景与对象。

第七章，总结与展望。针对本论文的主要创新成果进行总结，对研究工作中的不足进行归纳，并对之后研究工作的重点进行展望。

第2章 目标跟踪与深度学习基础理论

2.1 目标跟踪概述

作为计算机视觉领域非常重要的组成部分和关键技术之一，目标跟踪在近些年成为了一个备受关注的研究课题。其核心旨在给定目标在视频或图像序列第一帧中的初始化状态，在后续过程中逐帧处理并准确预测出给定目标的边界框及运动轨迹等重要信息。虽然，利用目标检测算法也能够计算出给定目标的位置。但是，目标检测算法往往没有将视频或图像序列的时序信息进行利用，其只是单纯的检测出目标在一张图片或一个视频中的位置，并没有计算出与目标相对应的运动轨迹。另外，跟踪处理的目标往往具有任意性，仅能够利用视频或图像序列第一帧中的初始化状态，训练样本和有效数据极少。而目标检测可以利用大量的样本和数据来训练检测器，从而实现准确、鲁棒的检测。所以，为了实现复杂场景下的稳定跟踪，目标跟踪算法往往需要相对完善的算法框架。由于本论文主要开展可见光单目标跟踪算法的研究，所以接下来将主要对单目标跟踪算法的基本框架和关键部分进行简要介绍。

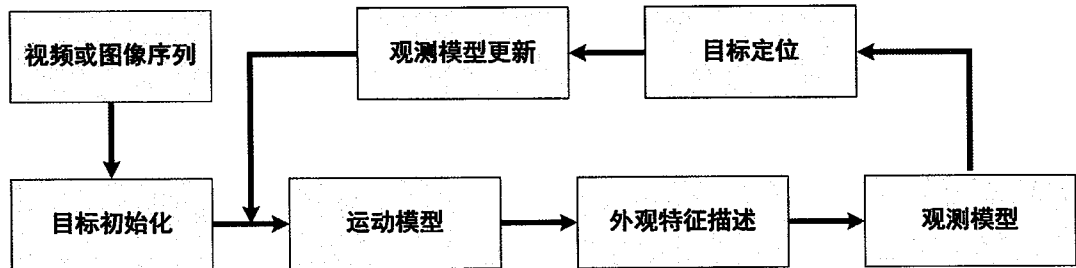


图 2.1 单目标跟踪算法的基本框架

Figure 2.1 The basic framework of the single object tracking

单目标跟踪算法的基本框架如图 2.1 所示。从图中可知，单目标跟踪算法主要由运动模型、外观特征描述、观测模型和观测模型更新四个关键部分组成。首先，利用视频或图像序列第一帧中的目标信息来实现目标初始化。接着，目标跟踪算法在运动模型预测出的搜索区域内生成候选样本，并对候选样本进行外观特征描述，从而提取出算法需要的外观特征。然后，利用观测模型将背景及目标的

特征进行区分,实现目标的精确定位。最后,利用定位后的目标信息对观测模型进行更新,以实现目标外观变化的不断适应。

(1) 运动模型

目标运动模型旨在利用上一帧中的目标信息预测出目标在下一帧中可能的位置,同时将该位置附近区域设定为搜索区域并生成相应的候选样本。在跟踪算法中,粒子滤波和滑动窗是应用较为广泛的两种运动模型。

粒子滤波^[101]是一种以蒙特卡罗方法为基础的模型,其主要被用来解决非高斯和非线性问题。粒子滤波模型的核心思想是利用一组具有关联权重的随机状态粒子去近似表示目标的后验概率分布。近些年,许多基于稀疏表示模型和基于子空间建模的目标跟踪算法都是以粒子滤波作为运动模型的。这些算法首先利用高斯分布对前一帧跟踪结果的状态向量进行采样,从而得到一组候选样本。然后,将这些样本输入到观测模型中进行计算,并得到每个样本的后验概率。最后,对各个样本的后验概率进行筛选,将样本集中具有最大后验概率的样本作为最终的跟踪结果。与其它运动模型相比,粒子滤波能够有效地减少样本的数量,不仅能保证样本的数量保持不变,还能保证其不会因为目标的尺寸发生变化而改变。同时,粒子滤波的状态变量可以自行设定。一般情况下,粒子滤波的状态变量包含了旋转、位移和尺度等信息,不需要再建立其它的模型对旋转变化及尺度变化等进行估计。然而,增加状态变量无疑会使搜索空间正比例变大,需要增加模型中的粒子数目来改善。所以,基于粒子滤波的算法往往计算量较大,实时性较比其它模型也逊色一些。

滑动窗运动模型^[102-105]通常会假设目标进行服从均匀分布的运动,在搜索区域内通过滑动窗口来产生所有需要的候选样本。与粒子滤波相比,滑动窗只能对目标的位置进行估计,并不能对目标的旋转、尺度等状态变量进行估计。并且,滑动窗模型所产生的样本数量会随着搜索区域的增大而指数性增加,从而在计算量上带来很大的负担。但是,滑动窗往往在应对快速运动的目标时会具有较好的跟踪效果。目前,基于相关滤波的跟踪算法往往会采用滑动窗运动模型。其通过将搜索框进行循环滑动来采集全部的候选样本,同时巧妙地利用卷积定理,将候选区域与相关滤波器通过快速傅里叶变换映射到频域中进行点乘运算,再利用逆

变换映射回空域便可得到所有候选样本的响应。整个计算过程不仅非常高效，还具有较强的准确性与鲁棒性。

虽然，运动模型在跟踪算法中具有重要的作用。但是，近些年大部分优秀的跟踪算法多是以 *tracking-by-detection* 方法为主导的。这些算法不再将运动模型作为重点的研究内容，而是将主要精力投入在观测模型的建立与更新上，从而获得更加高效、精确、鲁棒的跟踪框架。

(2) 外观特征描述

外观特征在目标跟踪算法中具有十分重要的作用，如何对外观特征进行描述会明显影响跟踪算法的性能。好的外观特征描述方式具有较强的鲁棒性，能够在外观发生不同的变化时都保持较强的统计特性，如梯度特征会使算法避免光照变化的影响，直方图特征会使算法避免目标形变的影响。另外，良好的外观特征描述方式还会具有较好的区分性，即描述出来的背景特征与目标特征在统计层面拥有非常明显的区别。在跟踪算法中，根据外观特征描述方式大致可将特征分为颜色特征、纹理特征、梯度特征和深度特征等类别。

颜色特征是一种应用较为广泛的外观特征，其对图像本身尺寸、视角、方向的依赖性较小，较比其它外观特征具有更高的鲁棒性。颜色直方图^[106]是一种简单有效的颜色特征，不仅能够反映出外观颜色的组成分布，还对图像的物理变换具有较低的敏感性。但是，颜色直方图是针对颜色分布所统计的结果，一定程度上丢失了像素间的位置特性。因此，当背景具有与目标相似的颜色分布时，颜色直方图的识别效果就会明显下降。作为另一种有效且易实现的颜色特征，颜色矩^[107]旨在利用颜色的一阶矩、二阶矩和三阶矩对图像的颜色分布进行表示。与颜色直方图相比，颜色矩不需要对特征进行向量化，只需要 9 个分量就可以进行描述。但是，由于其维度较低，往往需要与其它外观特征结合使用以避免低次矩所带来的影响。

纹理特征是一种能够对外观强度变化进行度量的常用特征，其能很好地对外观的规则性和平滑性进行量化。灰度共生矩阵^[108]是一种常用的纹理特征，其通过计算灰度图像的共生矩阵对指定距离和方向上的强度共存信息进行描述。虽然，灰度共生矩阵提取的纹理特征具有很好地鉴别力，但该方法的计算量较大并缺少

对全局信息的利用，从而在一定程度上限制了该方法的发展与应用。局部二值模式^[109]是一种能够快速提取纹理特征的方法，其不仅能够很好地应对跟踪过程中的光照变化，还具有较为简单的计算过程，通常被简称为 LBP (Local Binary Patterns) 特征。二位二值模式^[110] (2bit Binary Patterns, 2bitBP) 特征是以 LBP 特征为基础进行改进的特征。该特征能够通过计算积分图像获取，并与 haar-like 特征^[111]较为相似，较比 LBP 特征更加简洁有效且不容易发生过拟合。

梯度特征是一种能够反映图像轮廓、边缘和纹理等重要信息的特征，其被广泛应用于检测、识别、跟踪等领域。SIFT (Scale Invariant Feature Transform) ^[112]特征是一种具有光照不变性和尺度不变性的较为稳定的局部梯度特征。其利用高斯微分函数来获取对于尺度和光照不发生变化的关键点，同时基于局部的梯度方向为各个关键点分配方向值，并将关键点邻域内的梯度统计结果作为特征描述子。这样不仅能够使 SIFT 特征具有良好的不变性，还能增强其独特性并产生丰富的信息量。SURF (Speed-Up Robust Features) ^[113]特征是在 SIFT 特征的基础上进行改进的。相比于 SIFT 特征，SURF 特征利用 Hessian 矩阵的行列式来获取关键点，并根据关键点邻域的 Haar 小波变换响应来确定特征描述子。这样使其具有更快的速度和更简单的计算过程，更适合进行实时的梯度特征提取。HOG (Histogram of Oriented Gradient) ^[61]特征是近些年模式识别和计算机视觉领域较为常用的一种梯度特征。其利用局部区域的梯度方向直方图来描述特征，不仅能在图像发生光照变化和几何形变时具有较高的鲁棒性，还能保证在外观发生微小改变时不影响特征的识别效果。因此，HOG 特征经常与支持向量机、相关滤波器等模型结合，从而提升跟踪算法的跟踪性能。

上述外观特征都是手工制作的浅层特征，这些特征一般在特定的任务中跟踪效果较好，但是当应用在通用场景或较为复杂的场景时，跟踪效果就会出现明显下降的趋势。近些年，随着深度学习的发展，深度特征已经逐渐成为跟踪算法应对复杂场景的关键点。深度特征是一种具有结构性的由简单到复杂的特征。与人类大脑的视觉皮层结构相似，深度特征通常采用深度网络模型进行提取。如图 2.2 所示，在网络浅层通常会获取边缘、颜色、纹理等简单的外观特征，随着网络层数的加深，富含更多语义信息的高层次抽象特征会不断被提取。这些特征不

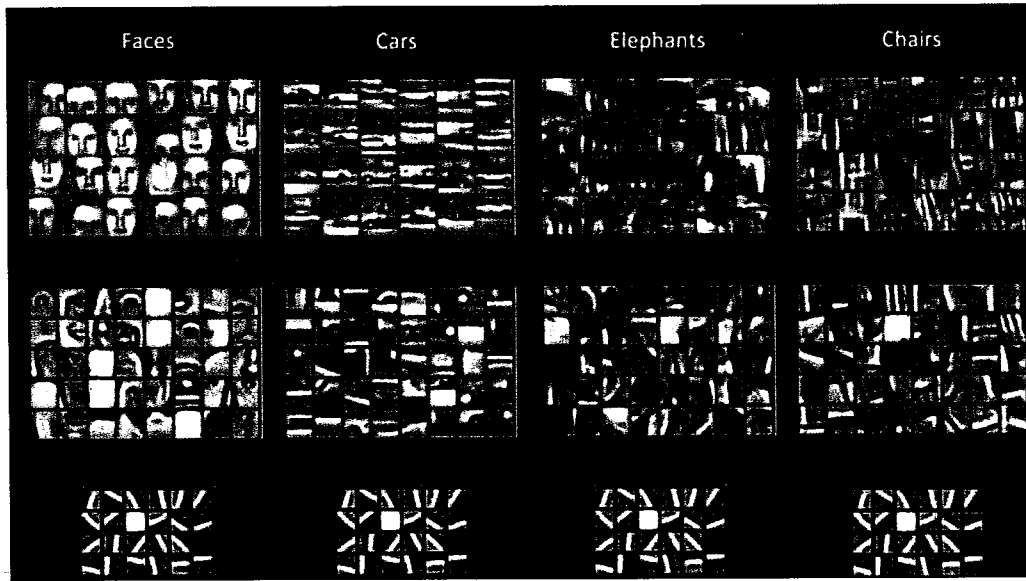


图 2.2 深度特征提取示意图

Figure 2.2 The diagram of the deep feature extraction

仅具有较高的识别性，还具有一定的平移不变性等优良特性，从而极大地提高了跟踪算法的鲁棒性与准确性。常见的深度特征有 CNN 特征、RNN（Recurrent Neural Network）特征等。其中，CNN 特征的应用最为广泛并最具代表性，是本文主要采用的外观特征描述方式。

（3）观测模型

在获取外观特征之后，观测模型会利用这些特征进行建模、分析和判别等处理，从而计算出在候选区域中目标出现的可能性。通常情况下，目标在跟踪过程中会出现两种情况的变化，即外在变化与内在变化。外在变化是指由于外在干扰所引起的目标外观变化，如视场角度变化、光照变化、遮挡、相机运动等。内在变化是指由于目标本身扰动所引起的变化，如目标形变、目标姿态变化、目标旋转等。这些变化都时刻需要观测模型利用鲁棒的特征进行建模和决策，从而实现候选区域中目标与背景的准确区分。在跟踪算法中，无论是传统的目标跟踪算法还是基于深度学习的目标跟踪算法，观测模型都可以分为生成式模型与判别式模型两大类。其中，生成式模型旨在学习目标的外观变化，通过计算目标模板与候选样本之间的相似度将目标从背景中识别出来。判别式模型旨在利用样本训练一个鲁棒的分类器，通过分类器将图像分为背景与目标两类，从而实现目标的精确跟踪。本论文已在章节 1.2 中从传统目标跟踪算法和基于深度学习的目标跟踪算

法两大类出发，对上述两种观测模型的相关研究进行了总结，所以这里将不再赘述。

(4) 观测模型更新

通常情况下，目标的外观会在跟踪过程中不断地发生变化。因此，观测模型在进行建模与决策的同时还需要不断地更新，从而对目标的外观变化不断适应，保证目标与背景的精确区分。鲁棒的观测模型更新策略是优秀的目标跟踪算法至关重要的环节，其需要在跟踪过程中对模型的稳定性与适应性进行良好的权衡。稳定性是指观测模型区分或识别目标与背景的能力稳定。保证模型稳定性的最佳方式其实是不对模型进行更新，但当目标外观发生严重变化时，不更新的观测模型会无法进行准确的检测，从而造成跟踪失败。适应性是指观测模型对外观变化的不断适应。增强适应性最佳的方式是在跟踪过程中将每一帧跟踪到的目标作为样本去更新观测模型。但当目标定位不准确时，目标样本往往会引入背景信息，从而在观测模型更新的过程中引入误差，导致模型的检测能力下降。如果这种错误长期累积并得不到纠正，跟踪算法将会出现跟踪漂移甚至是跟踪失败的情况。所以，无论是采用模板更新策略来实现生成式模型的更新，还是采用更新分类器判别超曲面的方式来实现判别式模型的更新，都需要在稳定性与适应性之间寻找一个较优的平衡点。

2.2 深度学习基本原理

2.2.1 深度学习基本结构

作为机器学习领域的新兴方向，深度学习（Deep Learning）已经被广泛应用于数据挖掘、自然语言处理、计算机视觉及多媒体学习等众多领域，极大地促进了人工智能相关技术的发展与人类社会的进步。深度学习在原有人工神经网络的结构上进行扩展，通过训练不断加深的多层神经网络结构去提取具有丰富语义信息的特征，从而弥补传统机器学习算法中人工设计的浅层特征所带来的缺陷。下面，本小节将主要对多层神经网络的基本结构进行介绍。

多层神经网络是对浅层神经网络的一种延续。其主要是在浅层神经网络的输出层后面继续添加感知层，使浅层神经网络原有的输出层变为中间的隐藏层，并

形成如图 2.3 所示的结构。

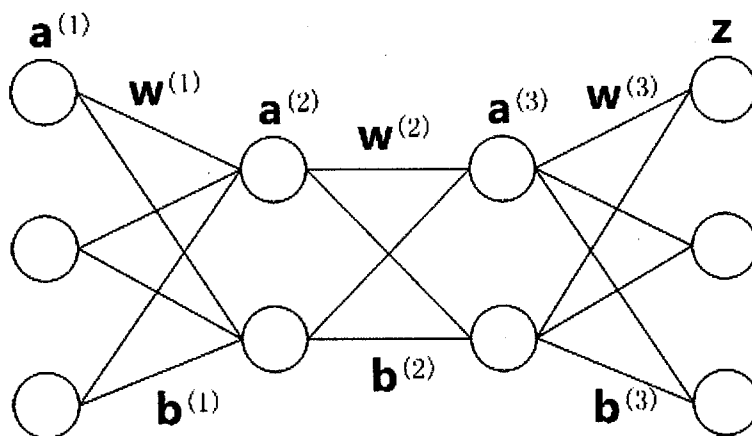


图 2.3 多层神经网络基本结构

Figure 2.3 The basic structure of the multilayer neural network

其中，所有的变量都代表矩阵变量。 $\mathbf{a}^{(1)}$ 为输入层， $\mathbf{a}^{(2)}$ 和 $\mathbf{a}^{(3)}$ 为隐藏层， $\mathbf{W}^{(1)}$ 、 $\mathbf{W}^{(2)}$ 和 $\mathbf{W}^{(3)}$ 为连接各层的权值参数， $\mathbf{b}^{(1)}$ 、 $\mathbf{b}^{(2)}$ 和 $\mathbf{b}^{(3)}$ 为各层的偏置参数， $\mathbf{z}$ 为输出层。各层输出可由下列推导公式计算：

$$g(\mathbf{W}^{(1)} * \mathbf{a}^{(1)} + \mathbf{b}^{(1)}) = \mathbf{a}^{(2)} \quad (2.1)$$

$$g(\mathbf{W}^{(2)} * \mathbf{a}^{(2)} + \mathbf{b}^{(2)}) = \mathbf{a}^{(3)} \quad (2.2)$$

$$g(\mathbf{W}^{(3)} * \mathbf{a}^{(3)} + \mathbf{b}^{(3)}) = \mathbf{z} \quad (2.3)$$

式中， $g(\cdot)$ 代表非线性激活函数。从公式中可以看出，多层神经网络的输出是按照逐层的方式进行计算的。从输入层开始，只有在当前层所有感知单元的值都计算完毕后，下一层的计算才会开始。这种不断推进的计算过程称为前向传播，这种多层神经网络的基本结构称为前馈神经网络。

设计多层神经网络时，在权重参数相同的情况下一般会选择层数较深的网络。如图 2.4 所示，该网络的权重参数数量为 33，但网络层数相对较浅。如图 2.5 所示，该网络权重参数数量依旧为 33，但网络层数相对较深。在较深的网络结构中，随着网络层数的增加，每一层感知单元会学习到比前一层更加抽象的表示。例如，第一个隐藏层能够提取外观的边缘特征，第二个隐藏层能够提取由边缘组成的形状特征，第三个隐藏层能够提取到由形状组成的图案特征，最后的隐藏层能够提取由图案所组成的全局外观特征。这种深入的特征表示模式能够使模型具

有更强的函数拟合能力，从而极大地提高算法检测与分类的性能。

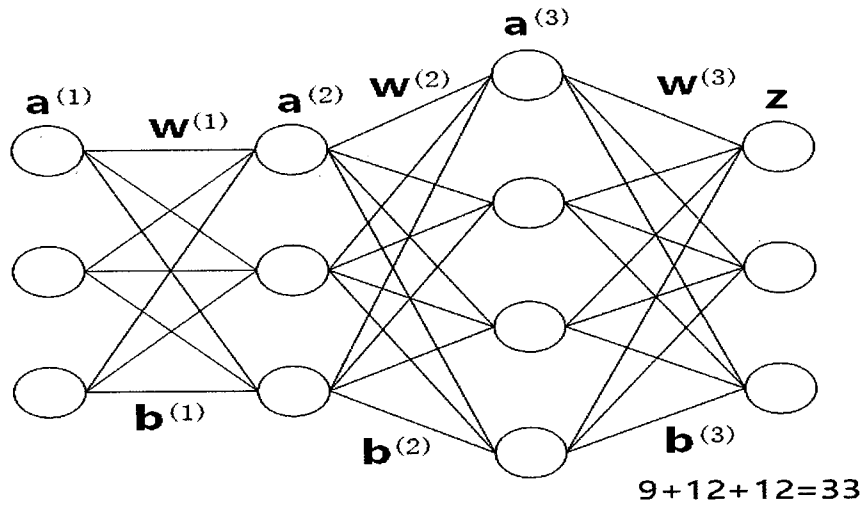


图 2.4 多层神经网络浅层结构

Figure 2.4 The shallow structure of the multilayer neural network

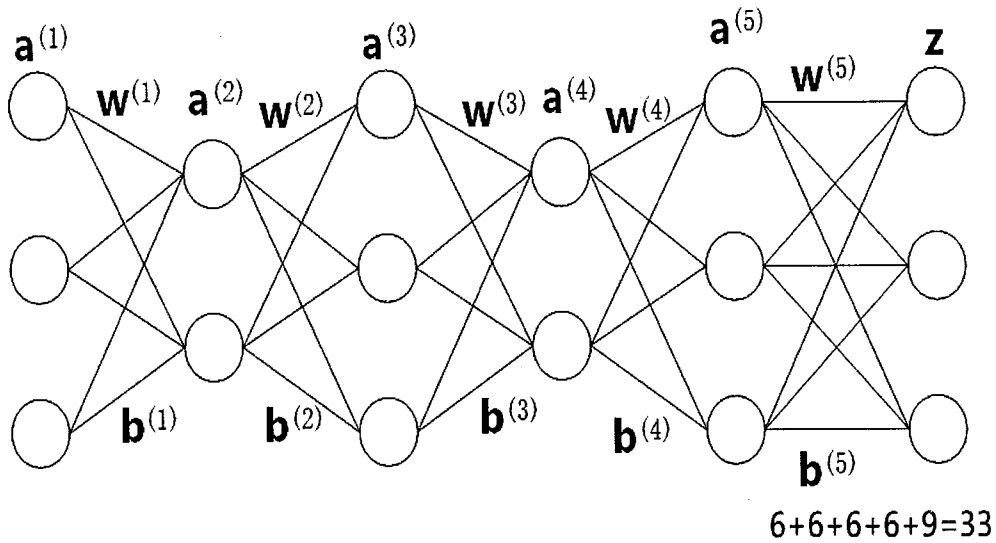


图 2.5 多层神经网络深层结构

Figure 2.5 The deep structure of the multilayer neural network

无论是多层神经网络的浅层结构还是深层结构，都需要对其中的权重参数和偏置参数等进行求解，从而实现输入到输出的映射。与传统机器学习算法不同，深度学习的多层神经网络结构会采用端到端学习（End-to-end Learning）的方式去训练网络。如图 2.6 所示，所谓端到端学习，就是将训练数据从输入端输入后会在输出端获得预测结果。将预测结果与真实标签数据利用损失函数计算后会得到

模型的误差，该误差会在网络中进行反向传播，并通过优化器对网络中的每一层参数进行更新，直到网络模型最终收敛。端到端学习的方式避免了多模型任务中复杂的训练过程，仅采用一种网络模型和一次整体的数据标注就可以获得理想的效果。

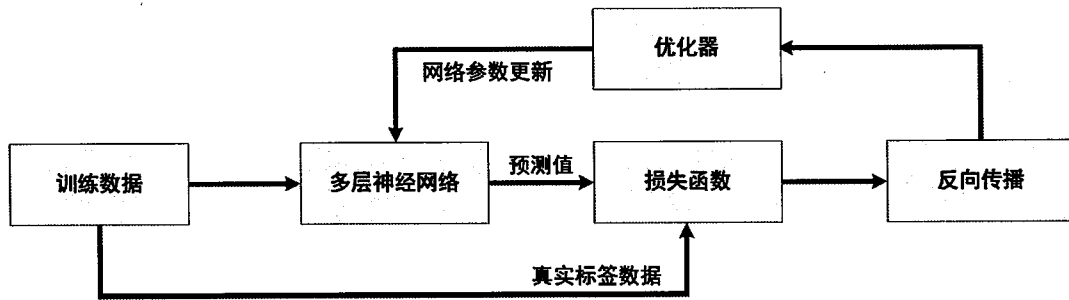


图 2.6 端到端学习流程图

Figure 2.6 The flow chart of the end-to-end learning

作为端到端学习中重要的一个环节，损失函数常被用来计算网络模型与真实标签数据之间的误差大小。合理选取损失函数不仅会有助于网络的训练过程，还会对网络模型最后的性能产生重要的影响。一般情况下，用于分类的网络模型会采用交叉熵作为模型的损失函数，用于回归的网络模型会采用均方误差作为模型的损失函数。虽然，损失函数变小能够在一定程度上表明网络模型预测的更加准确。但是，当在训练过程中发生过拟合时，损失函数也会越来越小，而网络模型的鲁棒性与准确性会明显下降。因此，在训练过程中一般会利用验证集来避免过拟合等问题所带来的影响。验证集与训练集的数据是不重叠的，训练过程经过一定周期后可以通过计算验证集损失函数的大小来确定网络模型的收敛程度，同时做出适当调整。验证集与训练集在网络训练过程中的准确度如下图所示。

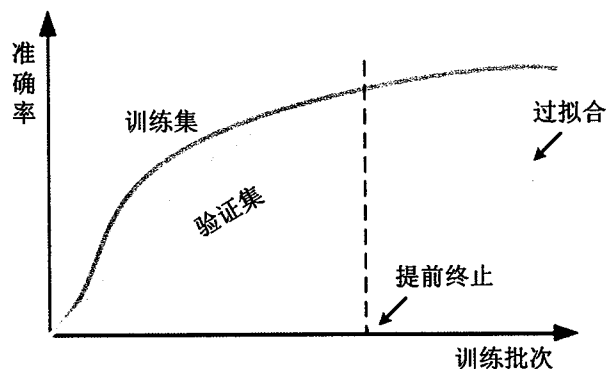


图 2.7 训练过程中验证集与训练集的准确度曲线

Figure 2.7 The accuracy curve of validation sets and training sets in tracking process

作为端到端学习中另一个重要环节，优化器能够利用优化算法对网络模型的参数进行更新。梯度下降法(Gradient Descent)是常用的最基本的优化算法之一，其公式如下所示：

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \eta \sum_{i=1}^N \frac{\partial E_i}{\partial \mathbf{W}} \quad (2.4)$$

式中， η 代表学习率， E_i 是第 i 个训练样本的损失函数， N 代表总的样本数。然而，如果按照公式对多层神经网络中的每个参数进行梯度计算，巨大的计算量会极大地降低算法的性能。所以，优化器采用链式求导法则将误差进行反向传递，

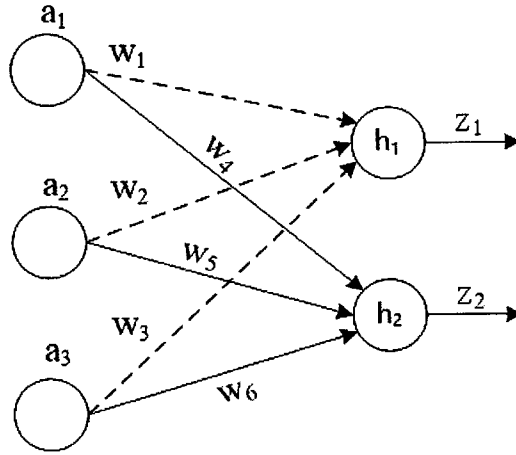


图 2.8 简单结构的神经网络

Figure 2.8 The simple structure of the neural network

逐层计算每个权重参数对模型误差的贡献。以图 2.8 为例，获取 W_1 参数的梯度可以采用下列公式进行计算：

$$z_1 = g(h_1) \quad (2.5)$$

$$E = (z_1 - \tilde{z}_1)^2 + (z_2 - \tilde{z}_2)^2 \quad (2.6)$$

$$\nabla W_1 = \frac{\partial E}{\partial W_1} = \frac{\partial E}{\partial z_1} * \frac{\partial z_1}{\partial h_1} * \frac{\partial h_1}{\partial W_1} \quad (2.7)$$

多层神经网络结构是深度学习中最基本的结构，是后续众多网络模型的基础。借鉴该结构下的端到端学习模式，深度信念网络(Deep Belief Network)、自编码器网络(Auto-encoder Network)、卷积神经网络(CNN)、递归神经网络(RNN)

等网络模型已得到较快的发展和广泛的关注。其中,自编码器网络与深度信任网络常被应用于无监督学习的领域,RNN 常被应用于处理序列化数据,CNN 常被应用于监督学习的领域。由于本论文采用 CNN 作为主要的网络模型,所以接下来将对 CNN 的基本原理进行介绍。

2.2.2 卷积神经网络基本原理

作为深度学习中最重要且应用广泛的网络模型之一,卷积神经网络(CNN)常被应用于处理视频或图像序列等非结构化的数据。其不仅继承了深度学习基本结构中的端到端学习模式,还在此基础上进行了扩展,从而一定程度上克服了模型复杂、参数过多的缺点,使其更适合于应用在计算机视觉等领域。

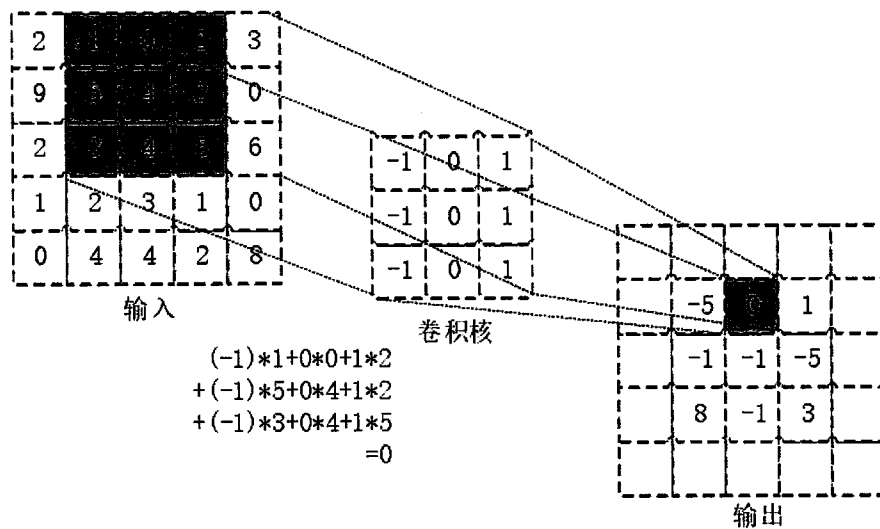


图 2.9 图像的卷积操作

Figure 2.9 The convolution operation of the image

在图像处理算法中,图像数据常被作为一种二维离散信号进行处理。如图 2.9 所示,图像的卷积操作是将卷积核作为一种滤波器模板,并将该模板翻转 180° 后在图像上进行滑窗操作,即翻转后的卷积核与输入图像对应位置上的像素逐一相乘并累加求和,从而得到滤波后的输出图像。CNN 依靠各式各样的卷积核对图像的特征进行提取,卷积核内部的操作系数根据不同的任务通过端到端学习自动确定。网络中卷积核在输入图像上覆盖的区域叫作感受野,其会随着网络层数的堆叠不断变大,卷积核提取的特征也会随着层数的堆叠逐渐趋于全局性,

如图 2.10 所示^[71]。

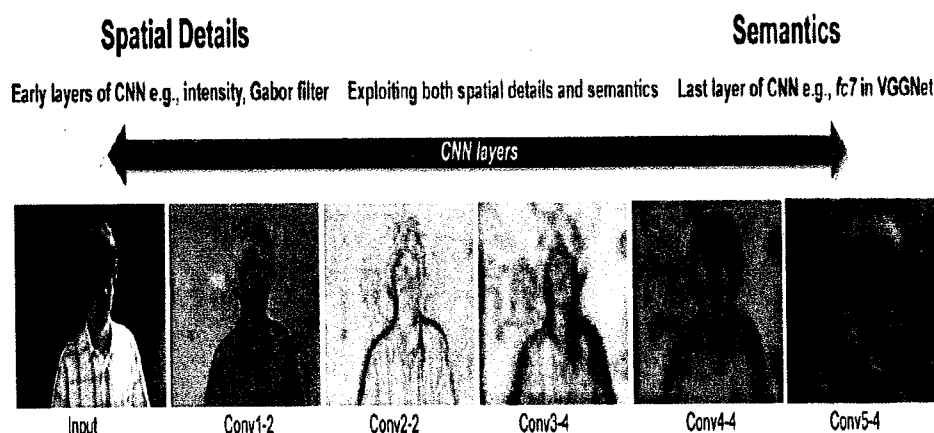


图 2.10 卷积神经网络的结构与特征提取

Figure 2.10 The structure and the feature extraction of CNN

在多层神经网络中，每个神经感知单元会接收上一层中所有感知单元的信号，从而形成紧密连接的全连接层。这种紧密连接的结构在数据量较大或是隐藏层中的神经感知单元数量增加时，网络模型的参数数量将会急剧增加，不仅会使算法的计算负担加重，还会提高过拟合的可能性。为了避免这些问题，CNN 采用卷积层、非线性激活函数、池化层和全连接层等部分进行网络的搭建，并在网络中利用权值共享和局部连接对模型进行改进。其中，权值共享是指同一个卷积核与输入的不同局部区域计算时，卷积核内部的权重系数是共享的。局部连接是指卷积层中的每个卷积核只与上一层中的局部区域相连，从而大幅度减少网络中的参数数量，如图 2.9 所示。

卷积层 (Conv) 在 CNN 中主要被用来提取输入数据的特征，并利用权值共享和局部连接极大地减少网络模型的计算参数。每层卷积层都由各种各样的卷积核组成，将上一层的输出作为当前层的输入并进行充分融合，从而提取语义信息更加丰富的特征。无论在卷积层中包含有多少个卷积核，无论各个卷积核内部的权重系数的形式如何，卷积层都可以被看作一个线性函数。作为卷积层提取特征的核心部分，卷积操作在各个卷积层中将通道和空间级别的局部特征融合，并提取出信息量丰富的特征。在设计 CNN 架构时，往往需要对卷积层的超参数进行人工设定，例如，卷积核的尺寸 (size)、卷积操作的步长 (stride) 和卷积操作

的填充方式 (padding) 等。当确定上述的超参数后, 各个卷积层输出的大小可由下列公式计算得出:

$$size_{out} = \frac{size_{in} - size_{kernel} + 2 * padding}{stride} + 1 \quad (2.8)$$

非线性激活函数通常会被运用在卷积层的后面, 主要用来对上一层的输出做非线性映射, 是 CNN 能够拟合各种复杂函数至关重要的部分。在非线性映射时, 一些相对较弱的激活值在经过非线性激活函数处理后会变为 0。这样不仅能够使网络模型更多地关注激活值较大的特征, 还能一定程度上提高模型的非线性表达能力, 使算法适应复杂的应用场景。常用的激活函数有 Sigmoid (公式 2.9)、ReLU (公式 2.10) 等。其中, Sigmoid 激活函数会将输入的激活值限制在 0~1 之间, 但当 CNN 的卷积层数较多时, 梯度会在经过逐层的反向传播后逐渐消失, 即出现梯度消失的情况, 使训练的收敛速度明显减慢。ReLU 激活函数会将输入中的负值滤除, 并在输入的激活值为正时保证梯度恒为 1。这样, 梯度消失的概率就会极大地降低, 保证网络模型的训练速度与准确性。所以, ReLU 经常被用于 CNN 的训练中。上述两种激活函数的函数曲线如图 2.11 所示。

$$g_{Sigmoid}(x) = \frac{1}{(1 + e^{-x})} \quad (2.9)$$

$$g_{ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{else} \end{cases} \quad (2.10)$$

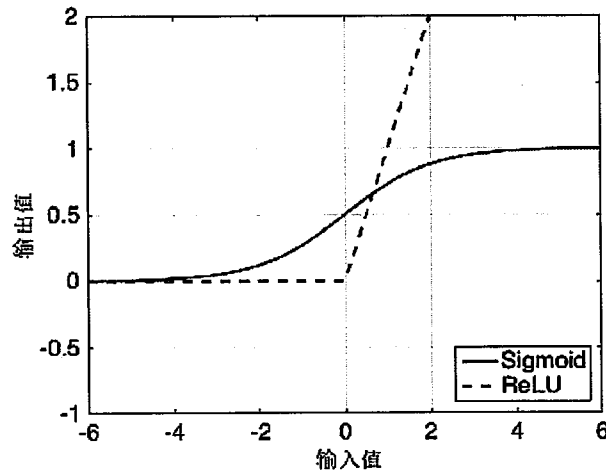


图 2.11 Sigmoid 和 ReLU 的函数曲线

Figure 2.11 The curve of the Sigmoid function and the ReLU function

池化层（Pool）通常会被用于对卷积层输出的特征进行降维，即下采样或欠采样。其主要目的是减少计算量和过拟合的可能性，同时降低网络模型中噪声的影响，使提取的特征更加鲁棒，提高网络模型的容错性。在特征提取的过程中，即使图像发生微小的偏移，池化层处理后的结果也不会发生明显变化，增强了 CNN 的平移不变性。并且，池化层中没有参数，在进行反向传播时只需要对其梯度进行计算并向上一层传递即可，不需要对参数进行任何更新。池化层中常用的操作主要有两种，即平均池化（Average Pooling）与最大池化（Max Pooling）。其中，平均池化是将输入中与过滤器对应尺寸的局部区域取平均值后作为该区域的输出。最大池化是将输入中与过滤器对应尺寸的局部区域取最大值后作为该区域的输出，是 CNN 中最常用的池化操作，如图 2.12 所示。

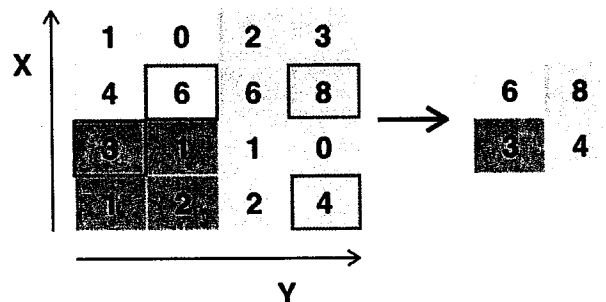


图 2.12 最大池化示意图

Figure 2.12 The sketch map of the max pooling

全连接层通常被应用于 CNN 的末端，是一种全连接的前馈神经网络。在经过前述三个部分的处理后，网络模型会提取出具有丰富语义信息、高质量的特征数据。将提取的特征输入到全连接层中进行决策，从而得到最后的回归或分类结果。然而，采用全连接层的 CNN 是图像级别的，很难做到更加精细的识别与定位。并且，全连接层的引入使输入数据的尺寸受到了限制，给网络模型训练与应用带来了局限性。所以，一些优秀的算法采用卷积层对全连接层进行替代，提出具有像素级精度的全卷积网络（Fully Convolutional Network, FCN）。

2.2.3 几种典型的卷积神经网络模型

近些年，卷积神经网络的迅猛发展引起了研究人员的广泛关注。他们不仅在原有基础上对网络进行改进和完善，还将其广泛应用在目标检测、目标跟踪、语

义分割等计算机视觉领域，并取得了前所未有的突破。下面，本小节将对几种常用的卷积神经网络模型进行介绍。

(1) AlexNet 网络模型

AlexNet^[114]网络模型是由 Alex Krizhevsky 在 2012 年首次提出的，其在 LeNet^[115]的基础上对网络结构进行加深并在 2012 年的 ImageNet 大赛上夺冠，重新掀起了 CNN 乃至深度学习的热潮。如图 2.13 所示^[114]，AlexNet 网络模型主要由五个卷积层和三个全连接层构成，包含了 650,000 个神经感知单元和 60,000,000 个网络参数，并在网络模型最后采用 Softmax 模型进行多分类。

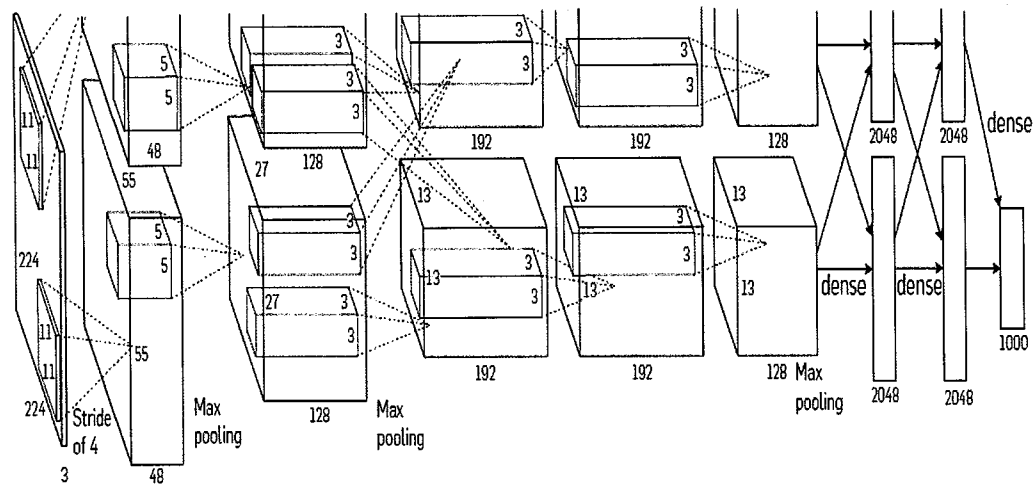


图 2.13 AlexNet 网络模型结构

Figure 2.13 The structure of the AlexNet

与基础的多层神经网络不同，AlexNet 采用 ReLU 激活函数来代替基础的 Sigmoid 激活函数，从而在一定程度上避免了梯度消失的问题。与此同时，为了防止模型训练的时候出现过拟合的情况，AlexNet 将数据扩充 (Data Augmentation) 与 Dropout 引入到了模型中，提升了模型的鲁棒性与稳定性。其中，数据扩充是一种较为简单的防止过拟合的方法，只需要将原始的数据进行适当的变换，就能够得到更多具有一定差异的数据集。Dropout 是在 CNN 的训练过程中经常采用并且效果较好的重要方法。其主要是在训练过程中，将网络中的神经感知单元按照一定的概率暂时丢弃，从而有效避免过拟合并增强了模型的泛化能力。

AlexNet 网络模型采用如下式所示的损失函数：

$$Loss = -\sum_{i=1}^n \hat{y}_i \log y_i \quad (2.11)$$

其中, $\hat{y}_i$ 代表真实数据标签, y_i 代表网络预测标签, n 代表训练用的样本个数。

网络模型权重参数的更新采用下式:

$$\mathbf{w}_j' = \mathbf{w}_j - \frac{\eta}{n} \sum_{i=1}^n \frac{\partial Loss}{\partial \mathbf{w}} \quad (2.12)$$

其中, $\mathbf{w}_j'$ 为更新后的权重参数, η 代表学习率, j 代表第 j 次迭代。

作为最早在大规模图像分类任务中应用的网络, AlexNet 以其优异的性能脱颖而出并成为了计算机视觉领域十分重要的基础网络。虽然, AlexNet 通过在拥有庞大数据量的 ImageNet 上训练获得了极佳的识别性能。但是, 网络深度较浅、全连接层权重参数过多等因素限制了 AlexNet 的发展及其在跟踪领域的应用。

(2) VGGNet 网络模型

VGGNet^[116]网络模型是牛津大学的研究团队在 2014 年首次提出的。如表 2.1 所示^[116], 该模型的基本架构与 AlexNet 网络模型十分相似, 但在一些细节上有所不同。

表 2.1 VGGNet 网络模型结构

Table 2.1 The structure of the VGGNet

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

从表格中可以看出, VGGNet 的网络层数较比 AlexNet 要更多一些。其中, 最浅的结构包含 11 层, 最深的结构达到了 19 层。另外, VGGNet 包含卷积核的个数更多。以 VGG-19 网络模型为例, 第一个卷积模块中包含了两个卷积层, 每一个卷积层由 64 种不同的 3×3 卷积核构成。第二个卷积模块同样由两个卷积层构成, 每一个卷积层包含的 3×3 卷积核的个数增加到了 128 个。第三个、第四个和第五个卷积模块都包含了 4 个卷积层, 每一层包含的 3×3 卷积核的个数成倍增长。这种利用连续多个 (3 或 4 个) 3×3 卷积核对尺寸较大的卷积核进行替代的模式, 不仅能引入更多的非线性函数并提高决策函数的判决力, 还能减少网络模型的参数。在网络结构后端, 网络模型采用与 AlexNet 一样的全连接层进行决策, 并同样采用 Dropout 方式进行优化, 从而有效避免过拟合并增强了模型的泛化能力。

目前, VGGNet 以其优异的性能被广泛应用在目标检测、语义分割和目标跟踪任务中。其中, VGG-16 和 VGG-19 是最常用的基础网络模型。虽然, VGGNet 的性能较比 AlexNet 已经有了明显的提升, 但其计算量也明显提高并在训练过程中耗费了更多的资源, 训练一个 VGGNet 网络模型花费的时间也明显增多。

(3) ResNet 网络模型

随着 CNN 研究的不断深入, 研究人员们发现当网络模型的层数增加到一定程度后, 网络模型的性能将会达到饱和。如果继续对模型的层数进行加深, 模型的性能反而会随之退化, 训练的难度也会不断增加。为了克服上述问题, He 等人^[117]首次于 2016 年提出了 ResNet 网络模型。该网络模型在不同卷积模块之间加入了跳跃连接, 并将网络设计成 $X_{i+1} = F(X_i) + X_i$ 的形式。其中, X_i 为上一个残差模块的输出, X_{i+1} 为当前残差模块期望的输出, $F(X_i)$ 为当前残差模块需要学习的残差函数。由此可以看出, ResNet 网络模型训练的目标不再是得到期望的输出, 而是对恒等映射与期望输出之间的残差进行学习。这种残差更容易优化, 并且在训练过程中进行反向传播时, 这种跳跃连接的结构在网络模型不断加深的过程中可以有效地避免梯度消失、网络模型退化的问题, 从而极大地提高了网络模型的性能。ResNet 的结构示意图如图 2.14 所示^[117], 不同层数 ResNet 网络模型的详细结构参数如表 2.2 所示^[117]。

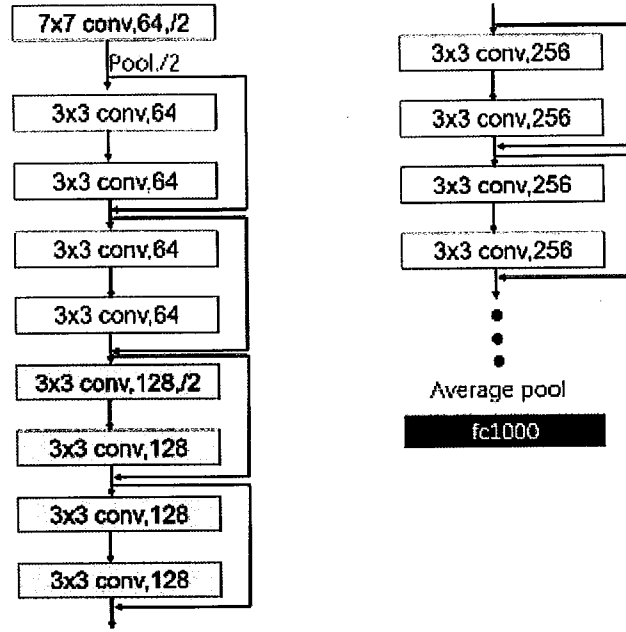


图 2.14 ResNet 网络模型结构示意图

Figure 2.14 The structure of the VGGNet

表 2.2 具有不同结构的 ResNet 网络模型

Table 2.2 The ResNet model with different structures

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

通过 ResNet 的结构可以看出, ResNet 较比 AlexNet 和 VGGNet 减少了全连接层的使用。同时, ResNet 将批归一化 (Batch Normalization, BN) 层引入到了网络模型中, 不仅使网络训练时变得更加稳定, 还允许网络采用更高的学习率和

更低概率的 Dropout, 从而在防止过拟合的同时提高训练的收敛速度。凭借着出众的性能, ResNet 已经成为许多优秀的目标识别、目标检测和语义分割等算法的基础网络模型。并且, 近几年许多优秀的目标跟踪算法也逐渐引入了 ResNet, 使算法的跟踪性能与稳定性有了明显的提高。

本小节介绍的三种典型的卷积神经网络模型是近些年应用较为广泛, 并具有一定影响力的重要模型。AlexNet 是最早在大规模图像分类任务中应用的卷积神经网络模型, 其中的 ReLU 非线性激活函数和 Dropout 方法对后续的先进模型产生了重要影响。VGGNet 利用连续多个卷积核对尺寸较大的卷积核进行替代的模式, 增强了网络的决策能力与表达能力, 成为后续众多优秀卷积神经网络模型的基础。ResNet 利用跳跃连接和残差模块克服了深层网络模型训练的问题, 使深层网络模型的性能有了明显的提高, 并促使卷积神经网络模型向着深度不断增加的趋势发展。

2.3 孪生卷积神经网络

近些年, 一些基于 CNN 网络模型的判别式目标跟踪算法以其优异的跟踪性能获得了研究人员广泛的关注。当采用端到端学习去训练 CNN 时, 虽然得到的分类器能够在线自适应的进行调整, 但模型的在线更新依然会带来几个问题。一方面, CNN 需要采用反向传播算法进行网络的在线更新, 当网络较深时, 反向传播算法的时耗会急剧增加, 极大地影响了算法的实时性。另一方面, 在线跟踪过程中采集的新样本会非常少, 在这种情况下更新的模型很容易过拟合, 从而使算法的准确性和鲁棒性明显下降。为了克服上述问题, 一些学者提出基于 CNN 网络模型的生成式方法来提升跟踪算法的鲁棒性与实时性。该方法采用 CNN 提取候选样本与目标模板的深度特征, 通过学习一个精确、鲁棒的相似度量函数, 实现深度特征之间的精确匹配。所以, 这种方法也可以称为基于深度特征匹配的目标跟踪算法。最近几年, 凭借在学习相似度量函数上的极大优势, 孪生卷积神经网络已经被广泛应用在基于深度特征匹配的跟踪算法中, 并以其均衡的鲁棒性、准确性和实时性获得了较多的关注。下面, 本节将对孪生卷积神经网络和经典的 SiamFC 目标跟踪算法进行概述。

2.3.1 孪生卷积神经网络概述

孪生网络 (Siamese Network) 是近些年被提出的用于计算输入间相似度的网络结构。该结构通常包含两个分支,能够分别处理对应的输入数据。网络训练时,将一对待匹配数据分别输入到对应的分支中,利用参数共享的网络分别提取数据的特征,并对提取到的特征进行相似度计算。将匹配结果与真实结果对比并进行反向传播,从而实现网络参数的更新。孪生网络的基本结构如图 2.15 所示。

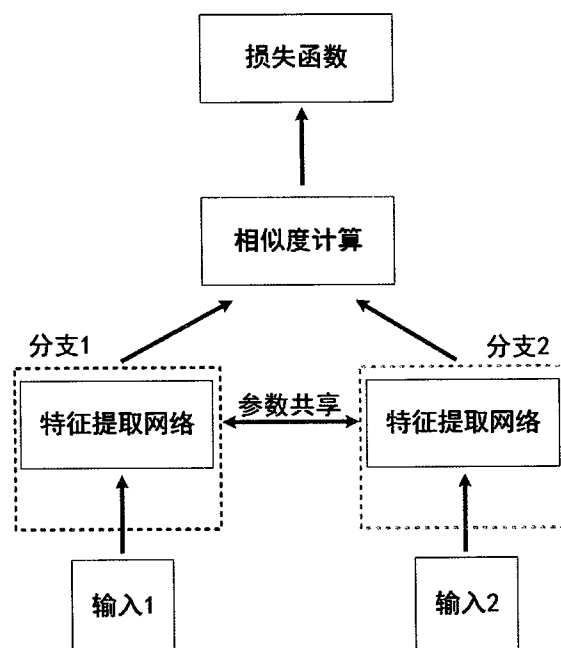


图 2.15 孪生网络的基本结构

Figure 2.15 The basic structure of the Siamese network

其中,用于特征提取的网络由于需要参数共享必须采用结构相同的网络模型。当特征提取网络选取 CNN 网络模型时,就构成了在基于深度特征匹配的跟踪算法中应用最为广泛的孪生卷积神经网络。

在目标跟踪算法中应用的孪生卷积神经网络一般会采用离线的方式训练网络模型。在训练过程中,该网络会将两个分支分别作为模板分支与搜索区域分支,并将剪裁过的模板图像与待匹配图像输入到对应的分支中。如果模板与待匹配图像来自同一类正样本对,则 CNN 提取出的特征会非常相似。反之,若模板与待匹配图像来自负样本对或不同类别的样本,则 CNN 提取出的特征会有明显的差距。孪生卷积神经网络就是采用这样的方式提升了泛化能力,使其可以在只有少量样本的情况下实现准确、鲁棒地跟踪。

在进行相似度计算时，一般会采用欧式距离或余弦距离作为相似度量函数。具体的计算公式如下：

$$sim(\mathbf{x}, \mathbf{y}) = dist(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.13)$$

$$sim(\mathbf{x}, \mathbf{y}) = \cos \theta = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \quad (2.14)$$

其中，当模板与搜索区域分支中的待匹配图像在光照和背景差异较大时，余弦距离会相对更加鲁棒一些。与测量特征距离不同，现有的一些算法还会采用交叉相关系数等来衡量两者之间的相似度，从而在保证精度的同时提高计算效率。

目前，孪生卷积神经网络已经被广泛应用在基于深度特征匹配的跟踪算法中。其中，Bertinetto 等人^[90]提出的 SiamFC 跟踪算法尤为引人注目。该算法首次利用全卷积网络来提取模板和搜索区域的深度特征，并将输出的深度特征运用交叉相关计算一次性得到匹配后的响应图。该算法不仅在跟踪性能上表现优异，在运行速度上也完美达到了实时性的要求。之后，许多学者都以此算法为基础进行改进和创新，为基于深度特征匹配的生成式目标跟踪算法开辟了新的分支。因此，下一小节将对 SiamFC 目标跟踪算法做简要介绍。

2.3.2 SiamFC 目标跟踪算法概述

SiamFC 目标跟踪算法以孪生卷积神经网络为基础结构，并利用共享参数的全卷积网络来提取模板和搜索区域的深度特征。其中，全卷积网络主要以经典的 AlexNet 网络模型为基础，并将其中的五个卷积层应用到网络架构中。当第一帧确定的目标模板 Z 和每一帧确定的搜索区域 X 经过全卷积网络处理后，算法会将得到的特征图进行交叉相关计算，从而获取响应图并能够进行精确定位。SiamFC 算法的结构框架如图 2.16 所示。

SiamFC 算法的核心主要包含两个部分，一个部分是特征提取网络，另一个是相似度量函数。算法得到响应图的过程可以由下式概括：

$$f(Z, X) = F_{\phi}(Z) * F_{\phi}(X) + \mathbf{b} \quad (2.15)$$

其中，特征提取网络 $F_{\phi}(\cdot)$ 主要采用的是离线训练好的全卷积网络。为了使

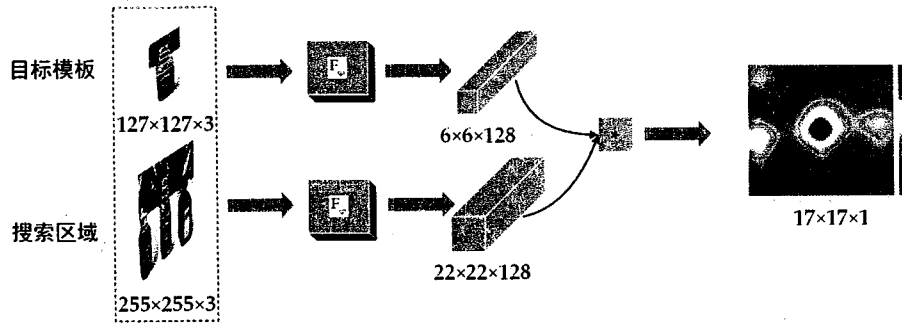


图 2.16 SiamFC 算法的结构框架

Figure 2.16 The framework of the Siamese algorithm

网络具有更强大、更鲁棒的特征提取能力，SiamFC 算法将 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [118] 的数据集作为训练数据集。该数据集包含了大量的视频数据，并且训练用的模板图像和搜索区域的图像都截选自同一个视频的不同帧。相似度量函数 $g(\cdot)$ 不再采用简单的特征距离计算，而是借鉴了相关滤波算法中的思想，将频域上的相关操作转换为 CNN 中的二维卷积操作，将模板分支提取的特征 $F_\phi(Z)$ 在搜索区域分支提取的特征 $F_\phi(X)$ 上进行滑动窗操作，从而在保证精度的同时提高计算效率。 $\mathbf{b}$ 代表一个二维偏置矩阵，矩阵中的每个元素代表响应图中对应位置的偏置系数，并且这些系数都是在算法的训练过程中学习到的，能够使相似度量变得更加鲁棒、准确、可靠。与此同时，为了在跟踪过程中不断地适应目标的尺寸变化，SiamFC 算法采用了多尺度的搜索策略，在 $1.0375^{[-1,0,1]}$ 三个尺度上对响应图进行搜索，从而在后续过程中能够对目标的位置和尺度进行精确判断。

表 2.3 SiamFC 的特征提取网络结构

Table 2.3 The structure of the feature extraction network in SiamFC

网络层	卷积核大小	分组滤波器 个数	步长	目标区域的激 活图大小	搜索区域的激 活图大小
输入				127x127x3	255x255x3
Conv1	3x11x11x96		2	59x59x96	123x123x96
Pool1	3x3		2	29x29x96	61x61x96
Conv2	48x5x5x128	2	1	25x25x256	57x57x256
Pool2	3x3		2	12x12x256	28x28x256
Conv3	256x3x3x384		1	10x10x384	26x26x384
Conv4	192x3x3x192	2	1	8x8x384	24x24x384
Conv5	128x3x3x192	2	1	6x6x256	22x22x256

SiamFC 算法的特征提取网络主要以经典的 AlexNet 网络模型为基础，具体的结构如表 2.3 所示。不同于目标检测任务，目标跟踪任务不需要对图像进行详细的分类，也不需要层数特别深的网络去提取高层的抽象特征。所以，SiamFC 算法只采用了 AlexNet 中的五个卷积层，并对其中的第二个、第四个及第五个卷积层进行了分组操作。卷积层的分组不仅提高了特征提取的效率，还在一定程度上减少了网络模型的参数量，提升了网络在训练和跟踪推理过程中的实时性。另外，由于目标跟踪任务往往具有更加精确的目标定位需求，SiamFC 的特征提取网络只在前两个卷积层后采用了最大池化层，减少了有用特征的损失并保证了算法的精度。与此同时，网络中不再使用全连接层并在训练过程中加入了 BN 层，从而在保证输入数据的尺寸不再受限的同时，有效提升网络训练时的收敛速度并适当避免了过拟合的问题。

SiamFC 算法训练时对应于响应图中某一点的损失函数如下：

$$l(k, s) = \log(1 + \exp(-ks)) \quad (2.16)$$

其中， $k \in \{-1, +1\}$ 代表真实标签， s 代表响应图中某一点的预测值。所以，SiamFC 算法训练时总的损失函数如下：

$$loss_{training}(\mathbf{k}, \mathbf{s}) = \frac{1}{|M|} \sum_{p \in M} l(\mathbf{k}[p], \mathbf{s}[p]) \quad (2.17)$$

其中， M 代表整个响应图的大小， $p \in M$ 代表响应图中的每个点。算法训练过程中就是利用上述的损失函数和重复不断的反向传播，使得真实标签 k 与预测值 s 越来越接近。

SiamFC 算法在跟踪过程中为了提高目标定位的精确性，其利用上一帧中的目标信息作为先验信息，并以上一帧的目标为中心来确定搜索区域，充分利用了视频或图像序列中相邻帧之间的相关性。在经过交叉相关计算并得到响应图后，算法还会运用惩罚项和余弦窗进行处理，从而提高算法的鲁棒性。从图 2.16 可以看出，算法最终输出的响应图尺寸为 17×17 。为了获取目标最终的精确位置，SiamFC 算法在最后会对响应图进行双立方插值，将响应图的尺寸变换为与原始图像相同的尺寸，从而获取目标在图像中的精确位置。

SiamFC 跟踪算法不仅以其均衡的鲁棒性、准确性和实时性获得了学者们的赞同与关注，还为基于深度特征匹配的生成式目标跟踪算法开辟了新的分支。近些年，DSiam^[92]、SiamRPN^[95]、DaSiamRPN^[96]和 SiamRPN++^[97]等众多优秀的基于深度特征匹配的目标跟踪算法，都是在 SiamFC 的基础上改进并提出的。因此，本论文的主要研究内容也依托于 SiamFC 的基本架构，对复杂场景下基于深度特征匹配的目标跟踪算法的局限性进行改进。

2.4 测试数据集及评价指标

作为近些年计算机视觉领域中最为重要的研究方向之一，每年都会有大量优秀的目标跟踪算法被提出。为了能够对这些跟踪算法的性能做出完整、准确、公正的评价，学者们构建并公开了许多测试专用的数据集以对比不同跟踪算法的性能，例如，通过无人机航拍获取的彩色视频数据集 UAV123^[119]、全彩色视频数据集 TC-128^[120]，综合性数据集 OTB（OTB-2013^[121]和 OTB-2015^[122]）和 VOT（VOT-2016^[123]和 VOT-2018^[124]）等。其中，OTB 和 VOT 是应用最广泛且最具代表性的测试数据集。这两种数据集中的测试序列均包含一种或多种具有挑战性的干扰因素，能够对跟踪算法在不同复杂场景下的优劣进行充分评估。下面，本节将分别对 OTB 和 VOT 数据集及相应的评价指标做简要介绍。

2.4.1 OTB 数据集及评价指标

OTB 数据集中的测试序列包含了完整的标注信息和种类繁多的跟踪目标。该数据集主要包含两个版本：一个版本是由 50 个测试序列组成的 OTB-2013，另一个版本是由 100 个测试序列组成的 OTB-2015，是 OTB-2013 的扩展集并简称为 OTB-100。OTB 数据集中的测试序列均包含一种或多种具有挑战性的干扰因素，这些干扰因素主要分为 11 种：背景杂乱（Background Clutter, BC）、形变（Deformation, DEF）、光照变化（Illumination Variation, IV）、平面内旋转（In-plane Rotation, IPR）、遮挡（Occlusion, OCC）、出视场（Out of View, OV）、平面外旋转（Out-of-plane Rotation, OPR）、快速运动（Fast Motion, FM）、运动模糊（Motion Blur, MB）、尺度变化（Scale Variation, SV）、低分辨率（Low Resolution, LR）。

OTB 数据集常采用的评价指标主要包含：中心位置误差（Center Location

Error, CLE)、重叠率 (Overlap Rate, OR)、成功率 (Success Rate, SR) 和曲线下面积 (Area Under Curve, AUC)。中心位置误差代表着目标真实位置与每一帧中算法预测的目标中心位置之间的欧几里得距离, 该误差的值越小说明算法的跟踪效果越好, 计算公式如下:

$$CLE = \sqrt{(x_t - x_{gt})^2 + (y_t - y_{gt})^2} \quad (2.18)$$

式中, (x_t, y_t) 代表算法预测的目标中心位置, (x_{gt}, y_{gt}) 代表目标的真实位置。重叠率是比中心位置误差更加全面合理的评价指标, 其不仅考虑了目标的中心位置信息, 同时还参考了目标的尺寸大小。重叠率的取值范围为[0,1], 取值越大说明算法的跟踪效果越好, 计算公式如下:

$$OR = \frac{Area_i \cap Area_{gt}}{Area_i \cup Area_{gt}} \quad (2.19)$$

式中, $Area_i$ 代表算法预测的目标区域, $Area_{gt}$ 代表真实的目标区域。成功率通常用来将无效帧进行筛选, 通过设定一个阈值将评分高于阈值的帧视为跟踪成功的有效帧, 将评分低于阈值的帧视为跟踪失败的无效帧。成功率的取值范围同样为[0,1], 取值越大说明算法的跟踪效果越好, 计算公式如下:

$$SR = \frac{F_s}{F_s + F_f} \quad (2.20)$$

式中, F_s 代表跟踪成功的有效帧, F_f 代表跟踪失败的无效帧。曲线下面积被定义为绘制曲线与坐标轴围成的面积, 是一种通常被用来衡量算法性能优劣的评价指标。

在跟踪算法测试的过程中, 为了能够清晰的展现出各种算法跟踪性能的优劣, 通常会采用精度曲线 (Precision Plot) 和成功曲线 (Success Plot) 进行直观的比较。其中, 精度曲线中的横坐标代表的是中心位置误差, 纵坐标代表的是以对应的中心位置误差为阈值时的成功率。成功曲线中的横坐标代表的是重叠率, 纵坐标代表的是以对应的重叠率为阈值时的成功率。与此同时, 利用 OTB 数据集进行评估时通常会采用一次通过的评价方法 (One-Pass Evaluation, OPE)。该方法旨在使用数据集中每个测试序列初始帧的目标信息对跟踪算法进行初始化, 之后会在整个测试序列上运行算法并绘制相应的精度曲线与成功曲线。

2.4.2 VOT 数据集及评价指标

VOT 数据集是除了 OTB 数据集以外另一种应用广泛的测试数据集。该数据集在 2013 年被首次构建并提出，目前为止已经被进行多次的改进与完善，主要用来对短期目标跟踪算法进行跟踪性能的评价与分析。作为 VOT 数据集中最常用的数据集，VOT-2016 包含了 60 个具有多种挑战性干扰因素的测试序列，其中有少部分的测试序列与 OTB 数据集相同。这些测试序列中包含了动物、人脸、玩具和交通工具等常见的目标，能够对跟踪算法在不同复杂场景下的优劣进行充分评估。

VOT 数据集常采用的评价指标主要包含：准确度（Accuracy, A）、鲁棒性（Robustness, R）和预期平均重叠率（Expected Average Overlap, EAO）。准确度主要用来对跟踪算法的精度进行评价，其数值越大代表算法的精度越高。准确度的计算与 OTB 数据集中的 OR 计算类似，计算公式如下：

$$\phi_t = \frac{Area_t^T \cap Area_t^{GT}}{Area_t^T \cup Area_t^{GT}} \quad (2.21)$$

式中， $Area_t^T$ 代表第 t 帧算法预测的目标区域， $Area_t^{GT}$ 代表第 t 帧真实的目标区域， ϕ_t 代表第 t 帧的精确度。为了更详细的对精确度进行表示，一般会在计算时将 $\phi_t(i, n)$ 设为第 i 个跟踪算法重复 n 次时在第 t 帧的精确度，计算公式如下：

$$\phi_t(i) = \frac{1}{N_{rep}} \sum_{n=1}^{N_{rep}} \phi_t(i, n) \quad (2.22)$$

式中， N_{rep} 代表总的重复次数， $\phi_t(i)$ 代表第 t 帧的平均精确度。所以，第 i 个跟踪算法总的平均精确度为：

$$\Phi_A(i) = \frac{1}{L_{valid}} \sum_{t=1}^{L_{valid}} \phi_t(i) \quad (2.23)$$

式中， L_{valid} 代表有效帧的数量。鲁棒性主要用来对跟踪算法的稳定性进行评价，其数值越大代表算法的稳定性越差。由于评估算法的鲁棒性一般会对失败次数进行统计，所以鲁棒性一般会被表示为平均失败次数（Failures），计算公式如下：

$$F_R(i) = \frac{1}{N_{rep}} \sum_{n=1}^{N_{rep}} f(i, n) \quad (2.24)$$

式中, $f(i, n)$ 代表第 i 个跟踪算法重复 n 次时的失败次数, $F_R(i)$ 代表跟踪算法总的平均失败次数。EAO 主要对准确度与鲁棒性的原始数据进行充分利用, 结合测试序列长度和重叠率对跟踪算法进行综合性评价。在利用 EAO 做评价时, 首先会将测试序列按照长度进行分类, 并在长度为 L_s 的序列上运行算法并得到在该序列上的平均精确度:

$$\Phi_{L_s} = \frac{1}{L_s} \sum_{t=1}^{L_s} \Phi(t) \quad (2.25)$$

由于长度为 L_s 的测试序列不止一个, 所以算法要在这 M_s 个序列上全部运行一遍, 并取平均为:

$$\hat{\Phi}_{L_s} = \frac{1}{M_s} \sum_{m=1}^{M_s} \Phi_{L_s} \quad (2.26)$$

另外, 如果在评测过程中对不同 L_s 值 $[L_{lo}, L_{hi}]$ 的 $\hat{\Phi}_{L_s}$ 再次取平均, 则会通过下式得到值 $\hat{\Phi}$:

$$\hat{\Phi} = \frac{1}{L_{hi} - L_{lo}} \sum_{L_s=L_{lo}}^{L_{hi}} \hat{\Phi}_{L_s} \quad (2.27)$$

其中, $\hat{\Phi}_{L_s}$ 常被用来绘制 EAO 曲线, $\hat{\Phi}$ 常被用来绘制 EAO 点图。

2.5 本章小结

本章首先对单目标跟踪算法的基本框架和关键部分进行简要介绍。接着, 对深度学习基本原理进行简述, 并深入介绍了卷积神经网络以及几种典型的卷积神经网络模型。然后, 对基于深度特征匹配的目标跟踪算法中常见的孪生网络架构进行简要描述, 并对经典的 SiamFC 算法原理进行概述。最后, 对本论文中用于实验分析的测试数据集及评价指标进行介绍。

第3章 结合再检测机制与自适应模板更新的

孪生卷积神经网络目标跟踪算法

3.1 引言

在跟踪过程中,当背景中出现相似干扰时,基于深度特征匹配的目标跟踪算法生成的响应图中易出现多个波峰,从而造成跟踪框不稳定甚至跟踪目标丢失的情况。与此同时,跟踪过程中遮挡、形变等复杂因素也时刻影响着基于深度特征匹配的目标跟踪算法的性能。为了克服上述问题,本章提出一种结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。该算法以 SiamFC 算法为基础,利用一种高效且准确的再检测孪生卷积神经网络作为再检测机制来应对响应图中的多波峰。再检测网络在出现的各个波峰周围采样并进行实时、准确的匹配计算,从而实现对于干扰的筛选与排除。另外,本章算法提出一种基于生成式模型的自适应模板更新方法。该模板更新方法将评估过的可靠跟踪结果进行特征提取与概率统计,利用统计结果与已有的目标模板特征进行自适应融合,从而实现模板的自适应更新。仿真实验结果表明,本章提出的算法能较好应对相似背景干扰、遮挡、目标形变等难题,有效提升了算法在复杂场景下的跟踪精度与鲁棒性。

3.2 结合再检测机制与自适应模板更新的孪生卷积神经网络

在本节中,详细介绍本章所提出的结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法,并将其简称为 Siam-RM (Object Tracking Framework with Siamese Network and Re-detection Mechanism)。首先,本节将对用于跟踪的孪生卷积神经网络模型进行介绍。接着,本节对基于再检测孪生卷积神经网络的再检测机制进行阐述。最后,本节对目标模板的构建与自适应更新进行论述。Siam-RM 跟踪算法的框架如图 3.1 所示。

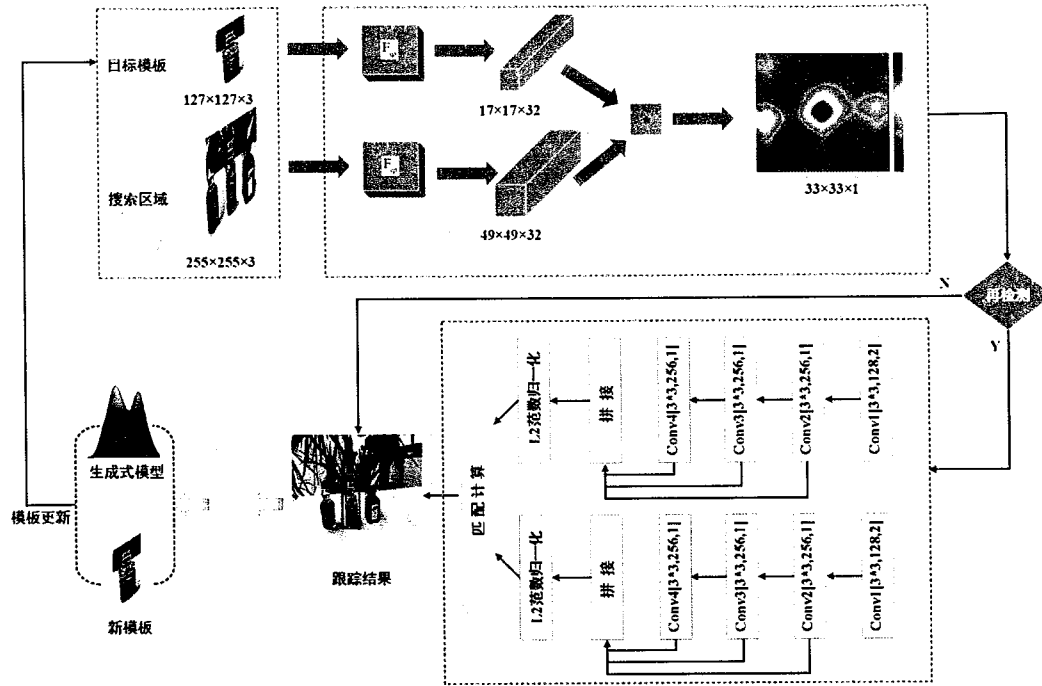


图 3.1 Siam-RM 算法的结构框架

Figure 3.1 The framework of the Siam-RM algorithm

3.2.1 孪生卷积神经网络

Siam-RM 算法采用孪生卷积神经网络作为主要的跟踪模型。该孪生卷积神经网络以 SiamFC 算法中的网络模型为基础，并在网络结构上进行改进，从而使网络更加高效、准确。

SiamFC 目标跟踪算法利用共享参数的全卷积孪生网络来提取目标模板和搜索区域的深度特征。其中，全卷积网络主要以经典的 AlexNet 网络模型为基础，并将其中的五个卷积层应用到网络架构中。当第一帧确定的目标模板 Z 和每一帧确定的搜索区域 X 经过全卷积网络处理后，算法会将得到的特征图进行交叉相关计算，从而得到响应图并能够进行精确定位。其结构框架和工作原理可参考 2.3.2 小节。

通常情况下，为了保证算法具有较高的实时性，孪生卷积神经网络会采用结构较为简单的网络模型。然而，如果采用和 SiamFC 算法一致的网络模型，最后生成的响应图尺寸会偏小，从而在定位目标的过程中带来一定的误差，并不利于再检测机制的处理。所以，Siam-RM 算法以 SiamFC 算法中的网络模型为基础，

并参考 Baseline-conv5 模型^[91]进行设计,成功将改进后的网络模型应用于深度特征的提取。改进后的网络模型结构如下表所示。

表 3.1 Siam-RM 算法中用于特征提取的网络模型结构

Table 3.1 The structure of the feature extraction network in Siam-RM

层名称	核尺寸	步长	输出尺寸		
			目标模板	搜索区域	通道数
			127×127	255×255	3
Conv1	11×11	2	59×59	123×123	96
Pool1	3×3	2	29×29	61×61	96
Conv2	5×5	1	25×25	57×57	256
Pool2	3×3	1	23×23	55×55	256
Conv3	3×3	1	21×21	53×53	384
Conv4	3×3	1	19×19	51×51	384
Conv5	3×3	1	17×17	49×49	32

从表中可以看出,改进后的网络模型相比 SiamFC 中的网络模型具有更少的步长。因此, Siam-RM 最后得到的响应图具有更大的尺寸,这样能够在定位目标时减少误差的引入,同时为后续的再检测机制提供可靠的位置信息。另外,改进后的网络模型将最后一层输出特征的通道数降为 32,一定程度上能够减少计算量,提升算法的计算速度与实时性。

Siam-RM 网络模型的训练与 SiamFC 的训练过程相同。在训练时, Siam-RM 网络模型会采用与 SiamFC 相同的损失函数(公式 2.17),并利用随机梯度下降(Stochastic Gradient Descent, SGD)进行网络优化:

$$\arg \min_{\omega} \underset{(Z, X, k)}{loss_{train}}(k, f(Z, X; \omega)) \quad (3.1)$$

训练用的样本对是从标注好的数据集中获取的。训练时,算法先从视频序列中选取一帧作为模板图像,并以模板图像中标注的目标为中心进行目标模板的截取。接着,算法以模板图像为中心,在其前后一定范围内进行搜索图像的选取并截取相应的搜索区域。最后,将截取好的目标模板与搜索区域组成一组样本对用于网络模型的训练。具体的训练细节可参考 2.3.2 小节或 SiamFC 原文。

3.2.2 再检测机制

在利用孪生卷积神经网络提取过深度特征后,算法会将得到的目标模板与搜

索区域的特征图进行交叉相关计算，从而获取响应图并能够进行精确定位。通常情况下，响应图中最大响应值的位置对应的就是目标位置。然而，当算法在复杂场景下运行时，由于背景中有时会出现相似背景干扰、遮挡等干扰因素，响应图中往往会出现多个波峰且最大响应值的位置有可能不再对应目标位置。为了避免这类问题，SiamFC 算法中采用余弦窗函数对边缘信息进行抑制。这使得离响应图中心越近的位置具有越大的权重，离响应图中心越远的位置具有越小的权值。虽然，采用这种余弦窗函数抑制干扰的策略能够在一定程度上提高算法的稳定性。但是，当目标与干扰都距离响应图中心较近或目标移动较快时，上述策略的效果将明显降低甚至很容易造成跟踪丢失。因此，Siam-RM 算法引入再检测机制来应对响应图中出现多波峰的情况。

再检测机制经常被应用于长期跟踪中，其主要用途是当跟踪过程中遇到干扰时保证算法的稳定性与准确性，从而在一定程度上提高算法的跟踪性能。Supancic 等人^[126]在算法中采用了一种带有自主学习功能的再检测机制。该机制能够适时的对训练集进行再检测，从而在保证训练集不被污染的同时选择可靠的训练数据去学习目标的外观。Hua 等人^[127]提出一种再检测方法来应对跟踪过程中遇到的严重遮挡和目标形变问题。该方法可以对目标在运动过程中遇到的遮挡和形变进行推断，从而在一定程度上提高跟踪效果，保证算法在复杂场景下的稳定性。Wang 等人^[128]利用再检测机制实现了长期跟踪。该机制可以对分类器产生的分类结果进行筛选，并将被污染的错误结果进行排除。同时，该算法采用的再检测机制能够根据分类结果决定是否对原始的目标模板进行替换，从而保证算法在复杂场景下的鲁棒性。另外，再检测机制在得到越来越多关注的同时也被引入到许多实际应用中，如多目标跟踪^[129-131]和人员再识别^[132,133]等。

受上述讨论的启发，当响应图中出现多波峰等干扰信息时，Siam-RM 采用再检测孪生卷积神经网络作为再检测机制对干扰进行排除，从而提高算法在复杂场景下的准确性与稳定性。再检测孪生卷积神经网络结构如图 3.2 所示。

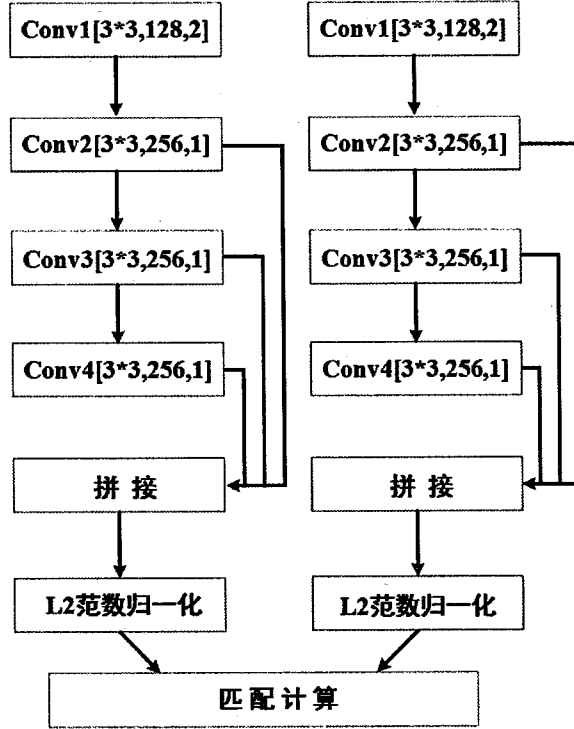


图 3.2 再检测孪生卷积神经网络结构

Figure 3.2 The structure of the re-detection network

从上图中可以看出，再检测孪生卷积神经网络主要包含 4 个卷积层，并将 Conv2、Conv3 和 Conv4 三个卷积层输出的特征进行拼接作为最后的输出。这样的拼接操作既能使输出的卷积特征包含丰富的细节信息，同时还能使其具有更多的语义信息，从而提高特征匹配时的鲁棒性与稳定性。在得到最终的拼接特征后，网络会对两个分支的特征进行 L2 范数归一化，并利用下式进行两个特征间的匹配度计算：

$$R_{1,2} = (\mathbf{F}_\phi^1)^T \mathbf{F}_\phi^2 \quad (3.2)$$

式中， $\mathbf{F}_\phi^1$ 和 $\mathbf{F}_\phi^2$ 代表拼接后的特征张量，上角标 T 代表转置。由于拼接后的特征张量都采用了 L2 范数归一化进行处理，所以匹配度 $R_{1,2}$ 的取值范围为 $[-1,1]$ 。

再检测孪生卷积神经网络也是采用样本对进行训练的。训练时，算法会在一个视频序列中随机的抽取两帧图像，并选取其中一帧作为模板图像，同时会以模板图像中标注的目标为中心进行目标模板的截取。接着，算法会从另一帧以同样的尺寸进行候选样本的截取。如果候选样本与对应帧的真实边界框的重叠率高于 0.75，则可认为该候选样本为正样本。如果重叠率小于 0.4，则可认为该候选样

本为负样本。训练时，将正、负样本与模板图像作为一组正、负样本对输入到网络中进行训练，采用的损失函数主要借鉴 SINT 算法^[86]中的边缘对比损失函数 (Margin Contrastive Loss):

$$\mathcal{L}=y(1-R_{1,2})^2+(1-y)\max(0,R_{1,2}-\varepsilon)^2 \quad (3.3)$$

式中, y 代表样本的标签且 $y \in \{0, +1\}$, ε 代表负样本的匹配度应该满足的最大间距, 由于训练采用的负样本是随机挑选的, 所以可以将图像视为不相关的, ε 可以设置为 0。

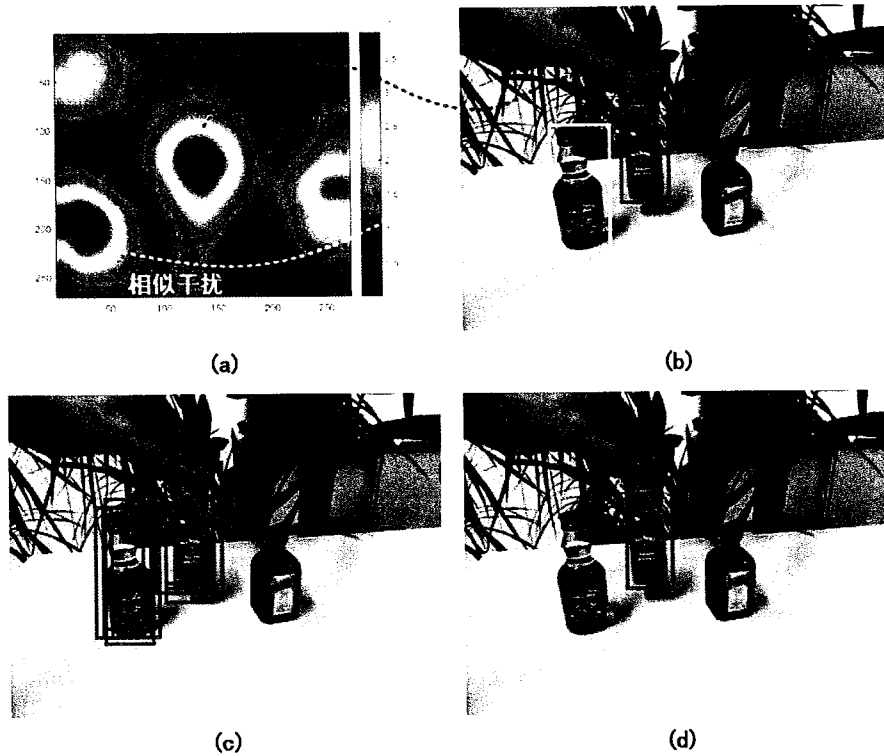


图 3.3 再检测机制工作示意图

Figure 3.3 The sketch map of the re-detection mechanism

再检测机制的示意图如图 3.3 所示。图 3.3 (a) 是 Siam-RM 算法在跟踪阶段利用孪生卷积神经网络获取的响应图, 其明显存在着多波峰的情况。图 3.3 (b) 是响应图所对应的真实搜索图像。从图中可以看出, 红框代表着真实目标并与响应图中的一个波峰对应, 黄框代表着干扰目标并与响应图中的另一个波峰对应。在这种情况下如果不进行再检测处理, 当干扰目标的响应值大于真实目标的响应值时, 算法就会默认为干扰目标为实际跟踪的目标, 从而造成跟踪失败。为此,

我们在算法中设置再检测的阈值为 $Thre_detection$ ，当响应图中的次级波峰与主波峰的比值超过 $Thre_detection$ 时，算法就会启动再检测机制进行处理。如图 3.3 (c) 所示，运行再检测机制时会在次级波峰与主波峰对应的图像位置进行采样，采样的尺寸会以前一帧跟踪到的目标尺寸为基准。采样过后，算法会将前一帧跟踪到的目标作为模板图像，并将采样得到的样本作为候选样本。将模板图像与候选样本以图像对的形式输入到再检测孪生卷积神经网络中进行匹配计算，从而对相似干扰进行排除并确定最后的跟踪目标，如图 3.3 (d) 所示。

3.2.3 自适应模板更新

虽然，Siam-RM 算法能够采用孪生卷积神经网络进行准确跟踪，并利用再检测机制避免相似背景干扰所带来的影响。但是，如果像 SiamFC 算法一样只以第一帧中的目标作为模板，并且在目标进行跟踪时不更新模板，在复杂场景下当目标发生遮挡、形变等较大变化时跟踪算法的性能将会明显降低，从而不能在后续帧中高效、准确地定位目标。因此，Siam-RM 算法利用自适应模板更新来提升算法在复杂场景下的跟踪性能。

模板更新主要应用于基于模板匹配的跟踪算法中^[125,134-136]，该类算法需要对目标模板进行适当的更新，从而使算法能够不断适应目标的外观变化。如果在跟踪过程中用每一帧跟踪到的目标去更新模板，则当目标定位不准确时模板往往会引入背景信息，从而导致跟踪算法的精确度下降。如果这种错误长期累积并得不到纠正，跟踪算法将会出现跟踪漂移甚至是跟踪失败的情况。为此，Jepson 等人^[135]采用三种因素：噪声、稳定性和瞬时因子去描述模板，并采用一种间歇更新的方式对模板进行更新。Matthews 等人^[136]将初始帧中的模板信息与当前帧的目标信息相结合，并采用一种阈值判断的方式进行模板的更新。Ross 等人^[30]从每帧的跟踪结果中筛选出适合的目标模板样本并进行收集。当收集的样本达到一定数量后，算法会采用 R-SVD (rank-R singular value decomposition) 的增量子空间模型进行目标外观的更新。这些模板更新策略不仅能够有效减少模板污染所带来的噪声信息，还能使鲁棒性与准确性得到提高，从而保证算法在复杂场景下依然能够准确、高效、稳定地定位目标。

受上述讨论的启发，Siam-RM 算法引入一种基于生成式模型的自适应模板

更新方法。该方法将评估过的可靠跟踪结果进行特征提取与概率统计，利用统计结果与已有的目标模板特征进行自适应融合，从而实现模板的自适应更新。

自适应模板更新方法是一种基于生成式模型的更新方法，该生成式模型的构建主要借鉴于高斯混合模型(Gaussian Mixture Model, GMM)，其概率分布如下：

$$P(T) = \sum_{k=1}^K \pi_k N(T; \mu_k; I) \quad (3.4)$$

式中， K 代表模型中组件 $N(T; \mu_k; I)$ 的个数， π_k 代表组件 k 的先验权重， μ_k 代表组件 k 的均值。为了减少在高维空间的计算负担，分布中的协方差矩阵在这里设置为单位矩阵 I 。在模型初始化的过程中，算法将之前跟踪到的目标当作模板并通过孪生网络的目标模板分支提取特征，利用提取特征的均值对各个组件进行初始化。当利用模型进行模板更新时，会将新跟踪到的目标输入到目标模板分支中提取特征，并利用下式进行模板特征的更新：

$$T_n = \eta * T_g + (1 - \eta) * \sum_{k=1}^K p(T_k) T_k \quad (3.5)$$

式中， η 代表学习率， T_k 代表构成组件 k 的特征， T_g 代表新得到的特征， $p(T_k)$ 代表新特征对应于高斯混合模型中组件 k 的概率。一般情况下，模板的更新会采用逐帧更新或者基于一定间隔的更新方式。逐帧更新的方式会给算法带来较大的计算负担，使算法的实时性明显下降。基于一定间隔的更新方式往往会引入背景信息，从而在模板更新的过程中引入误差。为此，Siam-RM 算法在模板更新时采用一种鲁棒性较高的模板更新策略。该策略主要依靠平均峰值相关能量 (Average Peak-to-Correlation Energy, APCE) 对响应图的质量进行衡量，其公式如下：

$$APCE = \frac{|M_{max} - M_{min}|^2}{Mean(\sum_{w,h} (M_{w,h} - M_{min})^2)} \quad (3.6)$$

式中， M_{max} 代表响应图 M 中的最大值， M_{min} 代表响应图 M 中的最小值， $M_{w,h}$ 代表响应图 M 中第 w 行和第 h 列的响应值。APCE 的值能够较好地反映出响应图的波动水平和置信程度。当响应图中的波动和噪声较小时，APCE 的值就会变大。反之，当响应图中波动和噪声较大时，APCE 的值就会变小。为此，算法在进行

模板更新时会设置一个阈值 Th_{APCE} 。每次利用孪生卷积神经网络得到响应图后，算法会利用 APCE 对响应图的质量进行衡量。如果获取的 APCE 值大于 Th_{APCE} ，算法则会将当前跟踪到的结果输入到生成式模型中进行目标模板的更新。

为了保证生成式模型具有更加鲁棒、稳定的性能，在算法运行过程中也需要对模型进行适当的更新。当得到新的特征后，会用其构建一个新的组件 g 并将其权重和均值设为 $\pi_g = \eta$ 和 $\mu_g = e_g$ 。如果模型中的组件数量小于 K ，则新的特征将会被直接加入到模型中。如果模型中的组件数量超过 K ，则需要对三种情况进行考虑：

1) 模型中的一个组件权重过低。在这种情况下，会通过设置一个阈值 W_{min} 对各个组件进行筛选。如果一个组件的权重低于 W_{min} ，则会用新的组件 g 替代权重过低的组件；

2) 新特征与现有组件间的欧式距离小于模型中任意两个组件间的欧式距离。在这种情况下，新的组件 g 会和与其距离最小的组件融合；

3) 新特征与现有组件间的欧式距离大于模型中任意两个组件间的欧式距离。在这种情况下，会将模型中距离最近的两个组件融合，并用新组件 g 填补空缺。

在上述情况中，算法会采用下式进行组件的融合：

$$\pi_n = \pi_x + \pi_y \quad (3.7)$$

$$\mu_n = \frac{\pi_x \mu_x + \pi_y \mu_y}{\pi_x + \pi_y} \quad (3.8)$$

3.2.4 算法描述

在跟踪过程中，首先会给定第一帧中的目标信息，Siam-RM 算法根据该信息进行算法初始化，如目标模板截取、模板特征提取和搜索尺度确立等。接着，Siam-RM 算法会在下一帧截取搜索区域，并将搜索区域图像输入到孪生卷积神经网络中进行特征提取，利用交叉相关计算获取响应图。然后，Siam-RM 算法会对响应图进行判断，计算次级波峰与主波峰的比值是否超过阈值 $Th_{re-detection}$ 。如果超过阈值，则会启动再检测机制滤除干扰。如果没有，则直接得到跟踪结果。最后，Siam-RM 跟踪算法会利用 APCE 对当前跟踪结果对应的响应图进行质量

评价, 如果 APCE 的值高于阈值 Th_{APCE} , 则会将新跟踪到的目标输入到孪生卷积神经网络的目标模板分支中提取特征, 并利用生成式模型对模板特征进行更新。

Siam-RM 算法的运行流程如下表所示:

表 3.2 Siam-RM 算法运行流程

Table 3.2 The execution process of Siam-RM

结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法 (Siam-RM)
输入: 预训练好的网络, 初始帧目标信息
Step 1: 利用初始帧目标信息进行算法初始化 (目标模板截取、模板特征提取和搜索尺度确立等)
对于图像序列帧数 $frame=2,3,\dots,N$ 依次循环 Setp2-5
Step 2: 在第 t 帧图像上的 (x_{t-1}, y_{t-1}) 周围确定搜索区域, 利用孪生卷积神经网络获取响应图
Step 3: 计算次级波峰与主波峰的比值并判断是否超过阈值 $Th_{re-detection}$ 。如果高于阈值, 则利用再检测卷积神经网络及式 (3.2) 对干扰进行滤除。反之, 确定当前帧跟踪结果 (x_t, y_t) 和 (w_t, h_t)
Step 4: 通过式 (3.6) 对当前跟踪结果对应的响应图进行质量评价。如果高于阈值 Th_{APCE} , 将新跟踪到的结果输入到孪生卷积神经网络的目标模板分支中提取特征, 利用式 (3.5) 实现模板特征的更新
Step 5: 利用式 (3.7) 和式 (3.8) 对生成式模型进行更新
输出: 当前帧目标的位置 (x_t, y_t) 和大小 (w_t, h_t)

3.3 实验结果与分析

本章算法在 Matlab-R2016a 下利用 MatConvnet^[137]和 MatCaffe^[138]深度学习工具集实现。用于实验的 PC 机配置为: Intel i7-9800X CPU (3.80GHz), 64GB RAM 和一块 NVIDIA TITAN RTX GPU。算法平均运行速度为 24 帧每秒。Siam-RM 算法中的孪生卷积神经网络和再检测孪生卷积神经网络是分别在线下训练的。在训练孪生卷积神经网络时, 采用 ILSVRC-2015 的 VID^[118]数据集进行离线训练, 且网络中每层参数的初始值遵循高斯分布。采用 SGD 方法对网络进行优化, 设置权重衰减 (Weight Decay) 为 0.0005。整个训练过程为 50 个迭代周期, 并设置初始学习率为 0.001。在每个迭代周期结束后会乘以一个固定的常数, 直到最后

一个迭代周期的学习率降为 0.00001。与此同时,训练过程中的动量值设置为 0.9,每个小批量 (mini-batch) 包含 32 个样本。在训练再检测孪生卷积神经网络时,采用 ALOV^[139]数据集进行离线训练,并将和 OTB 测试数据集重合的 12 个视频序列排除。整个训练过程为 15 个迭代周期,采用 Adam^[140]方法进行优化,设置初始学习率为 0.001 并在每个迭代周期后乘以 0.6,每个小批量包含 32 对正样本和 96 对负样本。另外, Siam-RM 算法中生成式模型的参数是通过大量实验验证得到的,学习率 η 设为 0.011,高斯混合模型中组件 $N(T; \mu_k; I)$ 的个数 K 设为 30,模型更新时的阈值 W_{min} 设为 0.0036,响应图质量评价时的阈值 Th_{APCE} 设为 8。

为了客观的对 Siam-RM 算法的性能进行评价,本小节采用 OTB 数据集和 VOT-2016 数据集对 Siam-RM 及其它一些先进的算法进行对比,从而验证本章算法的准确性、鲁棒性及有效性。

(1) OTB 数据集

OTB 数据集集中的测试序列包含了完整的标注信息和种类繁多的跟踪目标。该数据集主要包含两个版本:一个版本是由 50 个测试序列组成的 OTB-2013,另一个版本是由 100 个测试序列组成的 OTB-2015,是 OTB-2013 的扩展集并简称为 OTB-100。本小节的实验采用以上两个数据集,并利用 2.4.1 小节中介绍的评价指标对算法进行评估。其中,参与对比实验的算法包括 SINT^[86]、SiamFC^[90]、GOTURN^[87]、CFNet^[91]、DCFNet^[141]、LCT^[62]、LMCF^[142]、KCF^[59]和 DSST^[61]算法。与此同时,实验采用精度曲线 (Precision Plot) 和成功曲线 (Success Plot) 进行直观的比较,并采用 OPE 评价方法对算法进行评估。

(2) VOT 数据集

实验采用的 VOT 数据集是最常用的 VOT-2016 数据集^[123],该数据集包含了 60 个具有多种挑战性干扰因素的测试序列,其中有少部分的测试序列与 OTB 数据集相同。这些测试序列中包含了动物、人脸、玩具和交通工具等常见的目标,能够对跟踪算法在不同复杂场景下的优劣进行充分评估。在 VOT-2016 数据集上,开展了与其它 14 种跟踪算法的实验对比,包括 ECO-HC^[74]、MDNet_N^[78]、C-COT^[72]等,并利用 2.4.2 小节中介绍的评价指标对算法进行评估。

3.3.1 OTB 数据集实验分析

(1) 定量分析

图 3.4 和图 3.5 分别对 OTB-2013 上和 OTB-2015 上的实验结果进行了展示。从图中可以看出, Siam-RM 算法在 OTB-2013 数据集上排名第二, 与排名第一的 SINT 跟踪算法在精度和重叠率上相差 0.4% 和 0.9%, 比 SiamFC 算法在精度和重叠率上高 6.3% 和 3.4%。Siam-RM 算法在 OTB-2015 数据集上排名第一, 比 SINT 跟踪算法在精度和重叠率上高 5.7% 和 2.1%。比 SiamFC 算法在精度和重叠率上高 7.3% 和 2.8%。上述实验结果充分证明, Siam-RM 算法能够利用再检测机制和自适应模板更新有效提升复杂场景下算法的跟踪精度及准确度。

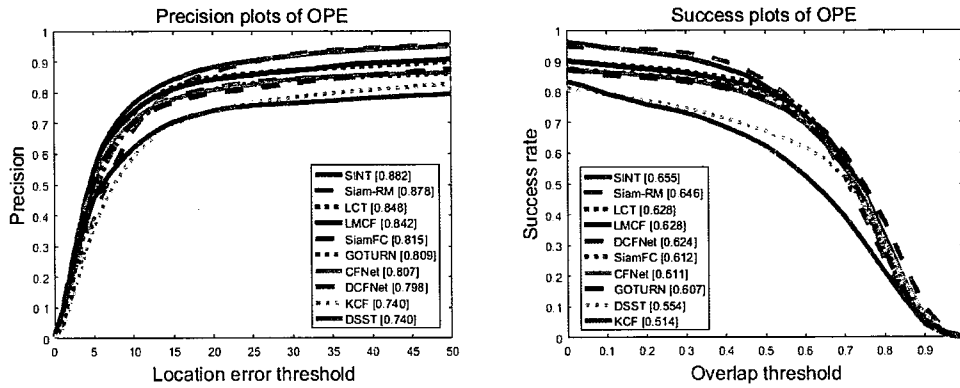


图 3.4 OTB-2013 数据集上的实验结果

Figure 3.4 The tracking results on OTB-2013 dataset

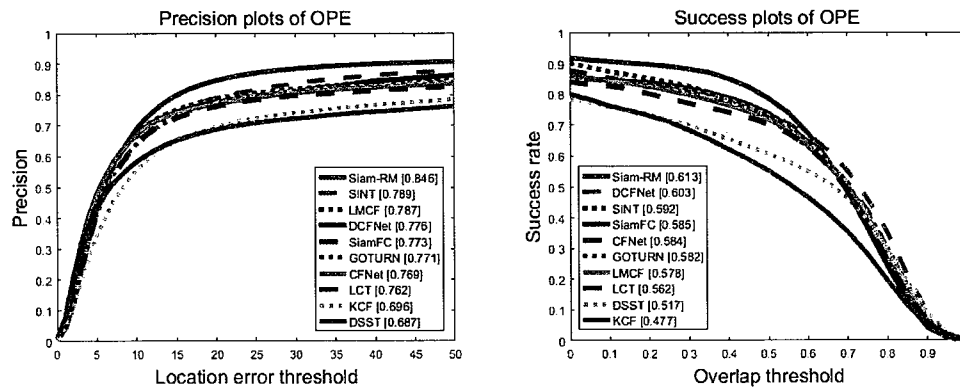


图 3.5 OTB-2015 数据集上的实验结果

Figure 3.5 The tracking results on OTB-2015 dataset

Siam-RM 算法的跟踪效果较比传统的基于浅层特征的算法要优越一些, 如 LCT、KCF 和 DSST 等。这说明 CNN 提取的深度特征要明显优于传统的人工进

行设计的表现特征。特别是 KCF 算法，虽然该算法的核心思想非常优越，但是算法采用 HoG 特征对目标的表现进行描述，该特征通过对图像不同方向的梯度进行统计来获取稳定且具有较高识别性的特征，然而当图像较为平滑时 HoG 特征就会出现失效的状况，因此在应对复杂场景时无法达到理想的跟踪效果。

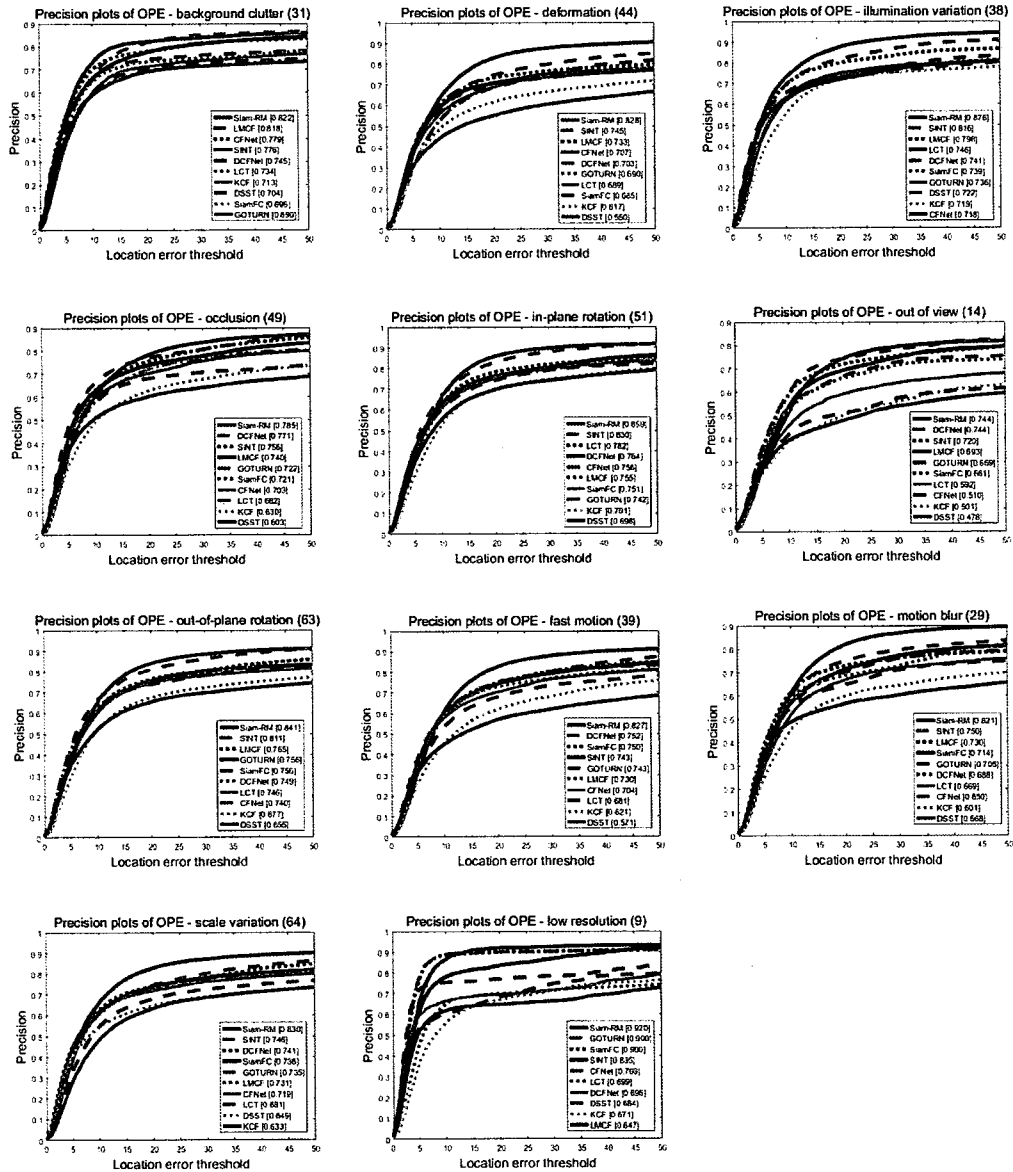


图 3.6 OTB-2015 数据集上 11 种属性下的实验结果（精度）

Figure 3.6 Tracking results on 11 challenging attributes on the OTB-2015 dataset (Precision)

为了进一步进行比较，图 3.6 和图 3.7 展示了各算法在 OTB-2015 数据集上 11 种不同属性（见 2.4.1 小节）下精度和重叠率的比较。

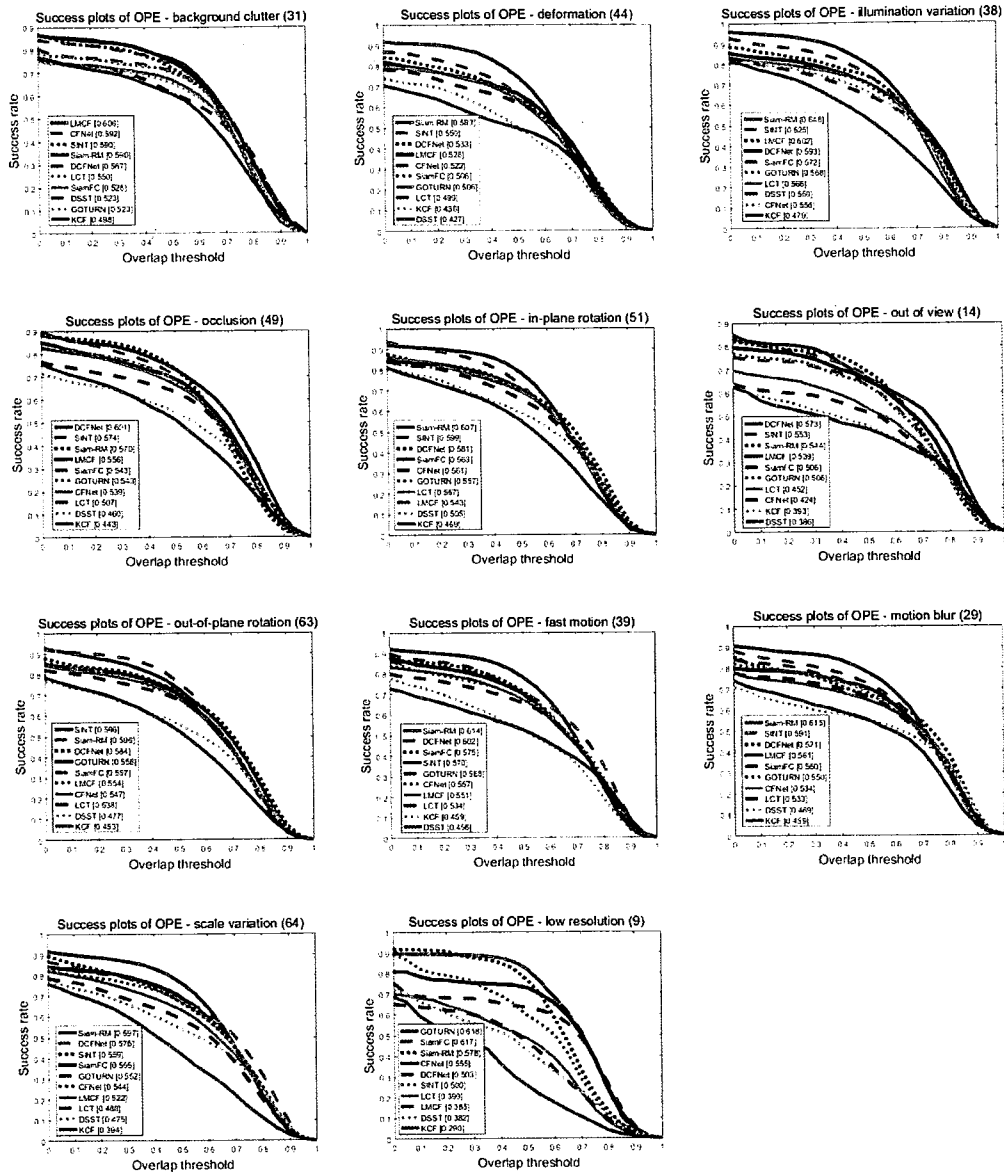


图 3.7 OTB-2015 数据集上 11 种属性下的实验结果（重叠率）

Figure 3.7 Tracking results on 11 challenging attributes on the OTB-2015 dataset (OR)

具体而言，由于 Siam-RM 算法采用了再检测孪生卷积神经网络作为再检测机制，因此对于复杂场景下的背景干扰、快速运动、运动模糊和光照变化等挑战因素具有排名第一的跟踪精度。值得一提的是，虽然再检测机制的运用能够提高算法在背景杂乱属性下的精度，但是其重叠率却较低。这说明单一尺度的再检测网络并不能较好地满足算法高重叠率的要求，需要在网络训练时融入多尺度的训练样本或在推理时融入多尺度的搜索策略，从而在杂乱背景下对相似干扰排除的同时提高算法的重叠率。另外，Siam-RM 算法采用基于生成式模型的自适应模

板更新方法,从而在遮挡、目标形变、目标旋转和尺度变换等属性下具有排名第一的跟踪精度。尤其在平面内旋转下, Siam-RM 算法的跟踪精度达到了 85.9%,同时重叠率也明显优于其它参与比较的跟踪算法。然而,在遮挡、低分辨率、出视场和平面外旋转属性下, Siam-RM 算法虽然具有较高的精度,但是在重叠率上的表现却并不是最优的。这说明自适应模板更新方法能够在一定程度上提高算法的精度,但其并不是复杂场景下保证算法最优性能的关键。所以,如何进一步提升孪生卷积神经网络以获得更加准确的跟踪结果才是解决问题的关键。

(2) 定性分析

为了对算法进行更好的分析,除了定量分析以外还进行了定性分析。图 3.8 展示了一些测试序列上的跟踪结果,这些测试序列代表着不同的复杂场景并具有不同的挑战因素,图像序列从上到下依次为 Deer、Liquor、Jumping、Freeman4、Soccer、Football、Ironman、Matrix、Shaking 和 CarDark。为了能够使结果更加清晰,每个视频只截取了一定数量的片段并显示了 6 个跟踪算法的结果。在截取的图像上,不同颜色的跟踪框代表相应算法的跟踪结果,不同颜色与跟踪算法的对应关系在图例上进行了说明。

1) 快速运动 (FM)

为了在目标快速移动时捕获目标,跟踪算法中不应使用较小的搜索区域。但是,大的搜索区域可能会增加引入背景干扰的风险。在 Deer 和 Liquor 序列中,目标的移动速度很快。目标与背景交叠并改变的非常突然,一些跟踪器有定位错误或停留在先前的位置 (KCF, LCT),但 Siam-RM 能够较为稳定的对目标进行跟踪。

在 Jumping 序列中,目标从第 33 帧开始跳动。在第 39 帧时, LCT、DSST 和 KCF 算法最先丢失了目标。DSST 算法在第 41 帧对目标重新定位,但在第 146 帧再次丢失了目标。虽然, SiamFC 算法采用了余弦窗函数对响应图进行了处理,但在跟踪过程中依然会出现目标丢失的情况。Siam-RM 算法将余弦窗从框架中移除,并采用再检测网络对响应图中的多波峰进行排除。这不仅能使算法在较大的搜索区域内抵抗相似干扰的影响,还能有效提升算法的跟踪精度。



图 3.8 不同测试序列上的实验结果

Figure 3.8 Visualization of the tracking results in different sequences

2) 遮挡 (OCC)

遮挡是目标跟踪过程中比较棘手的难题。如果目标在不具备有效模板策略的情况下被遮挡，跟踪算法会错误地将遮挡视为要跟踪的对象。在 Freeman4 序列中，目标从 157 帧开始几乎被完全遮挡。当目标在 161 帧再次出现时，Siam-RM 和 SiamFC 算法首先定位到了目标。然而，由于 SiamFC 算法并没有对模板进行更新，在第 202 帧时就出现了跟踪丢失的现象。Siam-RM 算法由于采用自适应模板更新方法并结合较为鲁棒的更新策略，其能够一直保持较为稳定的跟踪。KCF 和 DSST 算法在跟踪过程中将遮挡物当作目标，从而出现了跟踪丢失的现象。

在 Soccer 序列中, 目标从第 95 帧开始出现遮挡, 并在第 120 帧时几乎被完全遮挡, 所有的算法都进入到一种随机的状态。在第 144 帧时目标重新出现在视场中, Siam-RM 算法依靠再检测机制重新定位到了目标, 而其它算法由于定位失败无法继续跟踪。在 Football 序列中, 目标从第 115 帧开始被严重遮挡。在第 310 帧时, KCF 和 DSST 算法将目标旁的相似干扰当作了目标, 而 Siam-RM 依靠再检测机制和自适应模板更新实现了稳定的跟踪。

3) 背景杂乱和光照变化 (BC & IV)

背景杂乱是指目标附近背景的颜色或纹理与目标的颜色或纹理相似, 这就需要有一个具有高分辨能力的跟踪算法。由于有效利用了 CNN 特征, Siam-RM、SiamFC 和 CFNet 在区分杂乱背景中的相似干扰方面优于其它跟踪器。在一些不复杂的场景中, 光照变化比背景杂乱更容易克服, 但当与背景杂乱结合时, 跟踪框很容易漂移到目标附近的相似干扰上。

在 Ironman、Matrix、Shaking 和 CarDark 序列中, BC 和 IV 是同时存在的。从图 3.8 中可以看出, 带有相关滤波器的跟踪算法 (LCT、DSST 和 KCF) 在光照变化较大或背景较为杂乱时, 往往无法精确跟踪。相比之下, 利用 CNN 为主要架构的跟踪算法 (Siam-RM、SiamFC、CFNet) 具有更好的跟踪性能。这些算法不仅可以有效地区分目标和背景, 而且可以防止光照变化对特征张量的影响。然而, 由于缺乏有效的再检测机制和可靠的模板更新策略, SiamFC 和 CFNet 的跟踪框往往会向背景上漂移。Siam-RM 利用生成式模型解决了光照变化引起的外观变化问题。该模型使用所有可靠的模板特征做支撑, 同时采用高置信度的模板更新策略, 有效避免了模板更新时的污染问题。此外, 一种高效的再检测机制可以有效地防止杂乱背景的干扰, 从而保持稳定、准确的跟踪效果。

通过以上的定性分析与定量分析可以发现, Siam-RM 算法利用再检测机制和自适应模板更新能够在一定程度上提升算法的跟踪性能。然而, 单一尺度的再检测网络并不能较好地满足算法高重叠率的要求, 需要在网络训练时融入多尺度的训练样本或在推理时融入多尺度的搜索策略, 从而在杂乱背景下对相似干扰排除的同时提高算法的重叠率。自适应模板更新方法能够在一定程度上提高算法的精度和鲁棒性, 但其并不是复杂场景下保证算法最优性能的关键。所以, 如何进

进一步提升孪生卷积神经网络以获得更加准确的跟踪结果才是解决问题的关键。

3.3.2 VOT 数据集实验分析

VOT-2016 包含了 60 个具有多种挑战性干扰因素的测试序列,其中有少部分的测试序列与 OTB 数据集相同,能够对跟踪算法在不同复杂场景下的优劣进行充分评估。VOT 数据集常采用的评价指标主要包含:准确度 (Accuracy, A)、鲁棒性 (Robustness, R) 和预期平均重叠率 (Expected Average Overlap, EAO)。表 3.3 展示出了参与比较的各个算法在 VOT-2016 数据集上的测试结果,其中红、蓝、绿三种颜色分别对应跟踪性能排名第一、第二和第三的算法。

表 3.3 VOT-2016 数据集上不同算法的结果

Table 3.3 Tracking methods results on VOT-2016 dataset

跟踪器	EAO	Accuracy	Failure
TCNN [82]	0.3249	0.547	17.939
ECO-HC [74]	0.2665	0.510	21.402
SSAT [123]	0.3207	0.570	19.272
MLDF [123]	0.3106	0.487	15.044
DDC [123]	0.2929	0.534	20.981
SiamRN [123]	0.2766	0.549	23.996
MDNet_N [78]	0.2572	0.537	21.082
SiamAN [123]	0.2352	0.531	29.802
Staple [143]	0.2952	0.543	23.895
C-COT [72]	0.3310	0.533	16.582
TRACA [144]	0.1599	0.458	37.947
DCFNet [142]	0.2071	0.489	24.937
SiamFC [90]	0.2039	0.515	24.668
EBT [145]	0.2913	0.453	15.194
Siam-RM	0.3078	0.498	14.511

从表中可以看出, Siam-RM 在 EAO 指标的表现上排名第五,在准确度指标的表现上排名较差,在平均失败次数的表现上排名第一。这充分说明, Siam-RM 中再检测机制和自适应模板更新的应用能够在一定程度上提升算法的鲁棒性和准确性,特别是在鲁棒性方面的提升较为突出。然而,单纯的融合这两个模块并不能使精度大幅度提升,如何对孪生卷积神经网络进一步改进以获得更加准确的跟踪结果才是解决问题的关键。为了更加清晰地对比各算法的跟踪性能,在图

3.9 中对各算法的 EAO 排序进行了展示。

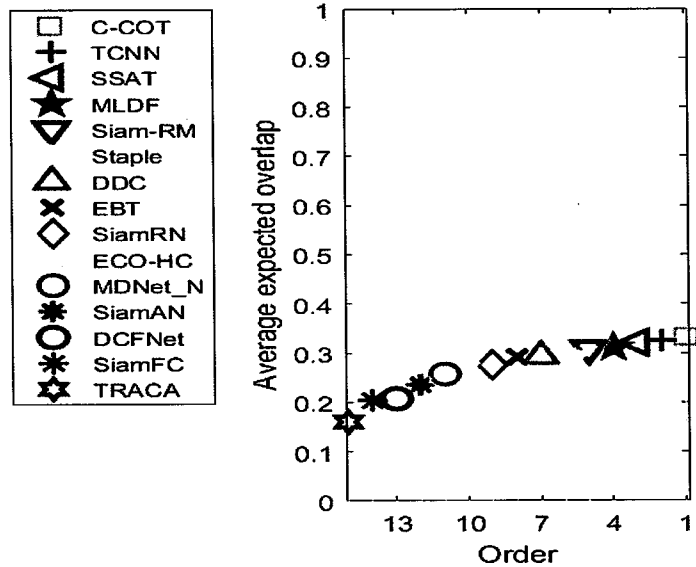


图 3.9 不同跟踪算法在 VOT-2016 数据集上的 EAO 结果

Figure 3.9 The EAO scores of different tracking algorithms on VOT-2016 dataset

3.3.3 消融实验

为了对本章提出的 Siam-RM 算法中再检测机制和自适应模板更新的有效性进行分析,本小节利用两个模块的增减组合构建出不同的跟踪算法,从而能够在 OTB-2015 数据集上进行实验对比。其中,构建的算法主要分为:有再检测机制和有自适应模板更新模块的 Siam-RM (Siam-RM_WRM+WU),有再检测机制和无自适应模板更新模块的 Siam-RM (Siam-RM_WRM+OU),无再检测机制和有自适应模板更新模块的 Siam-RM (Siam-RM_ORM+WU),无再检测机制和无自适应模板更新模块的 Siam-RM (Siam-RM_ORM+OU)。同时,在对比实验中还将 SiamFC 作为实验的基准算法,以便更加直观的看出各个构建算法的性能优劣。消融实验结果的精度曲线和成功曲线如图 3.10 所示。

从图中可以看出,带有再检测机制的 Siam-RM 算法无论是否搭配自适应模板更新模块都具有高于 SiamFC 算法的精度。其中,两个模块全部具备的 Siam-RM_WRM+WU 算法具有最高的精度。当 Siam-RM 算法去除再检测机制时,跟踪精度就会明显下降。Siam-RM 算法用于跟踪的孪生卷积神经网络虽然是在 SiamFC 的网络结构上进行改进的,但其中的一些优化策略用再检测机制加以代

替，如余弦窗函数等。所以，没有再检测机制和自适应模板更新的 Siam-RM 算法要比 SiamFC 算法具有更低的精度。然而，如果将自适应模板更新加入到 Siam-RM 中，会明显提升无再检测机制的 Siam-RM 算法的性能，充分证明自适应模板更新方法能够在一定程度上明显提升算法的精度和鲁棒性。

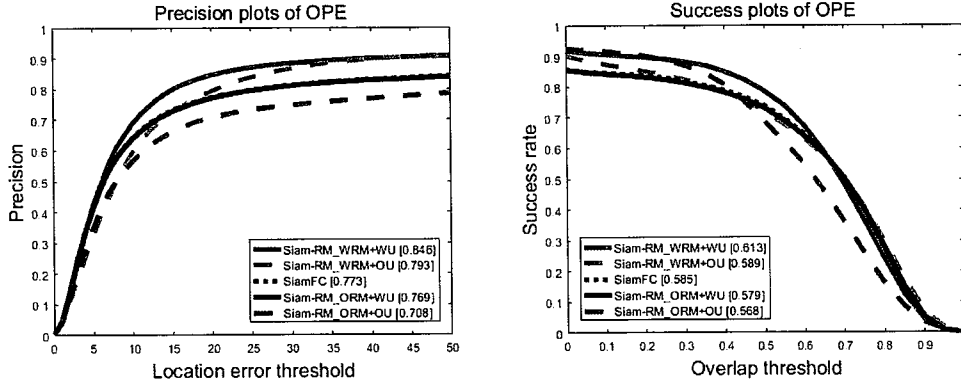


图 3.10 OTB-2015 数据集上的消融实验结果

Figure 3.10 The results of the ablation study on OTB-2015 dataset

3.4 本章小结

本章提出一种结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。该算法以 SiamFC 算法为基础，利用一种高效且准确的再检测孪生卷积神经网络作为再检测机制来应对响应图中的多波峰。另外，本章算法提出一种基于生成式模型的自适应模板更新方法。该模板更新方法将评估过的可靠跟踪结果进行特征提取与概率统计，利用统计结果与已有目标模板特征进行自适应融合，从而实现模板的自适应更新。仿真实验结果表明，本章算法能较好应对相似背景干扰、遮挡、目标形变等难题，有效提升复杂场景下的跟踪精度与鲁棒性。

本章主要研究内容已经整理成文，并发表学术论文一篇：

Daqun Li, Yi Yu, Xialin Chen. Object tracking framework with Siamese network and re-detection mechanism [J]. EURASIP Journal on Wireless Communications and Networking, 2019.

第4章 结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法

4.1 引言

第三章提出的结合再检测机制与自适应模板更新的目标跟踪算法,有效地利用了浅层卷积神经网络和离线训练的优势,能够在一些低复杂度的场景取得不错的跟踪效果。但是,由于缺少丰富的语义特征,第三章提出的算法在应对更为复杂的场景时跟踪效果并不理想。这充分说明,采用浅层网络结构会明显影响基于深度特征匹配的目标跟踪算法的跟踪性能。一方面,浅层网络提取的特征具有较少的语义信息,虽然在一些低复杂度的场景下具有较高的精度,但在复杂场景下的表现会明显下降。另一方面,如果直接使用 VGG 等深层网络,边缘填充、网络步长较大和神经元感受野的大幅度增加,会直接影响目标特征的可辨识性及定位精度,不利于目标的精确定位。并且,深层网络的引入在增加计算量的同时会打破全卷积孪生网络结构严格的平移不变性。虽然, Li 等提出的 SiamRPN++^[97] 算法成功克服了引入深层网络带来的问题,但多层间的卷积特征并没有采用合理的方式进行融合。

为了克服上述问题,本章提出一种结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。该算法以 SiamFC 算法为基础,利用一种改进的残差网络来替代传统的浅层网络,从而提取出具有丰富语义信息的深层特征。与此同时,本章算法提出一种高效的自适应多层特征融合策略,有效提升了响应图的质量,减少了在复杂场景下相似背景干扰所带来的影响。另外,本章算法利用一种自适应特征信息融合方法来实现模板的更新,该方法在第三章提出的自适应模板更新方法基础上进行优化与改进,从而在减少计算量的同时有效克服遮挡、目标形变等难题,提升了算法的跟踪精度与鲁棒性。

4.2 结合深层特征与鲁棒特征融合的孪生卷积神经网络

在本节中,详细介绍本章所提出的结合深层特征与鲁棒特征融合的孪生卷积

神经网络目标跟踪算法，并将其简称为SiamDF (Siamese Visual Tracking with Deep Features and Robust Feature Fusion)。首先，本节将对孪生卷积神经网络中改进结构的残差网络进行介绍。接着，本节对高效的自适应多层特征融合策略进行阐述。最后，本节对基于自适应特征信息融合的模板更新方法进行论述。SiamDF跟踪算法的框架如图4.1所示。

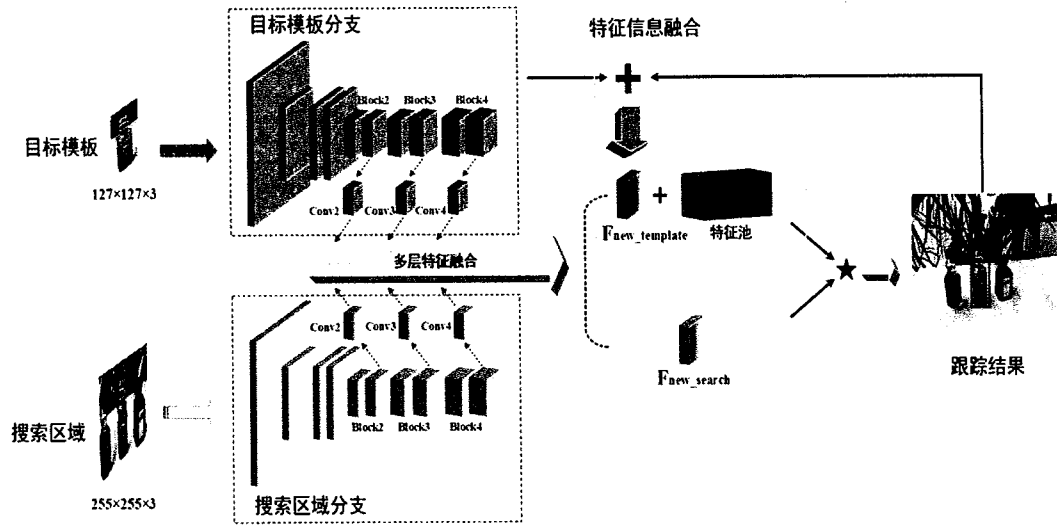


图 4.1 SiamDF 算法的结构框架

Figure 4.1 The framework of the SiamDF algorithm

4.2.1 改进结构的残差网络

大部分基于深度特征匹配的目标跟踪算法采用的网络结构都是浅层网络，无法充分利用深层神经网络带来的优势。一方面，浅层网络提取的特征具有较少的语义信息，虽然在一些低复杂度的场景下具有较高的精度，但在复杂场景下的表现会明显下降。另一方面，如果直接使用 VGG 等深层网络，边缘填充、网络步长较大和神经元感受野的大幅度增加，会直接影响目标特征的可辨识性及定位精度，不利于目标的精确定位。并且，深层网络的引入在增加计算量的同时会打破全卷积孪生网络结构严格的平移不变性。为了更好地利用深层网络来提升算法的性能，SiamDF 算法采用改进的 ResNet-18 来代替浅层的网络结构。改进结构的 ResNet-18 网络结构如表 4.1 所示。

表 4.1 SiamDF 中改进的残差网络结构

Table 4.1 The improved structure of the ResNet-18 in Siam-RM

输出尺寸			
层名称	目标模板	搜索区域	层结构
Conv1	62×62	126×126	7×7, 64,2,1
Maxpool	31×31	63×63	3×3,2,1
Block1	31×31	63×63	$\begin{pmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{pmatrix} \times 2, 1, 1$
Block2	16×16	32×32	$\begin{pmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{pmatrix} \times 2, 2, 1$
Conv2	16×16	32×32	1×1,128,1,0
Block3	16×16	32×32	$\begin{pmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{pmatrix} \times 2, 1, 1$
Conv3	16×16	32×32	1×1,128,1,0
Block4	16×16	32×32	$\begin{pmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{pmatrix} \times 2, 1, 1$
Conv4	16×16	32×32	1×1,128,1,0

在上表中，卷积层和卷积块（Conv*和 Block*）的层结构是由核尺寸、通道数、步长和填充数组成的。最大池化层的层结构是由过滤器尺寸、步长和填充数组成的。从表中可以看出，我们主要对 ResNet-18 网络进行了以下的改进：

1) 将 Block3 和 Block4 中的步长调整为 1。与分类任务不同，网络中的步长增大会使得定位精度下降，从而影响算法的跟踪精度。并且，后续的多层特征融合策略需要三个卷积块输出的卷积特征具有统一的尺寸。因此，将步长调整为 1 能够方便将多层的卷积特征融合为一个特定的特征张量，从而在交叉相关计算后得到最后的响应图。

2) 在网络结构中引入了 3 个卷积核尺寸为 1×1 的卷积层（Conv2、Conv3 和 Conv4）。这三个卷积层是多层特征融合的基础，主要用来对卷积块

Block2-Block4 输出的卷积特征进行调整，从而实现后续的多层特征融合。

另外，为了打破全卷积孪生网络结构严格的平移不变性，在训练改进的残差网络的过程中借鉴了 SiamRPN++^[97]算法中提出的空间感知采样策略。因此，在进行样本采集时该网络不会仅给图像中心赋予较大的权值，并能有效地缓解填充操作对平移不变性造成的破坏。在训练时，SiamDF 算法会采用与 SiamFC 相同的损失函数（公式 2.17），并利用 SGD 进行网络优化。训练用的样本对是从标注好的数据集中获取的。训练时，算法先从视频序列中选取一帧作为模板图像，并以模板图像中标注的目标为中心进行目标模板的截取。接着，算法以模板图像为中心，在其前后一定范围内进行搜索图像的选取并截取相应的搜索区域。在确定搜索区域后，会利用空间感知采样策略对其在 X 和 Y 方向上随机进行 ± 32 像素范围内的移动，并对移动的像素数进行记录以便于损失函数的计算。最后，将截取好的目标模板与搜索区域组成一组样本对用于网络模型的训练。具体的训练细节可参考 2.3.2 小节或 SiamFC 原文。

4.2.2 多层特征融合

在 CNN 网络模型中，网络提取的卷积特征是具有层级性的。在网络浅层通常会获取边缘、颜色、纹理等简单的外观特征，随着网络层数的加深，富含更多语义信息的高层次抽象特征会不断被提取。这些特征不仅具有较高的识别性，还具有一定的平移不变性等优良特性，从而极大地提高了算法的鲁棒性与准确性。Zeiler 等人^[146]在 2013 年利用反卷积方法对 AlexNet 网络模型^[114]提取的特征进行了可视化，并发现其提取的特征是具有一定层级性的。Kong 等人^[147]成功引入一个名为 HyperNet 的网络用于融合最后一层和中间两层的卷积特征，并将设计的网络应用于目标检测算法中，使得算法的精度和召回率都得到明显的提升。Long 等人^[148]利用 FCN 提取具有丰富语义信息的卷积特征，并利用上采样融合方法对多层特征进行融合，从而实现了像素级别的图像分割。Ma 等人^[71]发现浅层特征具有更多的空间信息，有利于精确定位。此外，深度特征包含更丰富的语义信息，对复杂场景下的遮挡、目标形变等干扰因素具有比浅层特征更好的不变性。因此，Ma 等人将 VGGNet^[116]中三种不同的卷积层（Conv3-4, Conv4-4, Conv5-4）提取的特征分别输入到三个对应的相关滤波器中，最后将滤波器输出的三个权重图进行

结合，从而得到目标的精确位置。

受上述讨论的启发，SiamDF 算法采用多层特征融合的方式对从 Block2、Block3 和 Block4 中提取的特征张量进行融合，从而获得高质量的响应图，减少在复杂场景下相似背景干扰等因素带来的影响。从图 4.1 和表 4.1 中可以看出，3 个卷积核尺寸为 1×1 的卷积层（Conv2、Conv3 和 Conv4）被引入到了网络结构的两个分支中。这些卷积层分别用于处理卷积块 Block2、Block3 和 Block4 的输出。为了减少交叉相关运算的负担，将每个卷积层的输出通道调整为 128。与此同时，对各个卷积块的输出裁剪大小以保留中心区域（ 8×8 和 24×24 ），这样不仅可以保留输入的特征，还能在一定程度上减少运算量。

在融合阶段，需要对模板分支和搜索区域分支中的 Conv2、Conv3 和 Conv4 卷积层的输出进行交叉相关计算。由此，我们可以分别得到对应于 Conv2、Conv3 和 Conv4 的三个响应图。同时，可以利用平均峰值相关能量（APCE）（式 3.6）对每个响应图进行计算。在对各个响应图进行 APCE 计算之后，可以利用下式对模板分支及搜索区域分支中 Conv2、Conv3 和 Conv4 提取的特征张量进行融合，并得到融合后的特征 $F_{new_template}$ 与 F_{new_search} ：

$$F_{new_template}, F_{new_search} = \sum_{i=2}^4 \alpha_i * OutConv_i \quad (4.1)$$

其中， $OutConv_i$ 代表对应的分支中 Conv (i) 提取的特征张量， $F_{new_template}$ 与 F_{new_search} 分别代表模板分支及搜索区域分支中最后融合得到的特征， α_i 代表融合过程中各特征张量对应的归一化权重，其可以利用下式进行计算：

$$\alpha_i = \frac{APCE_i}{\sum_{i=2}^4 APCE_i} \quad (4.2)$$

通过表 4.1 与公式 4.1 可以发现，利用上述多层特征融合方法得到的各分支融合特征 $F_{new_template}$ 与 F_{new_search} 具有相同的通道数。由此，可以利用融合后的特征进行最后的交叉相关计算，从而得到最终的响应图。

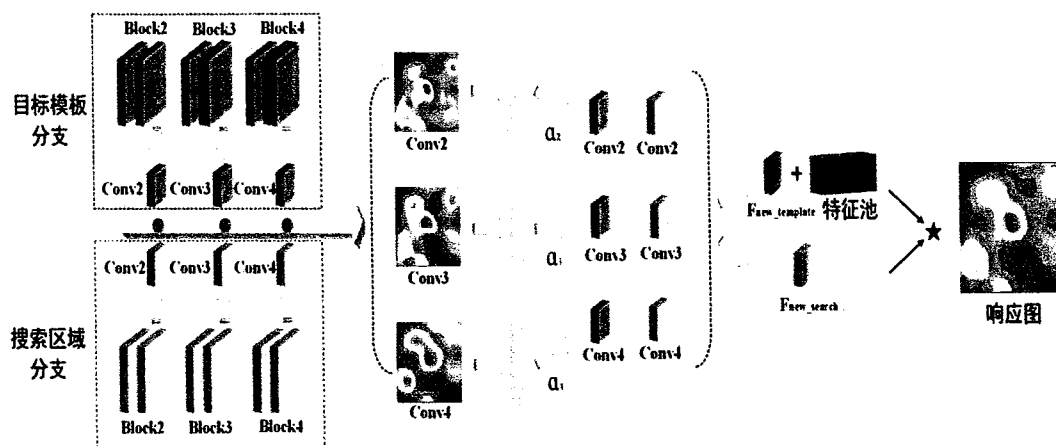


图 4.2 多层特征融合示意图

Figure 4.2 The diagram of the multi-layer feature fusion

多层特征融合示意图如图 4.2 所示。在对 Conv2、Conv3 和 Conv4 三个卷积层中输出的卷积特征融合后，SiamDF 算法可以利用融合后的特征进行交叉相关计算并得到最终质量较高的响应图。从图中可以看出，与 Conv2、Conv3 和 Conv4 三个卷积层分别对应的响应图并不具备较高的质量。在三个响应图中，不仅存在着许多次级波峰的干扰，主波峰的显著度也并不高。如果不对其进行处理，响应图中的干扰信息会对跟踪性能产生很大的影响，有时甚至会导致跟踪目标的丢失。然而，当经过多层特征融合处理之后，最终响应图的质量得到明显改善，响应图中的干扰信息得到明显抑制。因此，SiamDF 采用的多层特征融合策略可以显著降低干扰信息的影响，提高算法在复杂场景下的准确性与鲁棒性。图中的特征池代表所有可靠跟踪结果的可信特征集合，将在下一小节进行详细介绍。

4.2.3 基于特征信息融合的模板更新

在大多数基于深度特征匹配的目标跟踪算法中，常采用固定的模板去搜索目标。这种固定模板的方式不仅会使跟踪算法无法适应目标外观的各种变化，还会因为复杂背景下各种干扰因素的影响而导致跟踪失败。因此，采用一种自适应的模板更新策略显得尤为重要。

在第三章中提出的 Siam-RM 算法采用的是一种基于生成式模型的模板更新方法。该方法虽然在一定程度上提高了算法在复杂场景下的准确性与鲁棒性，但其计算量较大，对于特征张量的建模较为复杂。因此，为了在保证同等效力的同

时对其进行简化, SiamDF 算法采用一种基于特征信息融合的模板更新方式进行模板的更新。在初始化阶段, 算法建立一个特征池来存储可靠跟踪结果的可信特征, 这些可靠跟踪结果是通过对各帧的跟踪结果进行评估后提取出来的。在 SiamDF 算法中, 仍然使用 APCE 值对跟踪结果进行评估。如果与跟踪结果对应的最终响应图的 APCE 值大于某一阈值 T_{APCE} , 则将该跟踪结果放入模板分支(参数与当前帧相同), 并将提取的特征张量作为可信特征。当可信特征数小于上限 N 时, 融合过程可表示为:

$$F_{final} = \sum_{k=1}^n \omega_k F_k + \eta * F_{new_result} \quad n \leq N \quad (4.3)$$

$$\begin{cases} \omega_1 = (1-\eta)^C & k=1 \\ \omega_k = (1-\eta)^{C-k+1} * \eta & 1 < k \leq n \end{cases} \quad (4.4)$$

式中, η 代表学习率, ω_k 代表第 k 个特征的权重, n 代表特征池里现存的特征数量, F_k 代表特征池里的第 k 个特征, C 代表融合更新的次数, F_{new_result} 代表可靠跟踪结果在模板分支进行多层特征融合后输出的可信特征张量。因此, 融合得到的 F_{final} 便可作为更新后的模板特征, 实现了模板的自适应更新。与此同时, F_{new_result} 会被输入到特征池中进行存储。

与第三章中的自适应模板更新方法类似, 当特征池存储的特征达到上限时, 需要对特征池进行更新。在这里, 借鉴 3.2.3 小节中对生成式模型更新的方式, 假设新得到的可信特征张量 F_{new_result} 的权重和均值分别为 ω_{N+1} 和 μ_{n+1} , 则需要对三种情况进行考虑:

- 1) 特征池中的一个可信特征权重过低。在这种情况下, 会通过设置一个阈值 ω_{min} 对各个可信特征进行筛选。如果特征池中的一个可信特征的权重低于 ω_{min} , 则会用新得到的可信特征 F_{new_result} 替代权重过低的特征;
- 2) 新的可信特征与特征池里各特征间的欧式距离小于特征池中任意两个特征间的欧式距离。在这种情况下, 新的可信特征 F_{new_result} 会和与其距离最小的特征融合;
- 3) 新的可信特征与特征池里各特征间的欧式距离大于特征池中任意两个特

征间的欧式距离。在这种情况下，会将模型中距离最近的两个特征融合，并用新的可信特征 F_{new_result} 填补空缺。

在上述情况中，算法会采用下式进行组件的融合：

$$\omega_{update} = \omega_a + \omega_b \quad (4.5)$$

$$F_{update} = \frac{\omega_a F_a + \omega_b F_b}{\omega_a + \omega_b} \quad (4.6)$$

式中， a 和 b 代表需要融合的两个特征。由此，特征池便能实现自适应的更新并保持一定的稳定性与鲁棒性。此外，当特征池达到上限 N 后，特征信息融合过程可表示为：

$$F_{final} = (1-\eta) * \sum_{k=1}^n \omega_k F_k + \eta * F_{new_result} \quad n > N \quad (4.7)$$

其中， ω_k 采用式 4.5 进行更新。

4.2.4 算法描述

在跟踪过程中，首先会给定第一帧中的目标信息，SiamDF 算法根据该信息进行算法初始化，如目标模板截取、模板特征提取与多层特征融合、搜索尺度确立等。接着，SiamDF 算法会在下一帧截取搜索区域，并将搜索区域图像输入到改进结构的残差网络中进行特征提取与多层特征融合，利用交叉相关计算获取响应图，从而利用该响应图确定最后的跟踪结果。然后，SiamDF 算法会对响应图进行质量评价，计算其 APCE 值是否高于阈值 T_{APCE} 。如果高于阈值，则会将跟踪结果输入到模板分支中进行可信特征的提取。最后，SiamDF 跟踪算法会结合得到的可信特征与特征池进行特征信息融合，从而实现模板的自适应更新。

SiamDF 算法的运行流程如下表所示：

表 4.2 SiamDF 算法运行流程

Table 4.2 The execution process of SiamDF

结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法 (SiamDF)
输入: 预训练好的网络, 初始帧目标信息
Step 1: 利用初始帧目标信息进行算法初始化 (目标模板截取、模板特征提取与多层特征融合、搜索尺度确立等)
对于图像序列帧数 $frame=2,3,\dots,N$ 依次循环 Setp2-5
Step 2: 在第 t 帧图像上的 (x_{t-1}, y_{t-1}) 周围确定搜索区域, 利用改进结构的残差网络提取特征张量, 利用式 (4.1) 和式 (4.2) 进行多层特征融合, 最后利用交叉相关计算获取响应图, 并确定跟踪结果 (x_t, y_t) 和 (w_t, h_t) 。
Step 3: 通过式 (3.6) 对当前跟踪结果对应的响应图进行质量评价。如果高于阈值 T_{APCE} , 将新跟踪到的结果输入到目标模板分支中提取可信特征
Step 4: 利用式 (4.3)、式 (4.4) 和式 (4.7) 进行特征信息融合, 以实现模板特征的更新
Step 5: 利用式 (4.5) 和式 (4.6) 对特征池进行更新
输出: 当前帧目标的位置 (x_t, y_t) 和大小 (w_t, h_t)

4.3 实验结果与分析

本章算法在 PyCharm 下利用 pytorch-0.4.1 深度学习工具集实现。用于实验的 PC 机配置为: Intel i7-9800X CPU (3.80GHz), 64GB RAM 和一块 NVIDIA TITAN RTX GPU。算法平均运行速度为 32 帧每秒。SiamDF 算法中改进结构的残差网络是在线训练。在训练改进结构的残差网络时, 采用 ILSVRC-2015 的 VID^[118]和 COCO 数据集^[149]进行离线训练, 且网络中每层参数的初始值遵循高斯分布。采用 SGD 方法对网络进行优化, 设置权重衰减 (Weight Decay) 为 0.0005。整个训练过程为 50 个迭代周期, 并设置初始学习率为 0.001。在每个迭代周期结束后会乘以一个固定的常数, 直到最后一个迭代周期的学习率降为 0.00001。与此同时, 训练过程中的动量值设置为 0.9, 每个小批量 (mini-batch) 包含 64 个样本。另外, SiamDF 算法中用于特征信息融合的参数是通过大量实验验证得到的, 学习率 η 设为 0.4, 特征池存储可信特征的上限 N 设为 40, 特征池更新时的阈值 ω_{min} 设为 0.0036, 响应图质量评价时的阈值 T_{APCE} 设为 8。

为了客观地对 SiamDF 算法的性能进行评价，本小节采用 OTB 数据集和 VOT-2016 数据集对 SiamDF 及其它一些先进的算法进行对比，从而验证本章算法的准确性、鲁棒性及有效性。

(1) OTB 数据集

OTB 数据集中的测试序列包含了完整的标注信息和种类繁多的跟踪目标。该数据集主要包含两个版本：一个版本是由 50 个测试序列组成的 OTB-2013，另一个版本是由 100 个测试序列组成的 OTB-2015，是 OTB-2013 的扩展集并简称为 OTB-100。本小节的实验采用以上两个数据集，并利用 2.4.1 小节中介绍的评价指标对算法进行评估。其中，参与对比实验的算法包括 SINT^[86]、SiamFC^[90]、GOTURN^[87]、CFNet^[91]、DCFNet^[141]、SiamRPN^[95]、ECO_HC^[74]、TRACA^[144]和 RFL^[150]算法。与此同时，实验采用精度曲线（Precision Plot）和成功曲线（Success Plot）进行直观的比较，并采用 OPE 评价方法对算法进行评估。

(2) VOT 数据集

实验采用的 VOT 数据集是最常用的 VOT-2016 数据集^[123]，该数据集包含了 60 个具有多种挑战性干扰因素的测试序列，其中有少部分的测试序列与 OTB 数据集相同。这些测试序列中包含了动物、人脸、玩具和交通工具等常见的目标，能够对跟踪算法在不同复杂场景下的优劣进行充分评估。在 VOT-2016 数据集上，开展了与其它 14 种跟踪算法的实验对比，包括 SiamRPN^[95]、MDNet_N^[78]、C-COT^[72]等，并利用 2.4.2 小节中介绍的评价指标对算法进行评估。

4.3.1 OTB 数据集实验分析

(1) 定量分析

图 4.3 和图 4.4 分别对 OTB-2013 上和 OTB-2015 上的实验结果进行了展示。从图中可以看出，SiamDF 算法在 OTB-2013 数据集上具有排名第三的精度与排名第一的重叠率。在精度方面，SiamDF 与排名第二的 SiamRPN 仅相差 0.1%，比 SiamFC 跟踪算法在精度上高出 6.8%。在重叠率方面，SiamDF 比排名第二的 SiamRPN 高出 0.3%，比 SiamFC 算法高出 4.9%。SiamDF 算法在 OTB-2015 数据集上排名第一，比 SiamRPN 跟踪算法在精度和重叠率上高 1%和 1.9%。比 SiamFC 算法在精度和重叠率上高 8.8%和 7.1%。上述实验结果充分证明，SiamDF

算法能够利用改进的残差网络、多层特征融合和特征信息融合有效提升复杂场景下算法的跟踪精度及准确度。

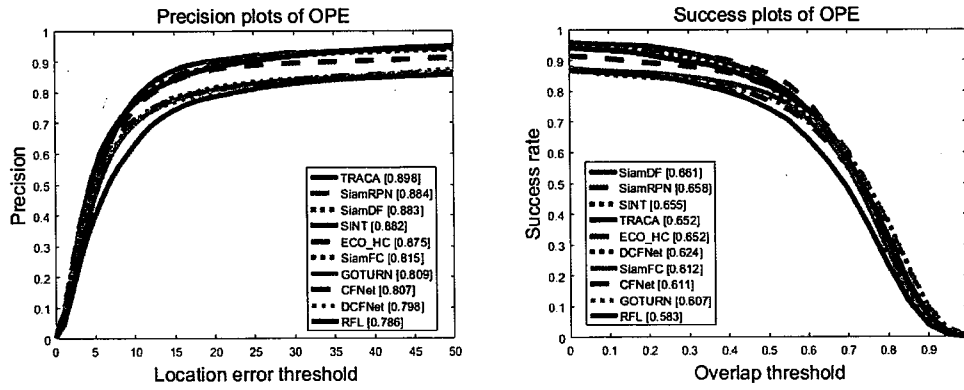


图 4.3 OTB-2013 数据集上的实验结果

Figure 4.3 The tracking results on OTB-2013 dataset

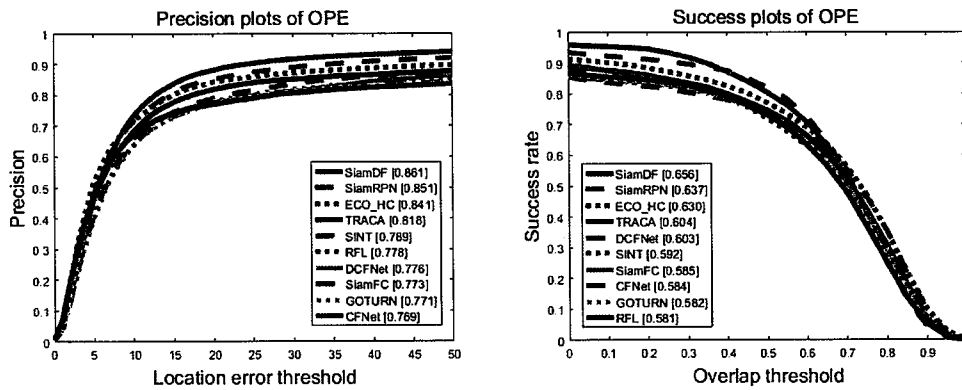


图 4.4 OTB-2015 数据集上的实验结果

Figure 4.4 The tracking results on OTB-2015 dataset

与其它基于深度特征匹配的跟踪算法相比，SiamDF 的优越性更加明显。SiamDF 利用改进的 ResNet-18 网络来代替浅层网络。深层网络提取的特征不仅对光照变化和背景杂波的敏感度较低，而且丰富的语义信息也能大大提高跟踪算法的识别能力。同时，参与比较的算法中使用的模板仅在第一帧中获取，并在跟踪过程中不再更新。这种固定模板的方式不仅会使跟踪算法无法适应目标外观的各种变化，还会因为复杂背景下各种干扰因素的影响而导致跟踪失败。SiamDF 采用自适应的特征信息融合来更新模板，使算法克服了这一限制并获得了良好的跟踪性能。

与基于相关滤波器的跟踪算法（ECO_HC 和 TRACA）相比，SiamDF 算法

仍然具有较好的性能。SiamDF 算法充分利用不同网络层级中的卷积特征进行自适应融合，从而能够得到高质量的响应图。这无疑是在复杂场景下与基于相关滤波器的跟踪算法竞争的关键。

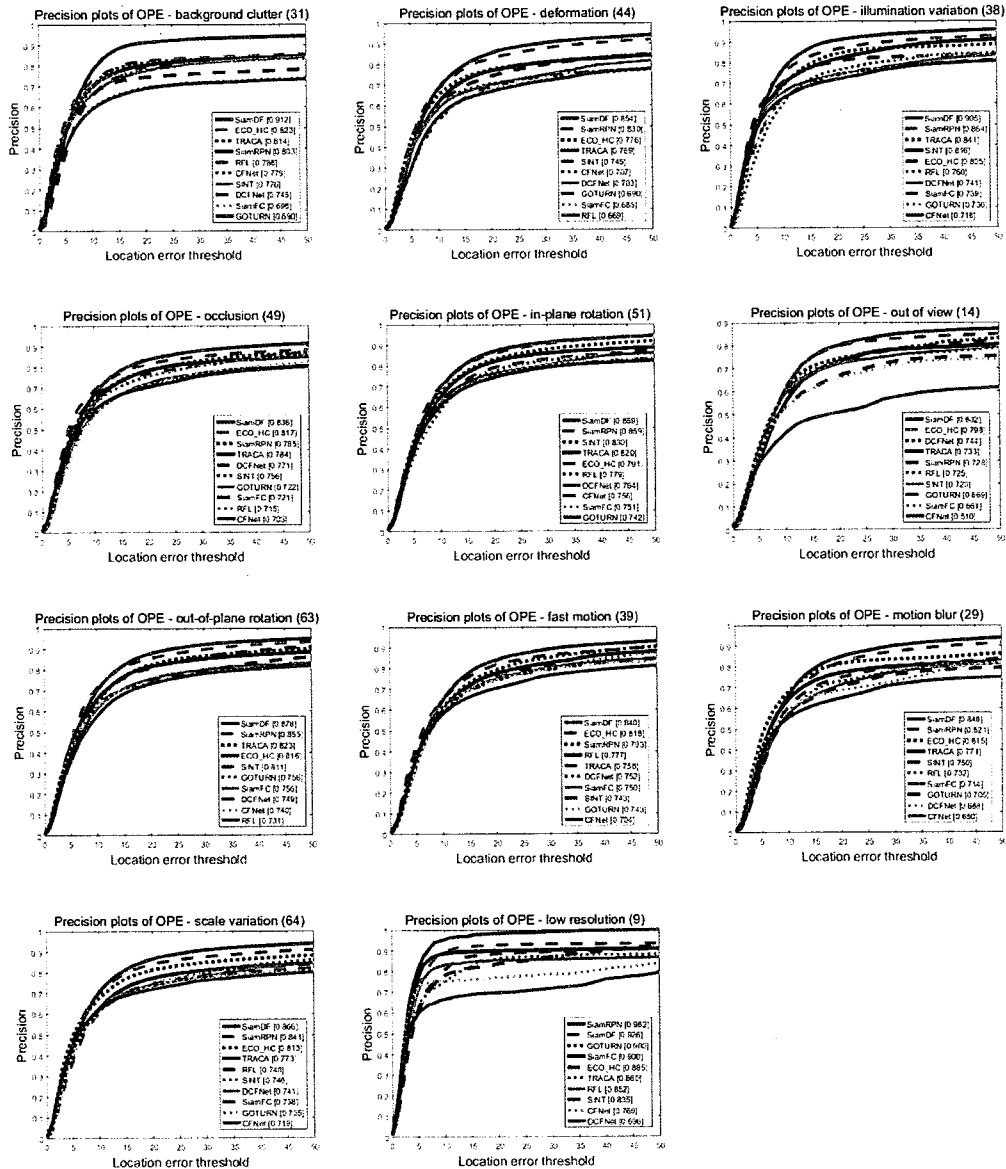


图 4.5 OTB-2015 数据集上 11 种属性下的实验结果（精度）

Figure 4.5 Tracking results on 11 challenging attributes on the OTB-2015 dataset (Precision)

为了进一步进行比较，图 4.5 和图 4.6 展示了各算法在 OTB-2015 数据集上 11 种不同属性（见 2.4.1 小节）下精度和重叠率的比较。

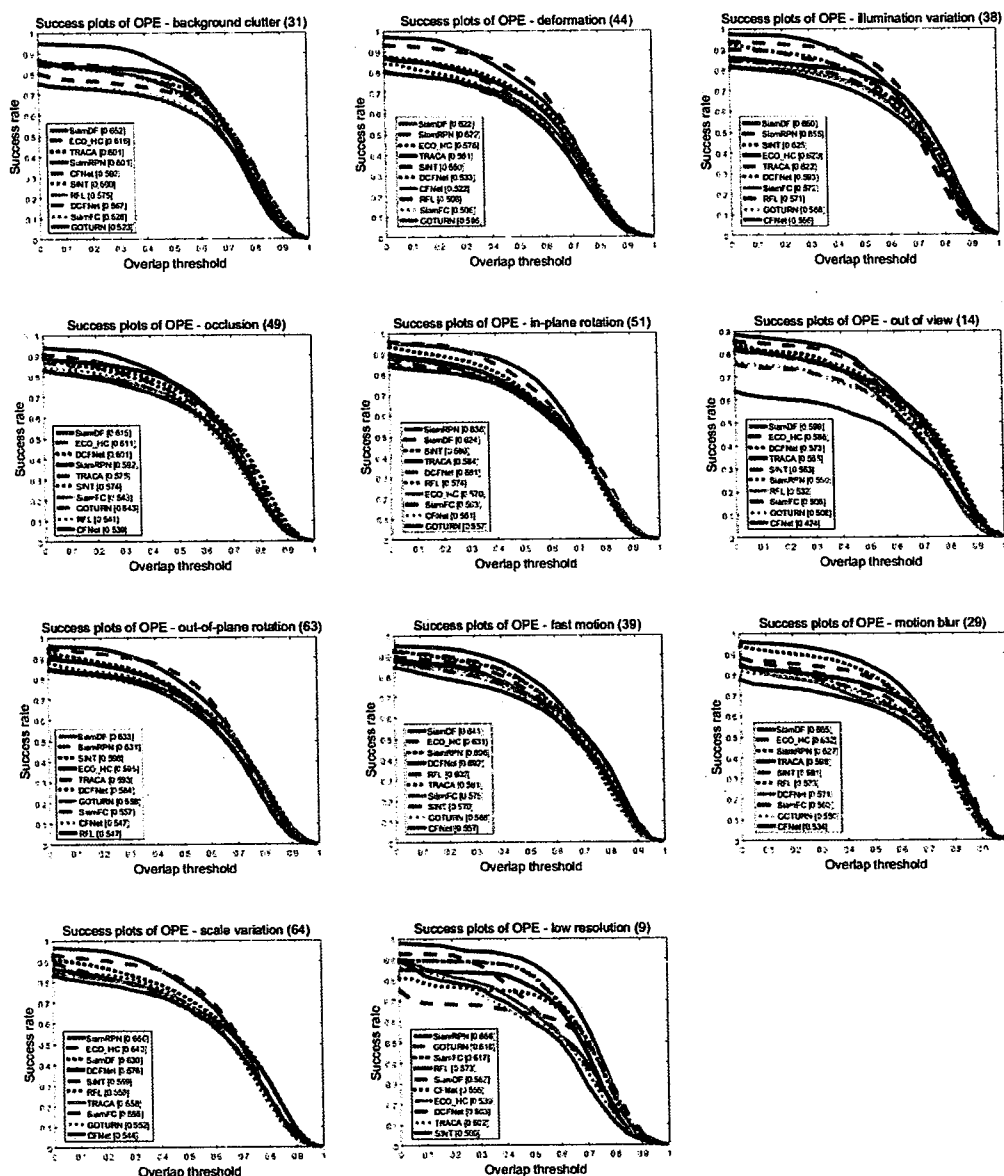


图4.6 OTB-2015数据集上11种属性下的实验结果（重叠率）

Figure 4.6 Tracking results on 11 challenging attributes on the OTB-2015 dataset (OR)

具体而言，由于 SiamDF 算法利用改进的 ResNet-18 网络来代替浅层网络，并采用多层特征融合策略对不同网络层级中的卷积特征进行自适应融合，因此对于复杂场景下的背景干扰、快速运动、运动模糊和光照变化等挑战因素具有排名第一的跟踪精度。值得一提的是，虽然 SiamDF 算法在尺度变化下的精度较好，但是其重叠率却较低。这说明采用深层网络和特征融合虽然能在一定程度上提升精度，但采用有限的尺度变换因子并不能较好地满足算法高重叠率的要求，需要在算法中融入更准确、更鲁棒的多尺度计算策略，从而在复杂背景下更好地应对

尺度变化带来的影响。另外，SiamDF 算法采用基于特征信息融合的自适应模板更新方法，从而在遮挡、目标形变和目标旋转属性下具有排名第一的跟踪精度。然而，值得注意的是，SiamDF 算法在低分辨率属性下的表现并不好，尤其是在重叠率方面。这说明深层网络的使用一方面能够提升算法的精度和鲁棒性，另一方面也会在提取特征的过程中损失一些目标信息，尤其是在目标具有低分辨率（不超过 400 个像素）特性时。所以，如何进一步对深层网络进行改进与提升以获得更加准确的跟踪结果才是解决问题的关键。

（2）定性分析

为了对算法进行更好的分析，除了定量分析以外还进行了定性分析。图 4.7 展示了一些测试序列上的跟踪结果，这些测试序列代表着不同的复杂场景并具有不同的挑战因素，图像序列从上到下依次为 Freeman4、Football、Ironman、Bolt2、Skating1 和 Basketball。为了能够使结果更加清晰，每个视频只截取了一定数量的片段并显示了 6 个跟踪算法的结果。在截取的图像上，不同颜色的跟踪框代表相应算法的跟踪结果，不同颜色与跟踪算法的对应关系在图例上进行了说明。下面将对在一些典型测试序列上的跟踪结果进行简要分析。

1) Freeman4 序列

该测试顺序主要是一个人在教室里自由走动，其主要挑战是遮挡。在第 42 帧时，跟踪目标几乎被完全遮挡，SiamFC 算法直接丢失目标，SiamRPN 算法和 TRACA 算法也在随后被遮挡物混淆并造成跟踪失败。然而，由于加入了自适应特征信息融合来更新模板，SiamDF 能够在整个跟踪过程中稳定地定位目标。

2) Football 序列

该测试序列主要展现了一场激烈的足球比赛，其主要挑战是遮挡和相似目标干扰。在第 107 帧时，SiamFC 算法丢失了跟踪目标，其它基于深度特征匹配的算法由于采用固定不变的模板也发生了跟踪框漂移的情况。虽然一些跟踪算法可以重新检测到目标，但跟踪性能已经受到影响。在整个跟踪过程中，SiamDF 的跟踪性能要优于其它跟踪算法。具有丰富语义信息的深层特征可以使 SiamDF 算法对相似背景干扰的敏感度降低，多层特征融合和基于特征信息融合的模板更新

方法也是获得稳定跟踪结果的关键。

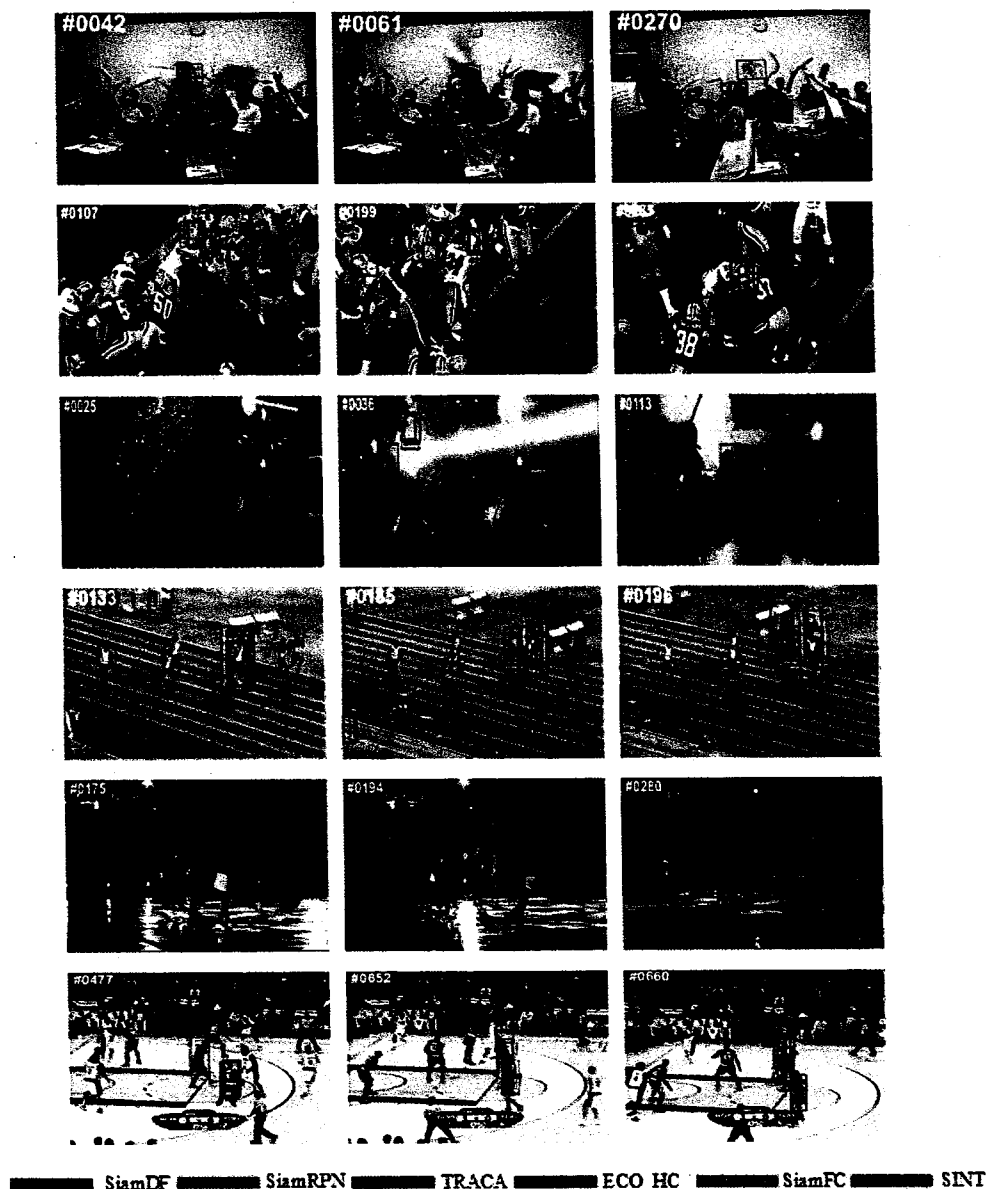


图 4.7 不同测试序列上的实验结果

Figure 4.7 Visualization of the tracking results in different sequences

3) Bolt2 序列

该测试序列主要展示了一场竞跑比赛，其主要的挑战是形变和相似背景干扰。在第 133 帧时，SINT 跟踪算法丢失跟踪目标并在后续过程中无法重新定位目标。其它的跟踪算法从第 185 帧开始发生了跟踪框漂移，ECO_HC 算法和 TRACA 算法甚至跟踪到了背景中的干扰物上。与这些跟踪算法相比，SiamDF 能够有效地处理复杂场景中的干扰并保持稳定的跟踪性能。

4) Skating1 序列

该测试序列的主要挑战是光照变化、遮挡和相似背景干扰。各跟踪算法的跟踪效果在第 175 帧时产生了不同，SiamRPN、TRACA 和 SINT 算法逐渐偏移了跟踪目标。在第 194 帧时，除了 SiamDF 算法和 TRACA 算法外，其它算法都发生了跟踪丢失的情况并跟踪到了背景中的相似干扰上。在第 280 帧，图像中的光照变暗，除了 SiamDF 算法外的跟踪算法都丢失了目标。实验结果表明，SiamDF 算法能适应跟踪过程中光照的变化，并在复杂场景下实现稳定跟踪。

4.3.2 VOT 数据集实验分析

本小节采用 VOT-2016 数据集对 SiamDF 及其它一些先进的算法进行对比，表 4.3 展示出了参与比较的各个算法在 VOT-2016 数据集上的测试结果，其中红、蓝、绿三种颜色分别对应跟踪性能排名第一、第二和第三的算法。

表 4.3 VOT-2016 数据集上不同算法的结果

Table 4.3 Tracking methods results on VOT-2016 dataset

跟踪器	EAO	Accuracy	Failure
TCNN [82]	0.3249	0.547	17.939
ECO-HC [74]	0.2665	0.510	21.402
SSAT [123]	0.3207	0.570	19.272
MLDF [123]	0.3106	0.487	15.044
DDC [123]	0.2929	0.534	20.981
SiamRN [123]	0.2766	0.549	23.996
MDNet_N [78]	0.2572	0.537	21.082
SiamAN [123]	0.2352	0.531	29.802
Staple [143]	0.2952	0.543	23.895
C-COT [72]	0.3310	0.533	16.582
TRACA [144]	0.1599	0.458	37.947
DCFNet [142]	0.2071	0.489	24.937
SiamFC [90]	0.2039	0.515	24.668
SiamRPN [95]	0.3441	0.579	20.138
SiamDF	0.3365	0.568	12.181

从表中可以看出，SiamDF 在 EAO 指标的表现上排名第二，在准确度指标的表现上排名第三，在平均失败次数的表现上排名第一。这充分说明，SiamDF 中改进的残差网络、多层特征融合和基于特征信息融合的模板更新方法能够在

定程度上提升算法的鲁棒性和准确性。然而,和在精度及 EAO 方面表现最好的 SiamRPN 算法相比,虽然 SiamDF 算法与其差距不大,但仍然需要对网络进行一定的改进以适应复杂场景下的尺度变化和低分辨率等挑战因素,从而进一步提升算法的跟踪性能。为了更加清晰地对比各算法的跟踪性能,在图 4.8 中对各算法的 EAO 排序进行了展示。

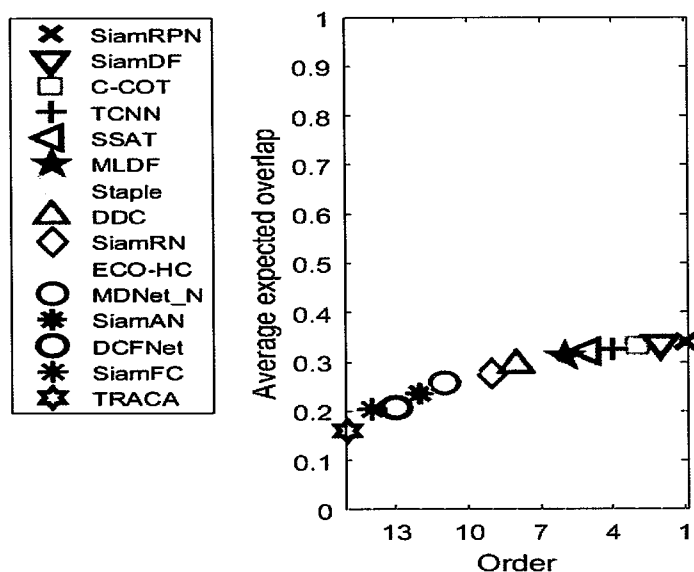


图 4.8 不同跟踪算法在 VOT-2016 数据集上的 EAO 结果

Figure 4.8 The EAO scores of different tracking algorithms on VOT-2016 dataset

4.3.3 消融实验

为了对本章提出的 SiamDF 算法中多层特征融合和基于特征信息融合的模板更新方法的有效性进行分析,本小节分别对两个模块进行增减并构建出不同的跟踪算法,从而能够在 OTB-2015 数据集上进行实验对比。

(1) 多层特征融合

为了对多层特征融合的有效性进行分析,本小节在 OTB-2015 数据集上对不同层级间特征的融合效果进行了对比验证。其中,采用融合策略的算法都运用了本章 4.2.3 小节的模板更新方法。图 4.9 展示了四种融合策略的对比实验结果,包括 Conv2 卷积层和 Conv4 卷积层特征融合 (SiamDF_WF2_4), Conv3 卷积层和 Conv4 卷积层特征融合 (SiamDF_WF3_4), Conv2、Conv3 和 Conv4 多层融合 (SiamDF_WF2-4) 和无多层特征融合 (SiamDF_OF4)。此外,在对比实验中还

将 SiamRPN 算法作为实验的基准算法，以便更加直观的看出各个构建算法的性能优劣。

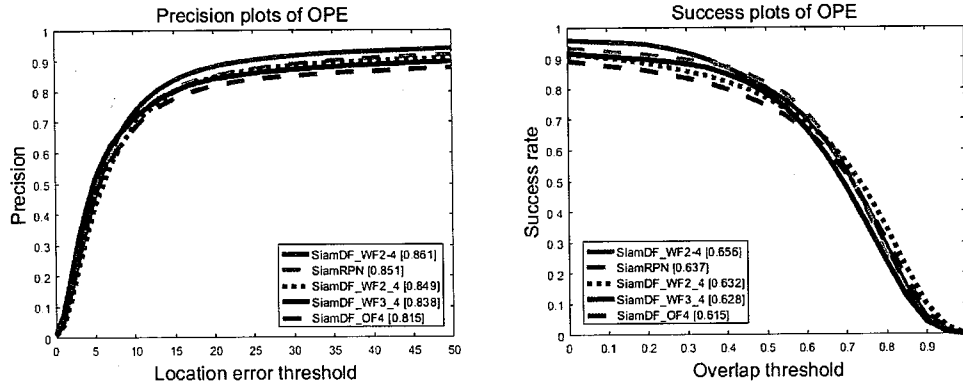


图 4.9 OTB-2015 数据集上不同融合策略的消融实验结果

Figure 4.9 The results of the ablation study with different fusion strategies on OTB-2015 dataset

从图 4.9 中可以看出，不采用任何融合策略的 SiamDF_OF4 算法具有较低的跟踪精度和重叠率，相比 SiamRPN 算法也具有较大的差距。然而，当在算法中引入融合策略时，算法的精度和重叠率都会得到明显的提升。与无任何融合策略的 SiamDF_OF4 相比，采用 Conv3 卷积层和 Conv4 卷积层特征融合的 SiamDF_WF3_4 算法在精度上可达到 83.8%，在重叠率上可达到 62.8%。采用 Conv2 卷积层和 Conv4 卷积层特征融合的 SiamDF_WF2_4 算法在精度上可达到 84.9%，在重叠率上可达到 63.2%。这充分说明，多层特征融合能够在一定程度上提升算法的准确性与鲁棒性。但是，上述三种算法和基准算法 SiamRPN 相比都没有取得优势。只有当采用 Conv2、Conv3 和 Con4 多层融合策略时，算法才能超过 SiamRPN 算法并取得最优的精度与重叠率。

(2) 基于特征信息融合的模板更新

为了对基于特征信息融合的模板更新方法进行分析，本小节也在 OTB-2015 数据集上对增减该更新方法的效果进行了对比验证。其中，构建的算法主要分为：有模板更新方法（SiamDF_WU）和无模板更新方法（SiamDF_OU），并都采用了 Conv2、Conv3 和 Con4 多层融合策略。与此同时，也在对比实验中将 SiamRPN 算法作为实验的基准算法，以便更加直观的看出各个构建算法的性能优劣。消融实验结果的精度曲线和成功曲线如图 4.10 所示。

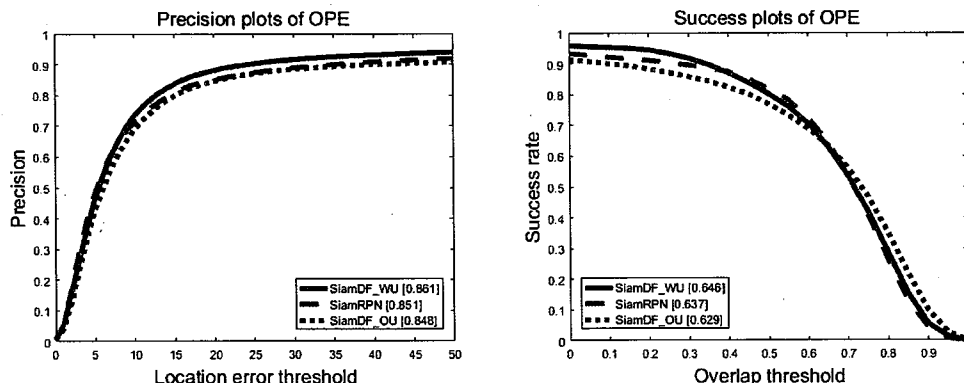


图 4.10 OTB-2015 数据集上的消融实验结果

Figure 4.10 The results of the ablation study on OTB-2015 dataset

从图中可以看出, 当不采用模板更新方法时, 虽然精度和重叠率达到了 84.8% 和 62.9%, 但和基准算法 SiamRPN 相比仍然存在一些差距。当在算法中采用模板更新方法时, 精度和重叠率都得到了提升并高于 SiamRPN 算法。这充分说明基于特征信息融合的模板更新方法能够在一定程度上提升算法的准确性与鲁棒性。

4.4 本章小结

本章提出一种结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。该算法利用一种改进的残差网络来替代传统的浅层网络, 从而提取出具有丰富语义信息的深层特征。与此同时, 本章算法提出一种高效的自适应多层特征融合策略, 有效提升了响应图的质量, 减少了在复杂场景下相似背景干扰所带来的影响。另外, 本章算法利用一种自适应特征信息融合方法来实现模板的更新, 从而能够适当减少模板更新过程中的计算量, 有效克服遮挡、目标形变等干扰因素的影响。仿真实验结果表明, 本章算法能够有效提升跟踪效果, 并且在准确性与鲁棒性等方面表现良好。

本章主要研究内容已经整理成文, 并发表学术论文一篇:

Daqun Li, Xize Wang, Yi Yu. Siamese visual tracking with deep features and robust feature fusion [J]. IEEE ACCESS, 2019(8): 3863-3874.

第5章 前景信息引导的孪生卷积神经网络目标跟踪算法

5.1 引言

通过第四章的实验结果分析可以看出,结合深层特征与鲁棒特征融合能够大幅提升目标跟踪性能并取得良好的跟踪效果。但是,当目标出现较严重形变或背景中出现较严重遮挡和相似背景干扰时,第四章提出的算法会出现跟踪性能不稳定甚至跟踪目标丢失的情况。通过对该算法进行研究,发现算法中网络训练采用的正样本对包含的挑战因素(遮挡、形变等)不够丰富,这会使网络提取的特征不够鲁棒。另外,大量训练数据集中的前景信息只是直接被应用,并没有采用策略对其进一步利用,从而造成算法无法有效地对语义背景和跟踪目标进行区分。这些问题同样也暴露出基于深度特征匹配的目标跟踪算法在训练时的缺陷。在离线训练阶段,大多数基于深度特征匹配的目标跟踪算法会采用包含丰富类别和训练数据的数据集进行训练,如 ILSVRC-2015 的 VID^[118]数据集。然而,这些具有有限类别的数据集并不足以提供充足的正样本对,从而难以训练一个具有高质量和强鲁棒性的网络模型。即使 DaSiamRPN 算法^[96]通过引入其它大规模的训练数据集来扩充正样本对的数量,但由于一些干扰因素(遮挡、形变等)具有长尾分布^[151],扩充数量的正样本对依然无法包含足够丰富的干扰因素来提高算法的鲁棒性。此外,大多数基于深度特征匹配的目标跟踪算法在训练过程中并没有对前景信息进行充分利用,这使得算法只能从一些简单的非语义背景中对跟踪目标进行定位。当应对语义背景丰富的复杂场景时,跟踪框会逐渐漂移到背景中的干扰源上,从而造成跟踪不稳定甚至是跟踪失败。

为了克服上述问题,本章提出一种前景信息引导的孪生卷积神经网络目标跟踪算法。该算法以 SiamFC 算法为基础,采用一种相对简单的采样策略来扩充离线训练时正样本对中的挑战因素,从而有效增强算法的鲁棒性。并且,为了进一步提升算法在语义背景下的识别能力,本章算法将背景信息进行填充遮挡,从而增强前景信息的显著性。与此同时,将处理后的数据输入到基于卷积神经网络的引导分支中,并运用一种填充损失计算方式来改进算法的损失函数,有效提升算

法在语义背景下的识别能力。此外，本章算法采用一种改进的特征信息融合方法来实现模板的更新，进一步应对目标的外观变化并提升算法的跟踪精度与鲁棒性。

5.2 前景信息引导的孪生卷积神经网络

在本节中，详细介绍本章所提出的前景信息引导的孪生卷积神经网络目标跟踪算法，并将其简称为 FIGSiam (Foreground Information Guidance for Siamese Visual Tracking)。首先，本节将对正样本对的扩充策略进行介绍。该扩充策略主要采用人工设计的遮挡掩膜和随机仿射变换对正样本对进行处理，这样不仅可以丰富数据集，还可以提高算法在应对遮挡和目标形变时的鲁棒性。接着，本节对前景信息引导和填充损失计算进行阐述，其可以引导跟踪算法在训练时更关注前景信息的变化，从而减少背景干扰对算法的影响。最后，本节对基于改进特征信息融合的模板更新方法进行论述。FIGSiam 跟踪算法的框架如图 5.1 所示。

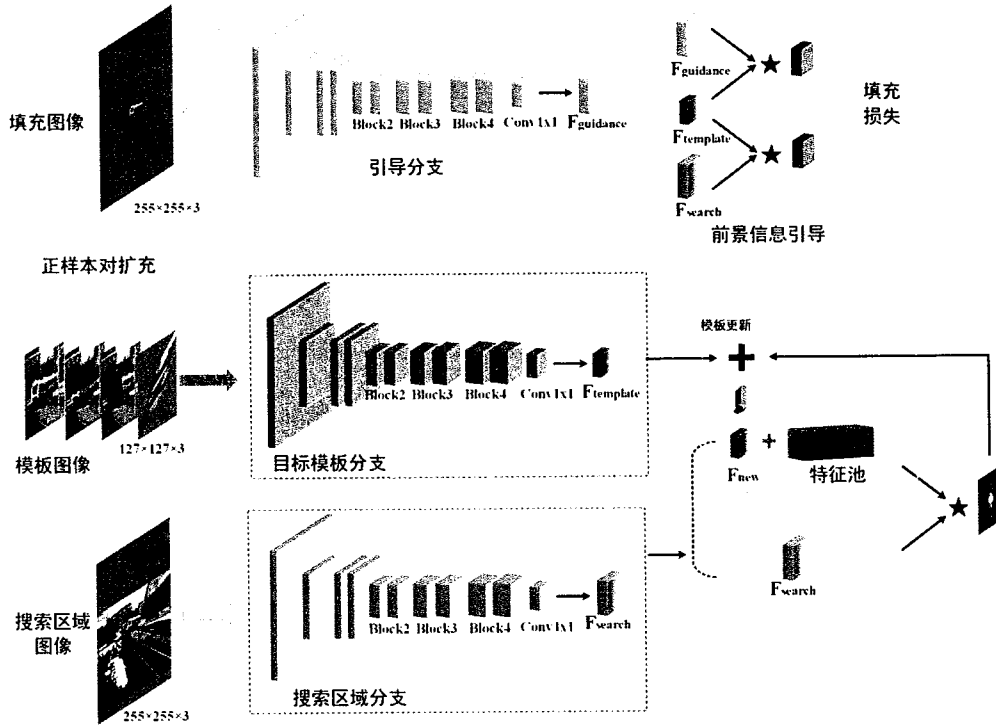


图 5.1 FIGSiam 算法的结构框架

Figure 5.1 The framework of the FIGSiam algorithm

5.2.1 正样本对的扩充

高质量的训练数据是基于深度特征匹配的目标跟踪算法成功的关键。目前，

大多数基于深度特征匹配的目标跟踪算法都是在大规模的数据集上训练的, 如 ILSVRC-2015 的 VID^[118]数据集, COCO^[149]数据集和 YouTube-BB 数据集^[152]等。然而, 这些具有有限类别的数据集并不足以提供充足的正样本对, 从而难以训练一个具有高质量和强鲁棒性的网络模型。尽管一些跟踪器通过组合一些大规模数据集来扩展正样本对, 但由于一些挑战性因素(遮挡、变形等)的长尾分布, 扩充数量的正样本对依然无法包含足够丰富的干扰因素来提高算法的鲁棒性。因此, FIGSiam 算法采用一种简单的采样策略来丰富正样本对中的挑战因素。

首先, 采样策略利用人工设计的遮挡掩膜来增加正样本对在遮挡干扰下的数量。在网络模型的离线训练阶段, 该策略将正样本对中的模板图像视为需要扩充的对象。具体而言, 根据模板图像中的目标信息可以得到目标的实际大小 $W \times H$ 。在得到目标大小后, 可以根据该信息生成预先设计好方向的遮挡掩膜, 并将掩膜的尺寸固定为 $W/2 \times H/2$ 。与此同时, 每种遮挡掩膜都将其覆盖的图像像素进行去除, 以此来模拟跟踪过程中的遮挡干扰并提高算法在应对遮挡时的鲁棒性。预先设计的 11 种不同方向的遮挡掩膜如图 5.2 所示。

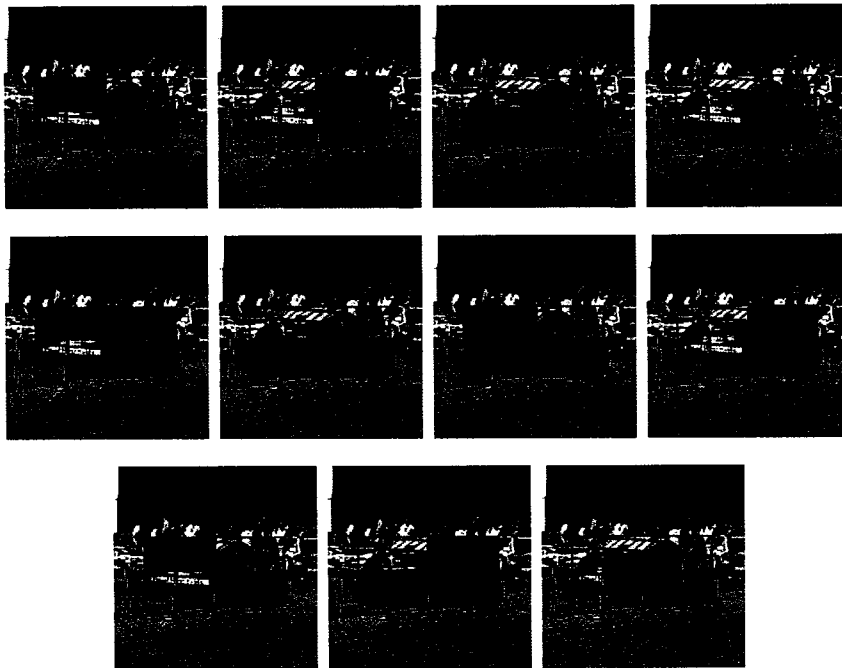


图 5.2 11 种不同方向的遮挡掩膜

Figure 5.2 11 directions of the handcrafted occlusion masks

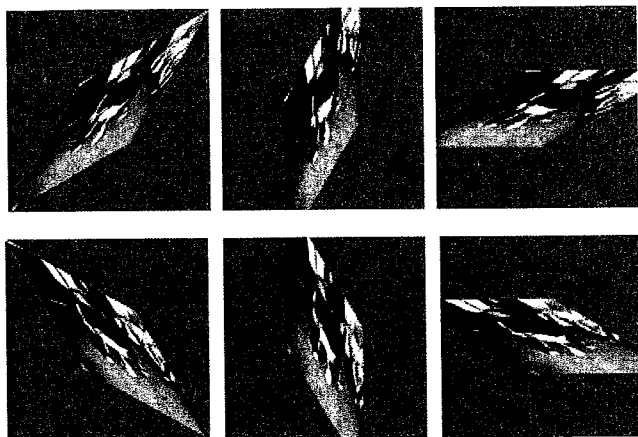


图 5.3 随机仿射变换样例

Figure 5.3 Representative samples of the random affine transformation

其次，采样策略采用旋转和错切映射（Shear Mapping）组合的随机仿射变换来模拟目标的形变。当给定模板图像后，该策略将模板图像进行范围在 $\theta = \pm 30^\circ$ 角度内的旋转。与此同时，模板图像会被进行 X 和 Y 方向上的错切映射，两个方向上的映射角度范围分别为 $\phi = \pm 25^\circ$ 和 $\psi = \pm 25^\circ$ 。上述的两种变换进行随机的组合，从而构成扩充策略中采用的随机仿射变换，这样不仅可以丰富数据集，还可以提高算法在应对目标形变时的鲁棒性。随机仿射变换的样例如图 5.3 所示。

5.2.2 前景信息引导

虽然，上一节中扩充的正样本对能够在离线训练阶段提升算法的鲁棒性和分辨能力，但其仍然难以从如图 5.4 所展示的语义背景中区分跟踪目标。为了解决这一问题，FIGSiam 算法对前景信息进行充分利用并运用一种填充损失计算方式来改进算法的损失函数，从而引导跟踪算法在训练时更关注前景信息的变化，有效减少背景干扰对算法的影响。

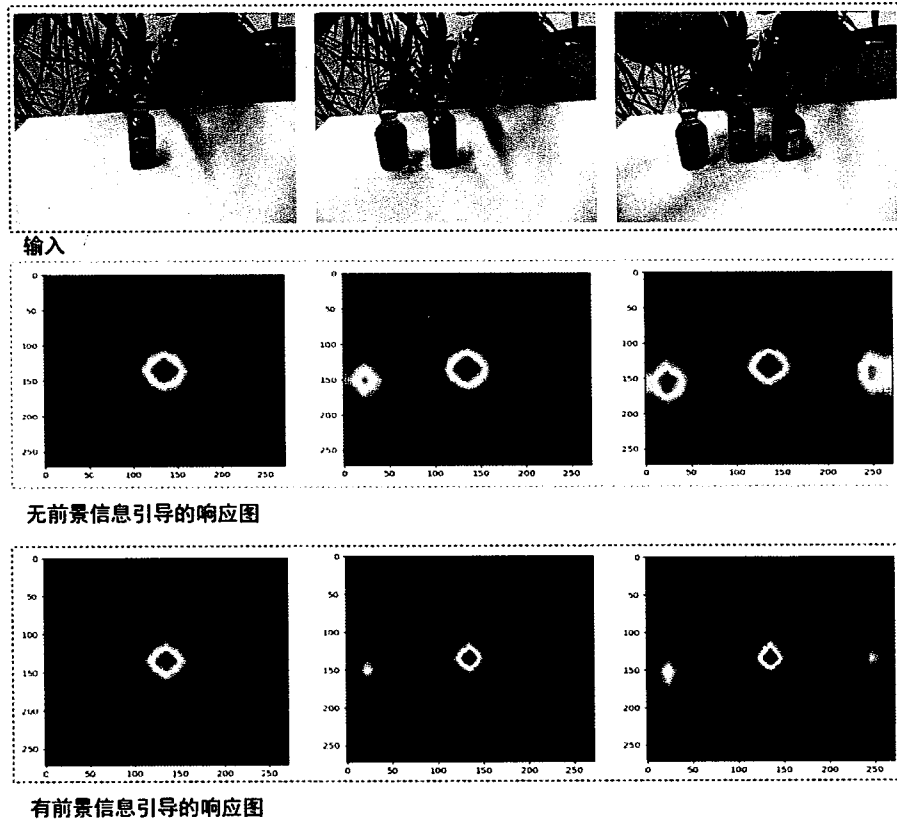


图 5.4 有无前景信息引导的响应图可视化

Figure 5.4 Visualization of response maps w/ or w/o the foreground information guidance.

在 SiamFC 算法中, 目标模板分支和搜索区域分支都采用 AlexNet 作为卷积特征提取网络。与此同时, SiamFC 算法采用式 (2.15) 获取最后的响应图, 并采用式 (2.16) 和式 (2.17) 所示的损失函数来训练网络模型。受 SiamFC 算法启发, FIGSiam 算法采用 4.2.1 小节中改进的 ResNet-18 网络作为各分支中提取卷积特征的网络。与 4.2.1 小节中的网络结构不同, FIGSiam 算法将原网络中大部分卷积核尺寸为 1×1 的卷积层去除, 只在网络最后保留了一个尺寸为 1×1 的卷积层以减少输出的维度。此外, 为了减少网络的计算负担, 将网络输出的特征张量的通道数调整为 256, 并裁剪其大小以保留中心区域(目标模板分支为 8×8 , 搜索区域分支为 24×24)。同时, 为了打破全卷积孪生网络结构严格的平移不变性, 在训练过程中依旧借鉴了 SiamRPN++^[97] 算法中提出的空间感知采样策略。

为了充分利用前景信息, FIGSiam 算法将引导分支加入到了整体架构中。如图 5.1 所示, 引导分支的输入为尺寸是 $255 \times 255 \times 3$ 的填充图像。该填充图像是采用整张图像的均值进行背景填充得到的, 填充方式如下:

$$Image = \begin{cases} Image(a, b) & (a, b) \in fg \\ Mean(Image) & (a, b) \in bg \end{cases} \quad (5.1)$$

式中， a 和 b 代表图像中第 a 行和第 b 列， fg 代表前景信息， bg 带边背景信息。并且，引导分支中用于特征提取的网络模型与目标模板分支和搜索区域分支中的模型相同，都采用改进的 ResNet-18 网络作为特征提取网络。当给定模板图像、搜索区域图像和填充图像后，FIGSiam 算法会将各个图像输入到对应的分支中提取卷积特征。接着，算法会在各分支输出的特征张量间进行深度互相关（Depthwise Cross Correlation, DW-XCorr）^[97] 计算。其中，目标模板分支与搜索区域分支间的深度互相关结果设为 S ，目标模板分支与引导分支间的深度互相关结果为 S' 。

在进行过各分支间的深度互相关计算后，FIGSiam 算法会在 SiamFC 算法的损失函数基础上引入一种填充损失，从而引导算法在训练时更关注前景信息的变化，减少背景干扰对算法的影响。填充损失的计算方式如下：

$$loss_{padding} = \sum_c^C \sum_a^H \sum_b^W \left\| \frac{S_{c,a,b}}{\max(\|S\|_2)} - \frac{S'_{c,a,b}}{\max(\|S'\|_2)} \right\|_2 \quad (5.2)$$

式中， C 、 H 和 W 分别代表深度互相关结果的通道数、高度和宽度。利用该填充损失，FIGSiam 算法会在训练时被引导去更少地关注背景信息，提升算法对前景信息的关注度，有效提升了算法在语义背景下的识别能力。融合填充损失后的损失函数如下：

$$loss_{final} = (1 - \lambda) loss_{training} + \lambda loss_{padding} \quad (5.3)$$

式中， λ 为融合各损失计算间的权重。值得注意的是，本小节提出的前景信息引导主要应用在网络的训练阶段。当网络训练结束并进行跟踪时，引导分支将会被屏蔽。

5.2.3 基于改进特征信息融合的模板更新

在大多数基于深度特征匹配的目标跟踪算法中，常采用固定的模板去搜索目标。这种固定模板的方式不仅会使跟踪算法无法适应目标外观的各种变化，还会因为复杂背景下各种干扰因素的影响导致跟踪失败。在第四章中提出的 SiamDF

算法采用一种基于特征信息融合的模板更新方法进行模板的更新。该方法虽然在一定程度上提高了算法在复杂场景下的准确性与鲁棒性,但其计算量较大,对于特征张量的建模较为复杂。此外,在特征池进行更新时,基于特征信息融合的模板更新方法将相似的模板特征进行融合,这样的更新策略容易将模板中的噪声引入到特征池中,不利于提升模板更新过程的稳定性与可靠性。为此,FIGSiam 算法采用一种改进特征信息融合的模板更新方法。该方法采用全局平均池化对特征进行降维处理,有效减少特征信息融合过程中的计算量。与此同时,该方法利用相似度计算对特征池中存储的特征进行选择替换,从而进一步提升模板更新过程的稳定性与可靠性。

在初始化阶段,FIGSiam 算法会建立一个特征池来存储可靠跟踪结果的可信特征:

$$F_{pool} = \{F_1, F_2, F_3, \dots, F_j \mid j \leq N\} \quad (5.4)$$

其中,这些可靠跟踪结果是通过对各帧的跟踪结果进行评估(见 4.2.3 小节)后提取出来的, N 代表特征池存储特征的上限。当得到新一帧的跟踪结果后,算法会首先对跟踪结果进行评估,并将评估通过的结果输入到目标模板分支中提取特征,得到尺寸为 $8 \times 8 \times 256$ 的特征张量 F_{new} 。此外,为了减轻计算负担,算法会利用全局平均池化(Global Average Pooling, GAP)操作对 F_{new} 和特征池 F_{pool} 里的特征进行处理,从而得到尺寸为 $1 \times 1 \times 256$ 的 F_{new}' 和 F_j' 。接下来,算法会利用 L2 范数归一化对 F_{new}' 和 F_j' 进行归一化处理,利用下式得到二者之间的相似度:

$$S_{matching}^j = \max(0, (F_{new}')^T F_j') \quad (5.5)$$

最后,新的特征张量 F_{new} 和特征池中的特征 F_{pool} 可以采用下式进行融合:

$$F_{final} = \varepsilon F_{new} + (1 - \varepsilon) \sum_{j=1} F_j \delta_j \quad (5.6)$$

式中, ε 代表特征融合时的权重, δ_j 代表特征池中各特征的自适应权重,其计算公式如下:

$$\delta_j = \frac{S_{matching}^j}{\sum_{j=1} S_{matching}^j} \quad (5.7)$$

在进行模板更新的过程中，当特征池中存储的可信特征少于上限 N 时，算法会利用下列条件对特征 F_{new} 进行判断：

$$Mean(S_{matching}^i) \geq Th_{update} \quad (5.8)$$

如果 F_{new} 满足条件，则可将其直接放入特征池 F_{pool} 中。其中， $Mean(\cdot)$ 代表均值操作。当特征池中存储的可信特征高于上限 N 时，算法会用满足条件的特征替代特征池中与其相似度最低的特征，从而实现特征池的更新。

5.2.4 算法描述

在跟踪过程中，首先会给定第一帧中的目标信息，FIGSiam 算法根据该信息进行算法初始化，如目标模板截取、模板特征提取、搜索尺度确立等。接着，FIGSiam 算法会在下一帧截取搜索区域，并将搜索区域图像输入到改进的 ResNet-18 网络中进行特征提取，并利用交叉相关计算获取响应图，从而利用该响应图确定最后的跟踪结果。然后，FIGSiam 算法会对响应图进行质量评价，计算其 APCE 值是否高于阈值 T_{APCE} 。如果高于阈值，则会将跟踪结果输入到模板分支中进行可信特征的提取。最后，FIGSiam 跟踪算法会结合得到的可信特征与特征池进行特征信息融合，从而实现模板的自适应更新。FIGSiam 算法的运行流程如下表所示：

表 5.1 FIGSiam 算法运行流程

Table 5.1 The execution process of FIGSiam

前景信息引导的孪生卷积神经网络目标跟踪算法（FIGSiam）
输入：利用扩充正样本对和前景信息引导预训练好的网络，初始帧目标信息
Step 1: 利用初始帧目标信息进行算法初始化（目标模板截取、模板特征提取、搜索尺度确立等）
对于图像序列帧数 $frame=2,3,\dots,N$ 依次循环 Setp2-5
Step 2: 在第 t 帧图像上的 (x_{t-1}, y_{t-1}) 周围确定搜索区域，利用改进的 ResNet-18 网络提取特征张量，利用交叉相关计算获取响应图，并确定跟踪结果 (x_t, y_t) 和 (w_t, h_t) 。
Step 3: 通过式 (3.6) 对当前跟踪结果对应的响应图进行质量评价。如果高于阈值 T_{APCE} ，将新跟踪到的结果输入到目标模板分支中提取可信特征
Step 4: 利用式 (5.5)、式 (5.6) 和式 (5.7) 进行特征信息融合，以实现模板特征的更新

Step 5: 利用式 (5.8) 对特征池进行更新

输出: 当前帧目标的位置 (x_t, y_t) 和大小 (w_t, h_t)

5.3 实验结果与分析

本章算法在 PyCharm 下利用 pytorch-0.4.1 深度学习工具集实现。用于实验的 PC 机配置为: Intel i7-9800X CPU (3.80GHz), 64GB RAM 和一块 NVIDIA TITAN RTX GPU。算法平均运行速度为 35 帧每秒。FIGSiam 算法中目标模板分支、搜索区域分支和引导分支都采用改进的 ResNet-18 网络。在训练该网络时, 采用 ILSVRC-2015 的 VID^[118]和 COCO 数据集^[149]进行离线训练, 且网络中每层参数的初始值遵循高斯分布。采用 SGD 方法对网络进行优化, 设置权重衰减 (Weight Decay) 为 0.0005。整个训练过程为 50 个迭代周期, 学习率在对数空间从 0.01 衰减到 0.0001, 损失函数中融合各损失计算间的权重 λ 设为 0.5。与此同时, 训练过程中的动量值设置为 0.9, 每个小批量 (mini-batch) 包含 64 个样本。另外, FIGSiam 算法中用于特征信息融合的参数是通过大量实验验证得到的, 特征融合时的权重 ε 设为 0.6, 特征池存储可信特征的上限 N 设为 30, 特征池更新时的阈值 Th_{update} 设为 0.65, 响应图质量评价时的阈值 T_{APCE} 设为 8。

为了客观地对 FIGSiam 算法的性能进行评价, 本小节采用 OTB 数据集、VOT-2016 数据集和 VOT-2018 数据集对 FIGSiam 及其它一些先进的算法进行对比, 从而验证本章算法的准确性、鲁棒性及有效性。

(1) OTB 数据集

OTB 数据集中的测试序列包含了完整的标注信息和种类繁多的跟踪目标。该数据集主要包含两个版本: 一个版本是由 50 个测试序列组成的 OTB-2013, 另一个版本是由 100 个测试序列组成的 OTB-2015, 是 OTB-2013 的扩展集并简称为 OTB-100。本小节的实验采用以上两个数据集, 并利用 2.4.1 小节中介绍的评价指标对算法进行评估。其中, 参与对比实验的算法包括 SINT^[86]、SiamFC^[90]、GOTURN^[87]、CFNet^[91]、DCFNet^[141]、SiamRPN^[95]、ECO_HC^[74]、TRACA^[144]和 RFL^[150]算法。与此同时, 实验采用精度曲线 (Precision Plot) 和成功曲线 (Success Plot) 进行直观的比较, 并采用 OPE 评价方法对算法进行评估。

(2) VOT 数据集

实验采用的 VOT 数据集是最常用的 VOT-2016 数据集^[123]和 VOT-2018^[124]数据集。其中, VOT-2016 数据集包含了 60 个具有多种挑战性干扰因素的测试序列, 其中有少部分的测试序列与 OTB 数据集相同。VOT-2018 数据集是在 VOT-2016 数据集的基础上改进而来的。其将 VOT-2016 数据集中一些不具有挑战性的视频替换为一些更难的视频, 从而更加全面、客观、有效地对算法进行评估。在 VOT-2016 数据集上, 开展了与其它 14 种跟踪算法的实验对比, 包括 EBT^[145]、MDNet_N^[78]、C-COT^[72]等。在 VOT-2018 数据集上, 同样开展了与其它 14 种跟踪算法的对比, 包括 SA_Siam_R^[153]、SiamRPN^[95]、LADCF^[154]和 DensSiam^[155]等。并且, 在两个数据集上分别进行测试时, 都利用 2.4.2 小节中介绍的评价指标对算法进行评估。

5.3.1 OTB 数据集实验分析

图 5.5 和图 5.6 分别对 OTB-2013 上和 OTB-2015 上的实验结果进行了展示。从图中可以看出, FIGSiam 算法在 OTB-2013 数据集上具有排名第二的精度与排名第一的重叠率。在精度方面, FIGSiam 比排名第三的 SiamRPN 高出 0.4%, 比 SiamFC 跟踪算法在精度上高出 7.3%。在重叠率方面, FIGSiam 比排名第二的 SiamRPN 高出 2.1%, 比 SiamFC 算法高出 6.7%。FIGSiam 算法在 OTB-2015 数据集上排名第一, 比 SiamRPN 跟踪算法在精度和重叠率上高 1.5%和 2.1%, 比 SiamFC 算法在精度和重叠率上高 9.3%和 7.3%。上述实验结果充分证明, FIGSiam 算法能够利用改进的 ResNet-18 网络、扩充的正样本对、前景信息引导和改进的特征信息融合有效提升复杂场景下算法的跟踪精度及鲁棒性。

与其它基于深度特征匹配的跟踪算法相比, FIGSiam 算法的优越性更加明显。FIGSiam 采用人工设计的遮挡掩模和随机仿射变换来丰富正样本对中的挑战因素, 从而有效地提高跟踪算法的鲁棒性。同时, FIGSiam 算法通过填充背景来增强前景信息的显著性, 并运用前景信息引导使跟踪算法在训练时更关注前景信息的变化, 有效减少背景干扰对算法的影响。此外, 与其它基于深度特征匹配的跟踪算法不同的是, FIGSiam 算法在框架中引入了基于改进特征信息融合的模板更新方法, 使算法在复杂场景下应对遮挡、目标形变等干扰时, 能够克服这些限制

并取得良好的跟踪效果。

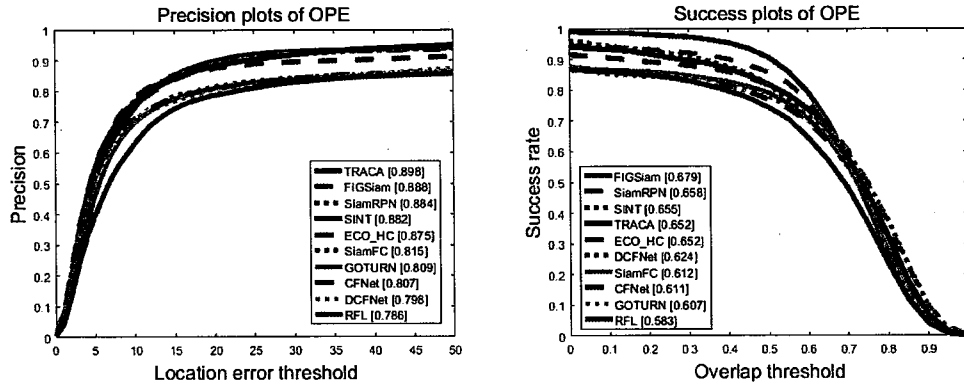


图 5.5 OTB-2013 数据集上的实验结果

Figure 5.5 The tracking results on OTB-2013 dataset

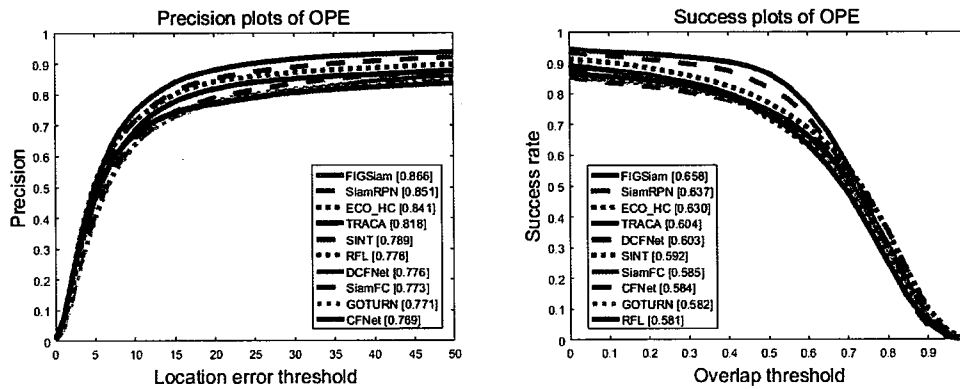


图 5.6 OTB-2015 数据集上的实验结果

Figure 5.6 The tracking results on OTB-2015 dataset

与基于相关滤波器的跟踪算法 (ECO_HC 和 TRACA) 相比, FIGSiam 算法仍然具有较好的性能。FIGSiam 将处理后的数据输入到基于卷积神经网络的引导分支中, 并运用一种填充损失计算方式来改进算法的损失函数, 从而有效提升算法在语义背景下的识别能力。此外, 基于改进特征信息融合的模板更新方法也是 FIGSiam 算法获得优势的关键。

为了进一步进行比较, 图 5.7 和图 5.8 展示了各算法在 OTB-2015 数据集上 11 种不同属性 (见 2.4.1 小节) 下精度和重叠率的比较。

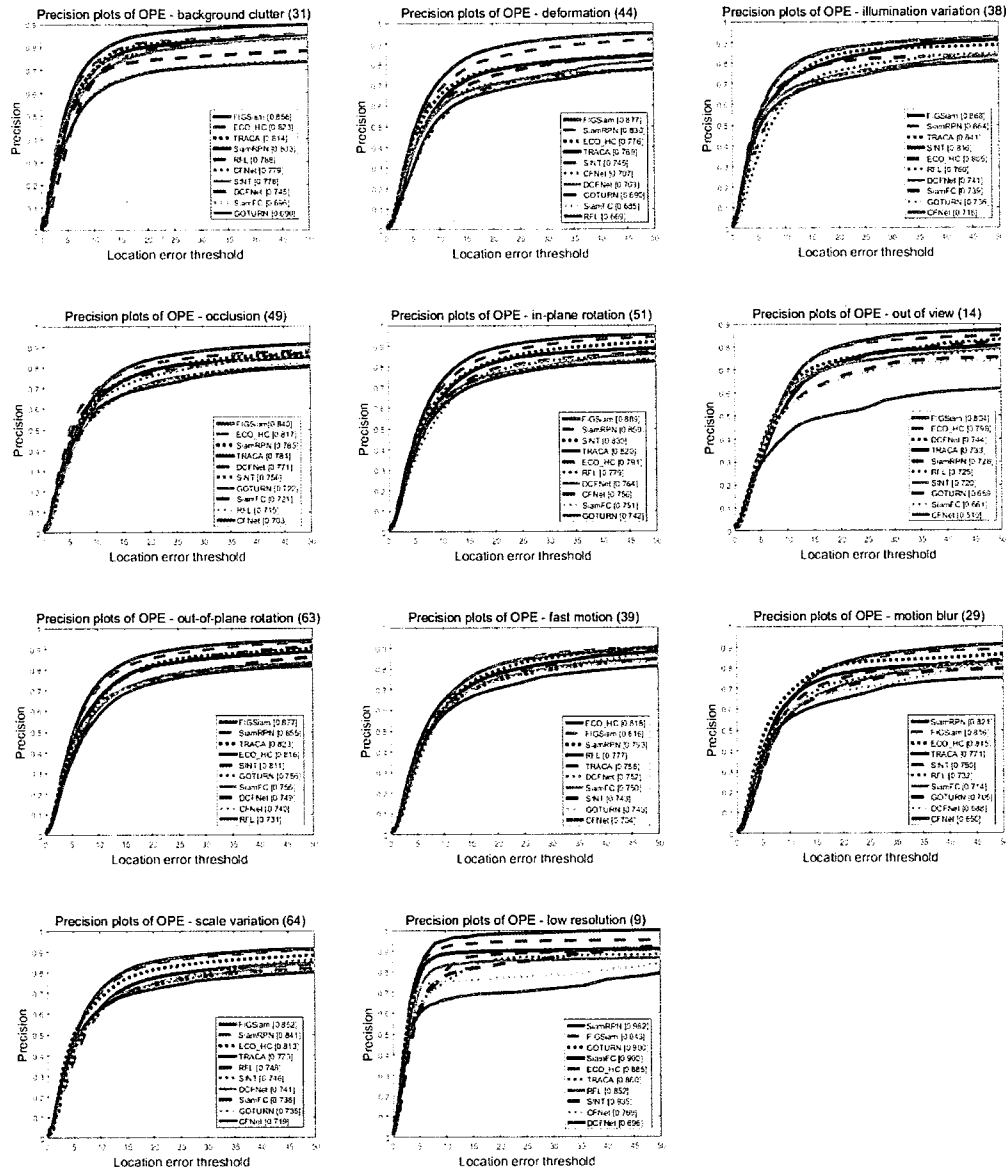


图5.7 OTB-2015数据集上11种属性下的实验结果（精度）

Figure 5.7 Tracking results on 11 challenging attributes on the OTB-2015 dataset (Precision)

具体而言，由于 FIGSiam 算法利用改进的 ResNet-18 网络来代替浅层网络，并采用前景信息引导和填充损失计算来减少背景干扰对算法的影响，因此对于复杂场景下的背景干扰和光照变化等挑战因素具有较好的跟踪精度。另外，FIGSiam 算法采用扩充的正样本对进行网络模型的训练，并在算法中融入了基于改进特征信息融合的自适应模板更新方法，从而在遮挡、目标形变、目标旋转和尺度变换等属性下具有排名较好的跟踪精度。值得一提的是，虽然 FIGSiam 算法在尺度变化下的精度较好，但是其重叠率却较低。这说明采用前景信息引导和

填充损失虽然能在一定程度上提升精度,但采用有限的尺度变换因子并不能较好地满足算法高重叠率的要求,需要在算法中融入更准确、更鲁棒的多尺度计算策略,从而在复杂背景下更好地应对尺度变化带来的影响。另外,FIGSiam 算法在快速运动、运动模糊和低分辨率等属性下的表现并不是最优的。这说明单纯地采用深层网络和优化损失函数并不能全面提高算法的精度,如何进一步对深层网络进行改进与提升才是解决问题的关键。

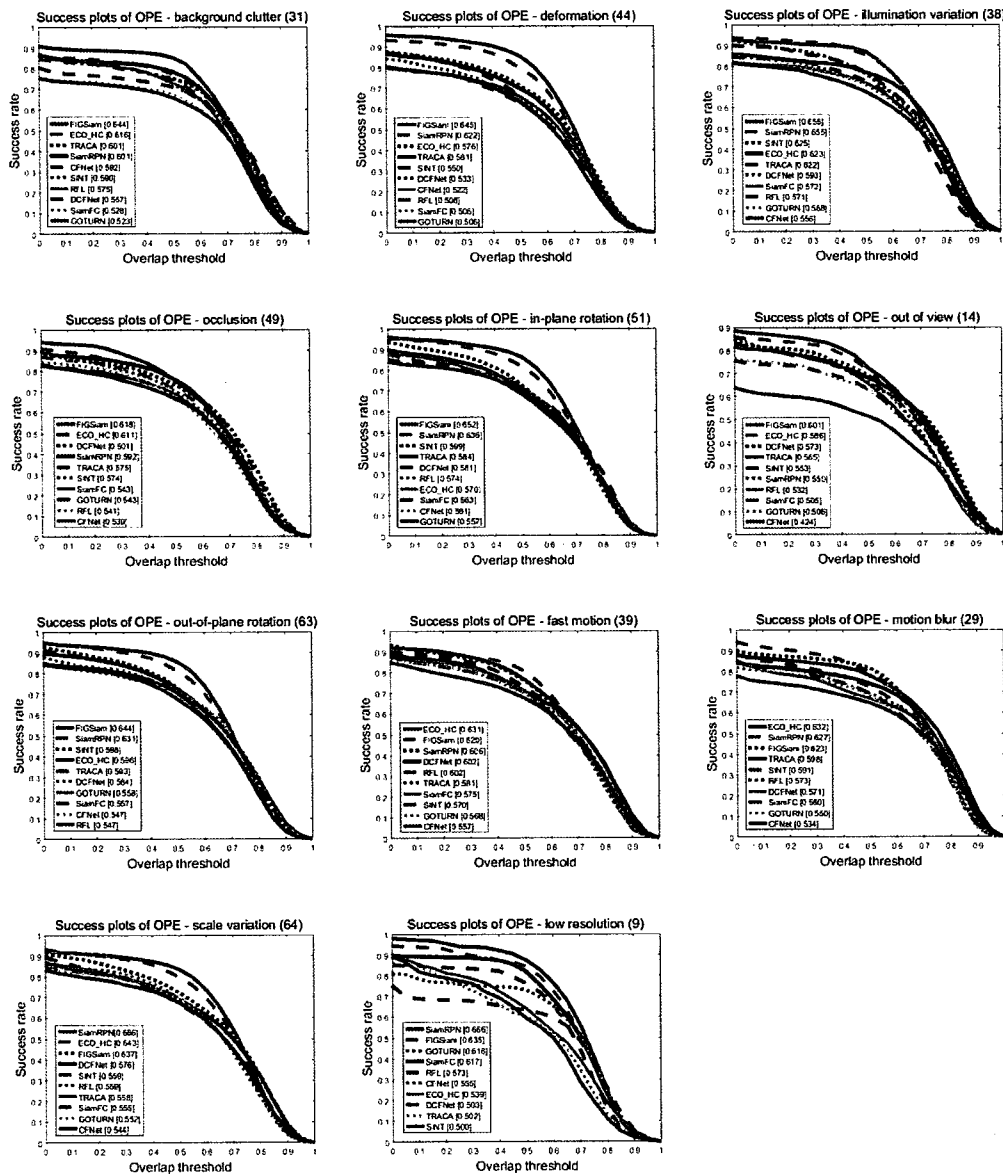


图5.8 OTB-2015数据集上11种属性下的实验结果(重叠率)

Figure 5.8 Tracking results on 11 challenging attributes on the OTB-2015 dataset (OR)

5.3.2 VOT 数据集实验分析

(1) VOT-2016 数据集

本小节采用 VOT-2016 数据集对 FIGSiam 及其它一些先进的算法进行对比, 表 5.2 展示出了参与比较的各个算法在 VOT-2016 数据集上的测试结果, 其中红、蓝、绿三种颜色分别对应跟踪性能排名第一、第二和第三的算法。

表 5.2 VOT-2016 数据集上不同算法的结果

Table 5.2 Tracking methods results on VOT-2016 dataset

跟踪器	EAO	Accuracy	Failure
TCNN [82]	0.3249	0.547	17.939
ECO-HC [74]	0.2665	0.510	21.402
SSAT [123]	0.3207	0.570	19.272
MLDF [123]	0.3106	0.487	15.044
DDC [123]	0.2929	0.534	20.981
SiamRN [123]	0.2766	0.549	23.996
MDNet_N [78]	0.2572	0.537	21.082
SiamAN [123]	0.2352	0.531	29.802
Staple [143]	0.2952	0.543	23.895
C-COT [72]	0.3310	0.533	16.582
TRACA [144]	0.1599	0.458	37.947
DCFNet [142]	0.2071	0.489	24.937
SiamFC [90]	0.2039	0.515	24.668
EBT [145]	0.2913	0.453	15.194
FIGSiam	0.3411	0.576	16.152

从表中可以看出, FIGSiam 在 EAO 指标的表现上排名第一, 在准确度指标的表现上排名第一。这充分说明, FIGSiam 中扩充的正样本对、前景信息引导、填充损失和基于改进特征信息融合的模板更新方法能够在一定程度上提升算法的鲁棒性和准确性。然而, 和其它在平均失败次数方面表现较好的跟踪算法相比, FIGSiam 在平均失败次数的表现上排名第三。这说明单纯地采用深层网络和优化损失函数并不能全面提高算法的精度, 需要对网络进行一定的改进以适应复杂场景下的尺度变化、低分辨率等挑战因素, 从而进一步提升算法的跟踪性能。为了更加清晰地对比各算法的跟踪性能, 在图 5.9 中对各算法的 EAO 排序进行了展示。

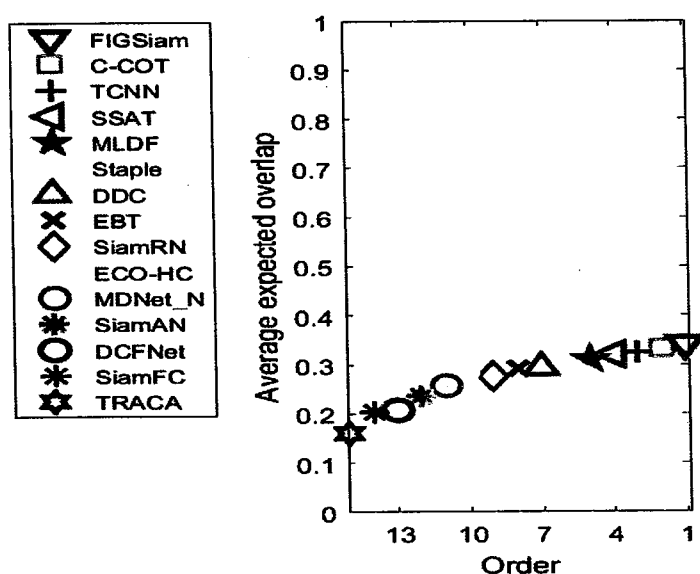


图 5.9 不同跟踪算法在 VOT-2016 数据集上的 EAO 结果

Figure 5.9 The EAO scores of different tracking algorithms on VOT-2016 dataset

(2) VOT-2018 数据集

我们在 VOT2018 数据集上对 FIGSiam 跟踪算法进行了评估, 并与 14 种先进的跟踪算法进行了比较。表 5.3 展示出了参与比较的各个算法在 VOT-2018 数据集上的测试结果, 其中红、蓝、绿三种颜色分别对应跟踪性能排名第一、第二和第三的算法。

表 5.3 VOT-2018 数据集上不同算法的结果

Table 5.3 Tracking methods results on VOT-2018 dataset

跟踪器	EAO	Accuracy	Failure
DLSTPP [124]	0.3273	0.542	13.695
ECO [74]	0.2809	0.476	17.663
SiamFC [90]	0.1875	0.498	34.026
SiamVGG [124]	0.2872	0.527	20.453
SA_Siam_P [124]	0.2863	0.521	19.467
SA_Siam_R [153]	0.3374	0.557	15.555
TRACA [144]	0.1385	0.431	53.678
LADCF [154]	0.3889	0.502	8.774
C-COT [72]	0.2674	0.485	20.414
DSiam [92]	0.1952	0.506	40.087
DCFNet [142]	0.1827	0.465	35.202
UpdateNet [99]	0.2442	0.510	26.872

DensSiam [155]	0.1732	0.456	44.448
SiamRPN [95]	0.3837	0.580	17.661
FIGSiam	0.3386	0.511	17.632

在对比的跟踪算法中，LADCF 算法在 EAO 和平均失败次数方面表现最佳。SiamRPN 算法的精度排名第一，EAO 排名第二。尽管我们的 FIGSiam 算法可以比大多数基于深度特征匹配的跟踪算法表现更好，但它的精度和鲁棒性较比优秀的跟踪算法还有一定差距。这充分说明，单纯地采用深层网络和优化损失函数并不能全面提高算法的精度。并且，FIGSiam 算法采用的模板更新策略比较传统，需要进一步结合深度学习以应对更严峻的挑战。为了更加清晰地对比各算法的跟踪性能，在图 5.10 中对各算法的 EAO 排序进行了展示。

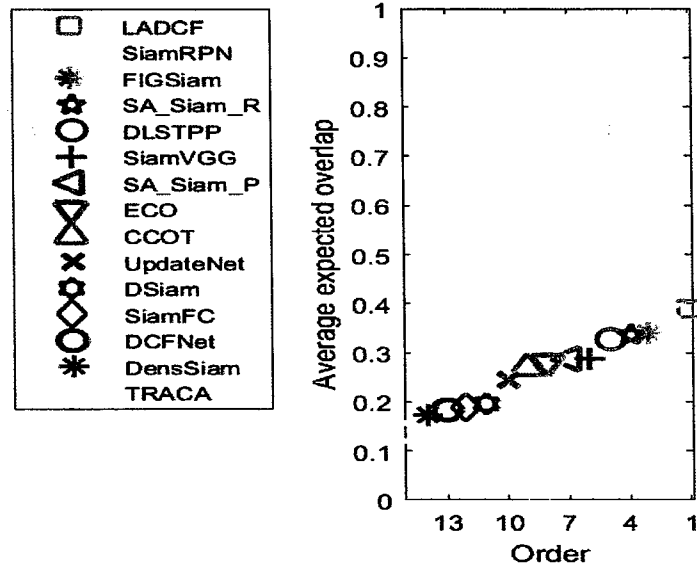


图 5.10 不同跟踪算法在 VOT-2018 数据集上的 EAO 结果

Figure 5.10 The EAO scores of different tracking algorithms on VOT-2018 dataset

5.3.3 消融实验

为了对本章提出的 FIGSiam 算法中扩充的正样本对、前景信息引导和基于改进特征信息融合的模板更新方法的有效性进行分析，本小节对三个模块进行增减并在 OTB-2015 数据集和 VOT-2018 数据集上进行实验对比。

如表 5.4 所示，在实验中采用 SiamRPN 算法作为对比的基准算法。在 OTB-2015 数据集中，当在训练中添加扩充的正样本对时，FIGSiam 算法的 AUC

得分增加到了 0.643。当在训练中添加前景信息引导时, FIGSiam 算法的 AUC 得分增加到了 0.655。当在训练中添加基于改进特征信息融合的模板更新方法时, FIGSiam 算法的 AUC 得分增加到了 0.658。这充分说明, FIGSiam 中扩充的正样本对、前景信息引导和基于改进特征信息融合的模板更新方法能够在一定程度上提升算法的鲁棒性和准确性。

虽然, SiamRPN 算法在 VOT-2018 数据集上的表现较比 FIGSiam 要更好一些。但是, FIGSiam 采用不同模块所产生的效果有所区别, 与 SiamRPN 的差距也明显不同。当采用前景信息引导时, FIGSiam 算法的 EAO 得分会从 0.2991 上升到 0.3367。此外, 当模板更新引入到 FIGSiam 算法中时, EAO 得分最高且与 SiamRPN 算法的差距最小。这充分验证了 FIGSiam 算法中各模块的有效性。

表 5.4 消融实验结果

Table 5.4 The results of the ablation study

模 块	SiamRPN	FIGSiam		
扩充的正样本对		✓	✓	✓
前景信息引导			✓	✓
模板更新				✓
AUC in OTB100	0.637	0.643	0.655	0.658
EAO in VOT2018	0.3889	0.2991	0.3367	0.3386

5.4 本章小结

本章提出一种前景信息引导的孪生卷积神经网络目标跟踪算法。该算法采用扩充的正样本对来增强算法的鲁棒性。并且, 为了进一步提升算法在语义背景下的识别能力, 本章算法将背景信息进行填充遮挡, 从而增强前景信息的显著性。与此同时, 将处理后的数据输入到基于卷积神经网络的引导分支中, 并运用一种填充损失计算方式来改进算法的损失函数, 有效提升算法在语义背景下的识别能力。此外, 本章算法采用一种改进的特征信息融合方法来实现模板的更新, 进一步应对目标的外观变化并提升算法的跟踪精度与鲁棒性。仿真实验结果表明, 本章算法在多种干扰因素下取得了较好的跟踪效果, 优于所对比的其它多个先进的跟踪算法。

本章主要研究内容已经整理成文，并发表学术论文一篇：

Daqun Li, Yi Yu. Foreground information guidance for Siamese visual tracking [J].

IEEE ACCESS, 2020(8): 55905-55914.

第6章 本文算法对比实验分析

为了客观地对本文提出的 Siam-RM、SiamDF 和 FIGSiam 三种算法的性能进行对比分析,本章采用 OTB 数据集、VOT-2016 数据集和 VOT-2018 数据集进行对比实验,从而能够对三种算法的准确性、鲁棒性及有效性进行全面分析。与此同时,为了有效验证本文提出的三种算法在实际应用场景中的跟踪性能,本章利用无人机航拍获取的彩色视频数据 UAV123^[119]及地基光测设备采集的视频数据进行实验分析,在进行跟踪性能比较和综合效果分析的同时,进一步明确了基于深度特征匹配的目标跟踪算法适用的场景与对象,使本课题的研究意义与应用价值更加凸显。

6.1 OTB 数据集对比实验分析

图 6.1 和图 6.2 分别对 OTB-2013 上和 OTB-2015 上的实验结果进行了展示。从图中可以看出, FIGSiam 算法在 OTB-2013 数据集上具有排名第一的精度与排名第一的重叠率。在精度方面, FIGSiam 比排名第二的 SiamDF 高出 0.5%, 比 Siam-RM 跟踪算法在精度上高出 1%。在重叠率方面, FIGSiam 比排名第二的 SiamDF 高出 1.8%, 比 Siam-RM 算法高出 3.3%。FIGSiam 算法在 OTB-2015 数据集上排名第一, 比 SiamDF 跟踪算法在精度和重叠率上高 0.5%和 0.2%, 比 Siam-RM 算法在精度和重叠率上高 2%和 4.5%。

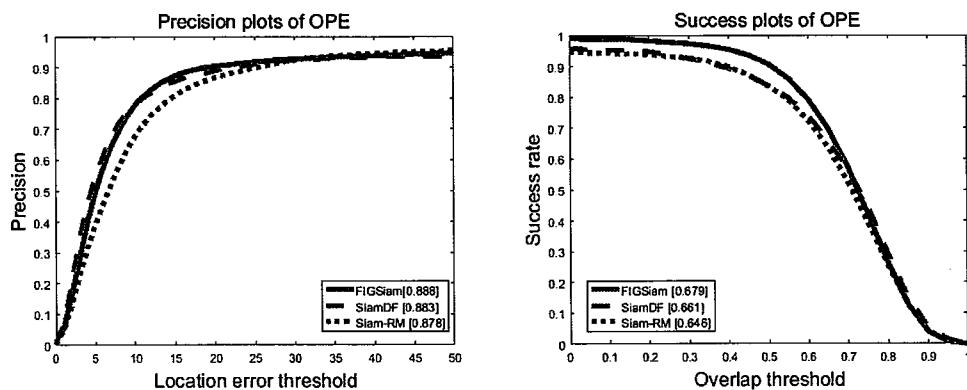


图 6.1 OTB-2013 数据集上的实验结果

Figure 6.1 The tracking results on OTB-2013 dataset

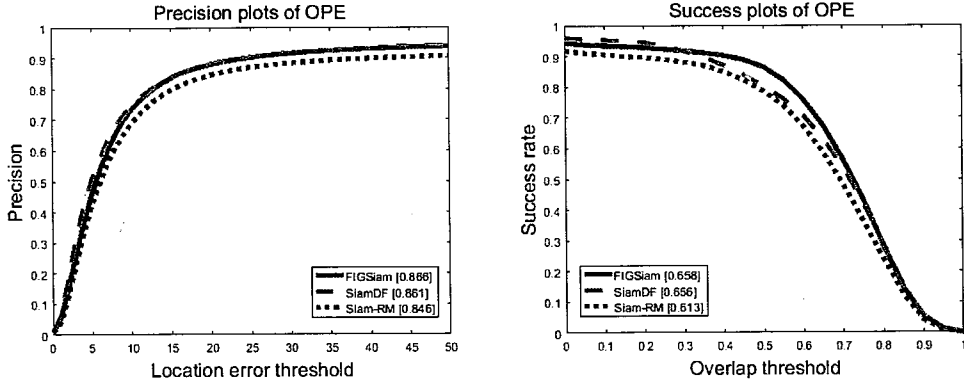


图 6.2 OTB-2015 数据集上的实验结果

Figure 6.2 The tracking results on OTB-2015 dataset

上述实验结果充分证明, FIGSiam 算法较比其它两种算法在综合性能方面具有更加优越的表现。与 SiamDF 算法相比, 虽然二者都采用深层网络进行语义特征的提取, 但扩充的正样本对和前景信息引导的融入使得 FIGSiam 算法在训练过程中具有更加丰富的训练样本和引导信息, 从而使网络模型具有更高的质量和更强的鲁棒性。此外, 改进的特征信息融合方法提升了模板更新过程的可靠性, 使算法在应对目标外观变化时更加稳定。与其它两种算法不同, Siam-RM 算法利用浅层卷积神经网络提取特征, 这使得提取的特征具有较少的语义信息。虽然, 再检测机制的引入能够使算法在一些低复杂度的场景下具有较高的精度, 但与其它两种具有深层网络结构的算法相比, 其在复杂场景下具有明显的劣势。

为了进一步进行比较, 图 6.3 和图 6.4 展示了三种算法在 OTB-2015 数据集上 11 种不同属性 (见 2.4.1 小节) 下精度和重叠率的比较。具体而言, FIGSiam 算法利用扩充的正样本对来提供更加丰富的训练样本和引导信息, 并采用前景信息引导和填充损失计算来增强网络模型的可靠性。另外, FIGSiam 算法中融入了基于改进特征信息融合的自适应模板更新方法, 从而在遮挡、目标形变和目标旋转等属性下具有排名较好的跟踪精度与重叠率。值得一提的是, 虽然 FIGSiam 算法的综合性能要比 SiamDF 算法优越, 但在背景杂乱、光照变化、快速运动和运动模糊等属性下具有劣势。这充分表明, SiamDF 算法在深层网络结构下采用多层特征融合策略对不同网络层级中的卷积特征进行自适应融合, 会明显提升算法在复杂场景下的跟踪性能, 从而在应对背景杂乱及光照变化等挑战性因素时更加稳定可靠。

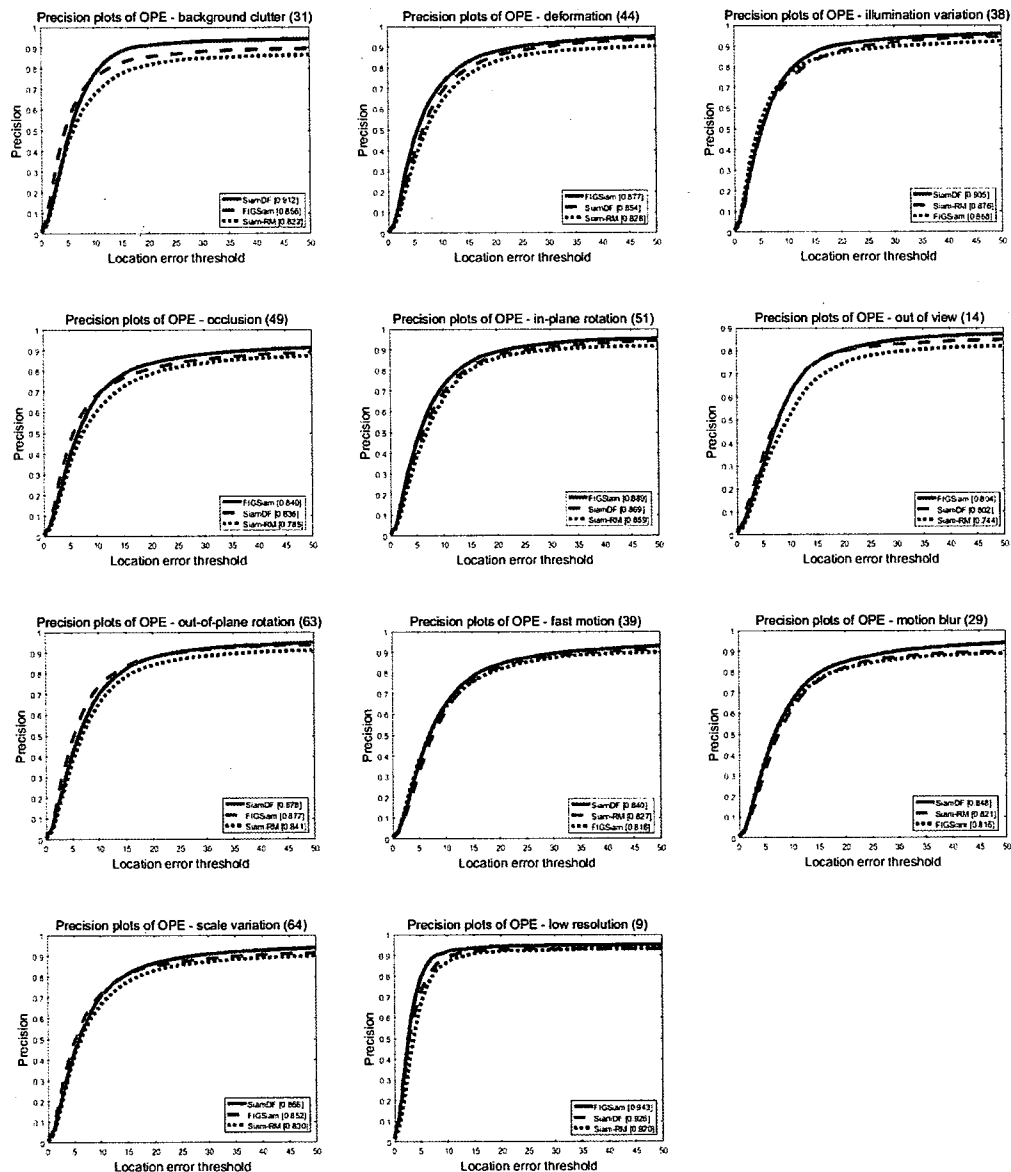


图6.3 OTB-2015数据集上11种属性下的实验结果（精度）

Figure 6.3 Tracking results on 11 challenging attributes on the OTB-2015 dataset (Precision)

与其它两种算法相比，Siam-RM 算法在 11 种属性下的表现较差。Siam-RM 算法采用浅层卷积神经网络提取特征，与其它两种具有深层网络结构的算法相比，其在复杂场景下具有明显的劣势。虽然，Siam-RM 算法引入再检测孪生卷积神经网络作为再检测机制，但单一尺度的再检测网络并不能较好地满足算法高精度与高重叠率的要求，需要在网络训练时融入多尺度的训练样本或在推理时融入多尺度的搜索策略。另外，本文提出的三种算法都采用了自适应模板更新方法来应对遮挡、目标形变等挑战因素。从实验结果可以看出，SiamDF 算法采用的基于

特征信息融合的模板更新方法要明显优于 Siam-RM 算法中基于生成式模型的自适应模板更新方法, FIGSiam 算法在训练时引入遮挡掩膜和随机仿射变换并采用改进特征信息融合的模板更新方法, 这使其在遮挡、目标形变等属性下表现最优。

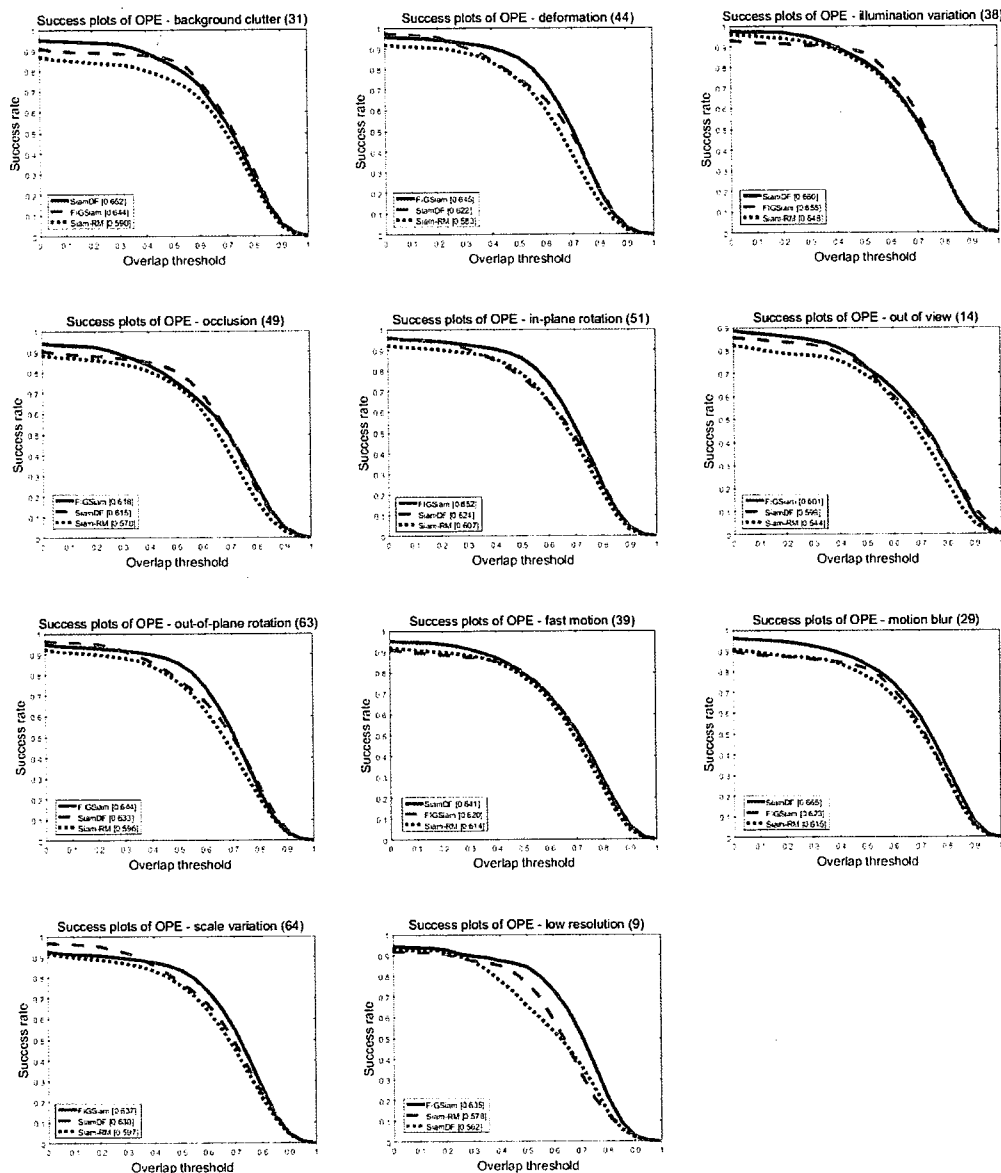


图6.4 OTB-2015数据集上11种属性下的实验结果（重叠率）

Figure 6.4 Tracking results on 11 challenging attributes on the OTB-2015 dataset (OR)

6.2 VOT 数据集对比实验分析

(1) VOT-2016 数据集

本小节采用 VOT-2016 数据集对本文提出的三种算法进行对比, 表 6.1 展示

出了三种算法在 VOT-2016 数据集上的测试结果，其中红、蓝、绿三种颜色分别对应跟踪性能排名第一、第二和第三的算法。

表 6.1 VOT-2016 数据集上三种算法的结果

Table 6.1 Proposed tracking methods results on VOT-2016 dataset

跟踪器	EAO	Accuracy	Failure
Siam-RM	0.3078	0.498	14.511
SiamDF	0.3365	0.568	12.181
FIGSiam	0.3411	0.576	16.152

从表中可以看出，FIGSiam 在 EAO 与准确度指标的表现上排名第一。这充分说明，FIGSiam 中扩充的正样本对、前景信息引导、填充损失和基于改进特征信息融合的模板更新方法能够让算法在精度方面具有更加优越的表现。然而，在平均失败次数方面，SiamDF 和 Siam-RM 分别位居第一和第二。该结果表明，多层特征自适应融合方法能够明显提升算法的鲁棒性，再检测孪生卷积神经网络虽然也能达到同样的效果，但仍需在网络结构和训练方式上进行改进。为了更加清晰地对比三种算法的跟踪性能，在图 6.5 中对三种算法的 EAO 排序进行了展示。

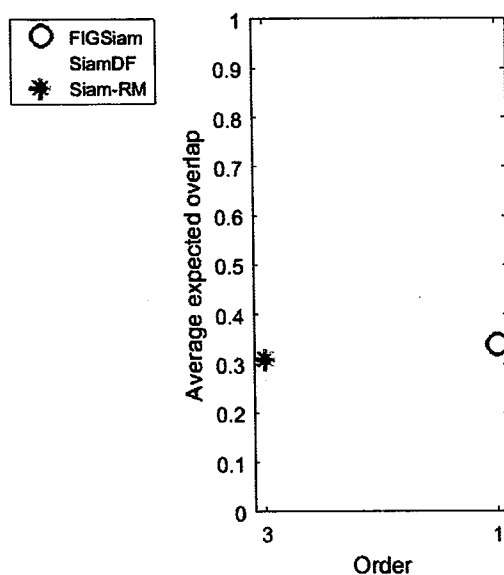


图 6.5 三种跟踪算法在 VOT-2016 数据集上的 EAO 结果

Figure 6.5 The EAO scores of proposed tracking algorithms on VOT-2016 dataset

(1) VOT-2018 数据集

为了更加客观地对本文提出的三种算法进行对比，本小节采用 VOT-2018 数

数据集对本文提出的三种算法进行对比，表 6.2 展示出了三种算法在 VOT-2018 数据集上的测试结果，其中红、蓝、绿三种颜色同样分别对应跟踪性能排名第一、第二和第三的算法。

表 6.2 VOT-2018 数据集上三种算法的结果

Table 6.2 Proposed tracking methods results on VOT-2018 dataset

跟踪器	EAO	Accuracy	Failure
Siam-RM	0.2806	0.492	14.530
SiamDF	0.3247	0.502	12.793
FIGSiam	0.3386	0.511	17.632

表中结果再次表明 FIGSiam 算法在精度方面表现更优，多层特征自适应融合与再检测机制的引入能够明显提升算法的鲁棒性。为了更加清晰地对比三种算法的跟踪性能，在图 6.6 中对三种算法的 EAO 排序进行了展示。

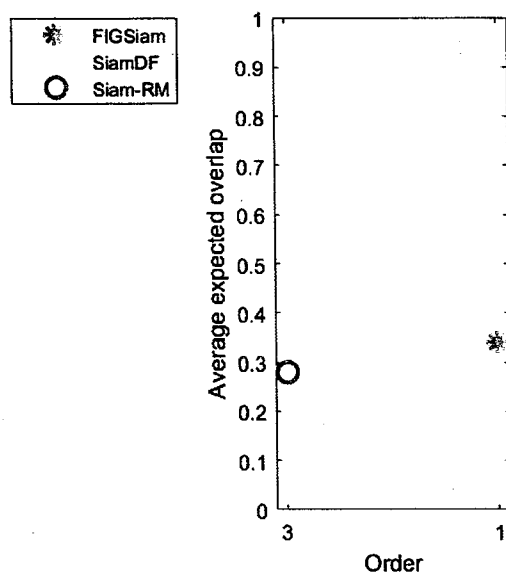


图 6.6 三种跟踪算法在 VOT-2018 数据集上的 EAO 结果

Figure 6.6 The EAO scores of proposed tracking algorithms on VOT-2018 dataset

6.3 实际应用场景对比实验分析

为了有效验证本文提出的三种算法在实际应用场景中的跟踪性能，本小节利用无人机航拍获取的彩色视频数据 UAV123^[119]及地基光测设备采集的视频数据进行实验分析，从而进一步明确基于深度特征匹配的目标跟踪算法适用的场景与

对象。

(1) 无人机航拍场景

近些年, 无人机 (Unmanned Aerial Vehicle, UAV) 已经被广泛应用于情报侦察、武装打击和城市反恐等尖端领域。其中, 目标跟踪作为 UAV 的核心技术之一, 往往在对地移动目标跟踪、对空高机动目标跟踪及海面复杂背景下目标跟踪等任务中发挥着重要作用。为了验证本文提出的三种目标跟踪算法在无人机航拍场景下的跟踪性能, 本小节利用无人机航拍获取的彩色视频数据进行实验分析。

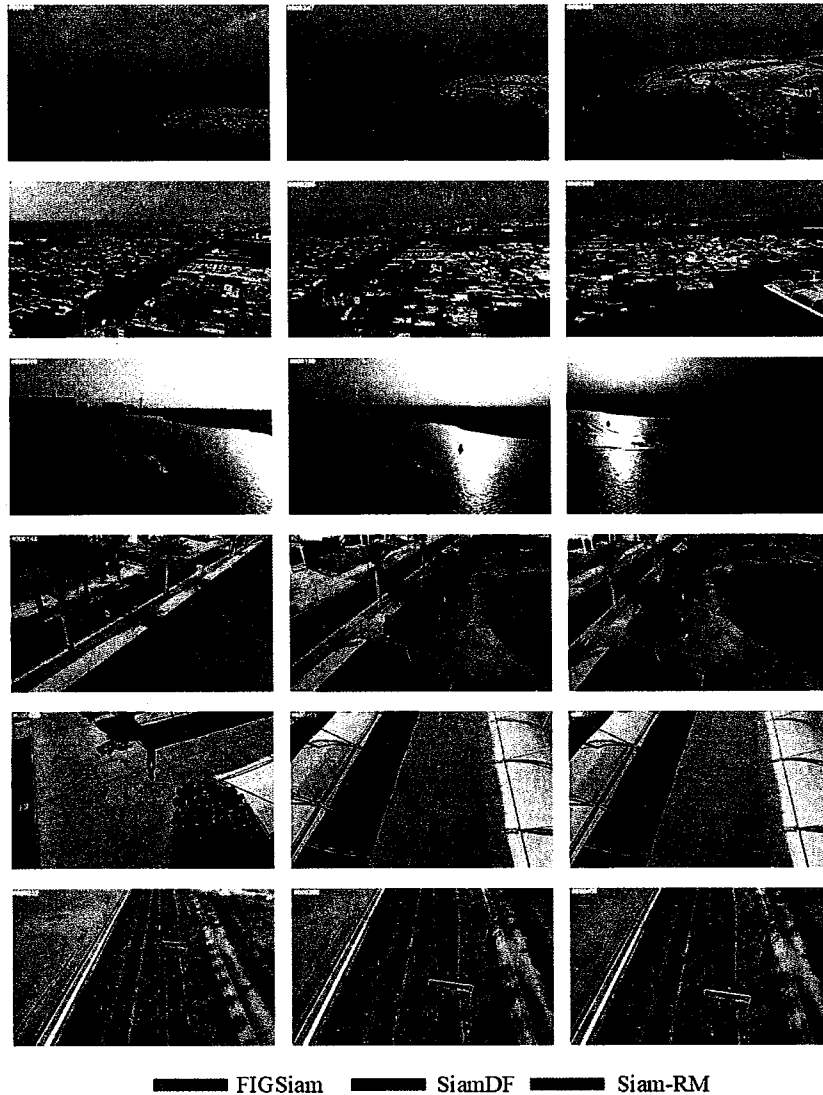


图 6.7 无人机航拍数据上的部分对比实验结果

Figure 6.7 Visualization of the tracking results in UAV sequences

图 6.7 展示了无人机航拍数据上的部分对比实验结果。从前三组实验结果可以看出,本文提出的三种目标跟踪算法能够较好地应对目标形变、背景杂乱、光照变化等挑战因素,充分证明本文提出的算法能够较好地适应无人机航拍场景,可用于空对地、空对海等场景下的目标跟踪任务。虽然,这三种算法在大部分的无人机航拍场景下能够获得较为稳定的结果。但是,从后三组实验结果可以看出,当目标遇到超严重遮挡时,三种算法的跟踪效果都会明显下降。尤其是在最后一组实验结果中,当目标几乎被完全遮挡后再次出现在视场中时,FIGSiam 算法误将背景中的相似目标作为跟踪目标,SiamDF 和 Siam-RM 算法直接丢失目标。该实验结果证明,本文提出的三种算法虽然从再检测机制、网络结构、引导训练和模板更新等方面进行改进,但仍需要提升算法在实际场景中应对超严重遮挡、多语义背景干扰等难题的能力。

(2) 地基光测场景

地基光测设备主要应用于我国各导弹、卫星测控站和试验训练基地,在保证实时获取高质量景象数据的同时,还需要进行高精度自动跟踪。目前,随着观测目标种类的日益增多,各种动态、多变的目标特性给地基光测场景下目标跟踪算法的研究带来非常严峻的考验。为了验证本文提出的三种目标跟踪算法在地基光测场景下的跟踪性能,本小节利用长春光机所地基光测设备获取的彩色视频数据进行实验分析。

图 6.8 展示了地基光测数据上的部分对比实验结果。从实验结果中可以看出,本文提出的三种目标跟踪算法在多数场景下能够较好地应对目标形变、相似目标干扰、背景杂乱等挑战因素,充分证明本文提出的算法能够适应地基光测场景,可用于靶场地对空场景下的目标跟踪任务。虽然,这三种算法在一些地基光测场景下能够获得较为稳定的结果。但是,从后三组的部分实验结果可以看出,当目标尺寸发生较大变化或遇到超严重遮挡时,三种算法的跟踪效果都会明显下降。在第二组和第三组实验结果中,当目标尺寸没有发生较大变化时,三种算法能够稳定地跟踪目标。然而,当目标尺寸变大时,三种算法的跟踪结果与真实目标的重叠率明显降低。这说明三种算法采用有限的尺度变换因子并不能较好地满足算法高重叠率的要求,需要在算法中融入更准确、更鲁棒的多尺度计算策略,从而

在复杂背景下更好地应对尺度变化带来的影响。在最后一组实验结果中,当目标被相似目标完全遮挡并再次出现在视场中时,三种算法都误将相似目标作为真实的跟踪目标。该实验结果表明,本文提出的三种算法在应对实际场景中的超严重遮挡、多语义背景等难题时仍具有明显的不足,需要进一步对网络结构、模板更新方法进行改进与提升。

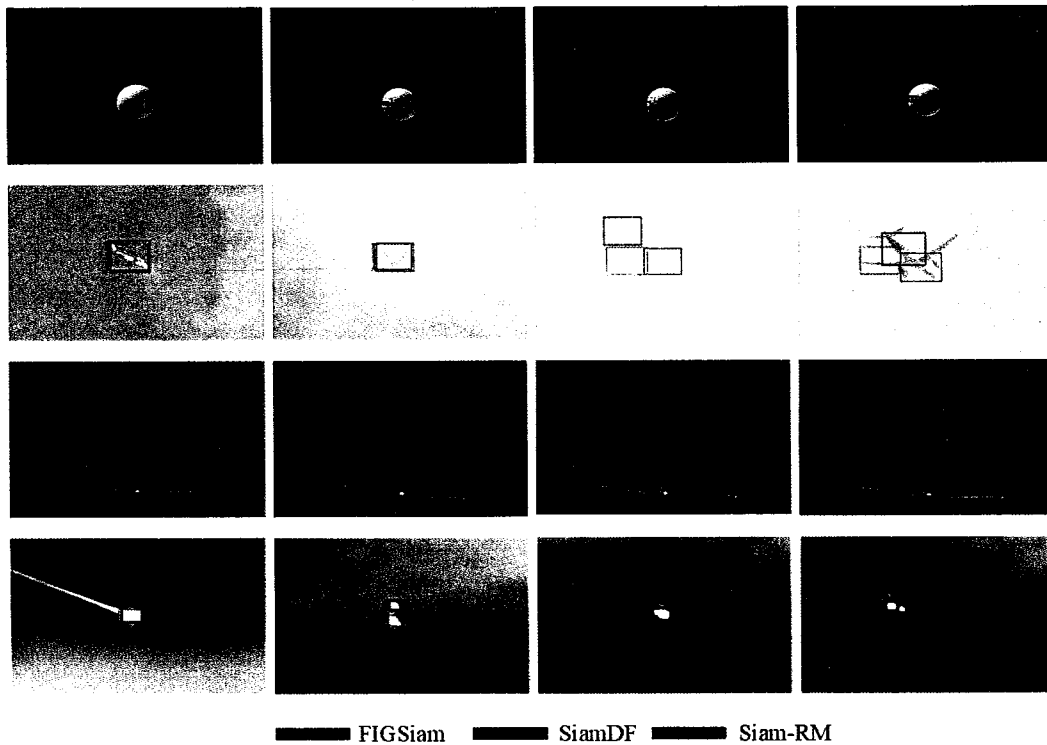


图 6.8 地基光测数据上的部分对比实验结果

Figure 6.8 Visualization of the tracking results in ground-based measurement sequences

6.4 运行速度分析

通过在 OTB 数据集、VOT 数据集和实际应用场景采集的视频数据中进行测试,本文提出的三种算法在测试中的平均运行速度如下表所示。

表 6.3 本文提出的三种算法平均运行速度

Table 6.3 The average running speed of the proposed algorithms

跟踪算法	平均运行速度(fps)
Siam-RM	24
SiamDF	32
FIGSiam	35

值得注意的是，三种算法在测试时的 PC 机配置均为：Intel i7-9800X CPU (3.80GHz), 64GB RAM 和一块 NVIDIA TITAN RTX GPU。从表中数据可以看出，Siam-RM 算法具有最低的运行速度，SiamDF 算法较比 Siam-RM 算法有所提升。与 Siam-RM 算法中基于生成式模型的自适应模板更新方法不同，基于特征信息融合的模板更新方法能够直接对模板特征进行操作，不再需要建立复杂的模型，因此会减少一定的计算量，使得算法的运行速度得到提升。相比之下，FIGSiam 算法具有更快的运行速度。这主要是因为 FIGSiam 算法中采用了基于改进特征信息融合的模板更新方法，该方法采用全局平均池化对特征进行降维处理，从而减少了特征信息融合过程中的计算量，有效提升了算法的运行速度。

6.5 本章小结

本章采用 OTB 数据集、VOT-2016 数据集和 VOT-2018 数据集客观地对本文提出的 Siam-RM、SiamDF 和 FIGSiam 三种算法的性能进行对比分析。实验结果充分证明，FIGSiam 算法中扩充的正样本对、前景信息引导、填充损失和基于改进特征信息融合的模板更新方法能够使算法在综合性能方面表现最优。SiamDF 算法在深层网络结构下采用多层特征融合策略对不同网络层级中的卷积特征进行自适应融合，会明显提升算法在复杂场景下的跟踪性能，从而在应对背景杂乱、光照变化、快速运动和运动模糊等挑战性因素时更加稳定可靠。Siam-RM 算法中再检测机制的引入能够使算法在一些低复杂度的场景下具有较高的精度，但与其它两种具有深层网络结构的算法相比，其在复杂场景下具有明显的劣势。为了验证本文提出的三种算法在实际应用场景中的跟踪性能，本章利用无人机航拍获取的彩色视频数据及地基光测设备采集的视频数据进行实验分析，进一步明确了基于深度特征匹配的目标跟踪算法适用的场景与对象，使本课题的研究意义与应用价值更加凸显。

第7章 总结与展望

7.1 论文工作总结

作为计算机视觉领域十分重要的组成部分和关键技术之一,目标跟踪在近些年成为了一个备受关注的研究课题。虽然,随着大数据和人工智能技术的发展,目标跟踪已经在智能交通、智能视频监控、视觉导航、人机交互、军事观测和医疗诊断等领域得到了广泛的应用。但是,在目标跟踪的发展道路上依然面临很多难题。一方面,在跟踪过程中目标经常会发生较为复杂的变化,比如光照变化、形变、遮挡、运动模糊等问题,这使得稳定且准确地跟踪目标变得尤为困难。另一方面,如何在复杂场景下全面且有效地提升跟踪器的性能也具有非常重要的研究意义。

近年来,基于深度学习的目标跟踪算法相比于其它目标跟踪算法展现出更加优异的性能,其在大数据的驱动下利用端到端学习不仅能够高效便捷的训练模型,还极大提升了模型的鲁棒性与准确性。其中,基于深度特征匹配的目标跟踪算法以其均衡的准确性、鲁棒性和实时性获得了较多的关注。虽然这类算法在很多方面表现优异,但由于网络结构中缺少层级间的自适应特征融合、模板更新策略不够完善、缺少再检测机制等原因,其在复杂场景下的鲁棒性与准确性都会明显降低。因此,本论文在研究大量国内外目标跟踪算法的基础上,针对复杂场景下基于深度特征匹配的目标跟踪算法的局限性进行一定的改进。本文主要的研究内容与取得的创新性成果总结如下:

(1) 提出一种结合再检测机制与自适应模板更新的孪生卷积神经网络目标跟踪算法。该算法利用一种高效且准确的再检测孪生卷积神经网络作为再检测机制,以此来应对响应图中的多波峰。另外,该算法提出一种基于生成式模型的自适应模板更新方法。该模板更新方法将评估过的可靠跟踪结果进行特征提取与概率统计,利用统计结果与已有目标模板特征进行自适应融合,从而实现模板的自适应更新。仿真实验结果表明,提出的算法能较好应对相似背景干扰、遮挡、目标形变等难题,有效提升复杂场景下的跟踪精度与鲁棒性。

(2) 提出一种结合深层特征与鲁棒特征融合的孪生卷积神经网络目标跟踪算法。该算法利用一种改进的残差网络来替代传统的浅层网络,从而提取出具有丰富语义信息的深层特征。与此同时,在算法中提出一种高效的自适应多层特征融合策略,有效提升了响应图的质量,减少了在复杂场景下相似背景干扰所带来的影响。另外,该算法利用一种自适应特征信息融合方法来实现模板的更新,从而能够适当减少模板更新过程中的计算量,有效克服遮挡、目标形变等干扰因素的影响。仿真实验结果表明,提出的算法能够有效提升跟踪效果,并且在准确性与鲁棒性等方面表现良好。

(3) 提出一种前景信息引导的孪生卷积神经网络目标跟踪算法。该算法采用扩充的正样本对来增强算法的鲁棒性。并且,为了进一步提升算法在语义背景下的识别能力,该算法将背景信息进行填充遮挡,从而增强前景信息的显著性。与此同时,将处理后的数据输入到基于卷积神经网络的引导分支中,并运用一种填充损失计算方式来改进算法的损失函数,有效提升算法在语义背景下的识别能力。此外,该算法采用一种改进的特征信息融合方法来实现模板的更新,进一步应对目标的外观变化并提升算法的跟踪精度与鲁棒性。仿真实验结果表明,提出的算法在多种干扰因素下取得了较好的跟踪效果,优于所对比的其它多个先进的跟踪算法。

本论文提出的三种算法从再检测机制、网络结构、引导训练和模板更新等方面进行改进,不仅有效提升基于深度特征匹配的目标跟踪算法的跟踪性能,还加深了跟踪领域的研究,为该领域的相关研究提供理论参考和有效支撑。

7.2 研究展望

尽管本文对复杂场景下基于深度特征匹配的目标跟踪算法进行了一些改进,但依旧无法达到目标跟踪期望的要求。为了进一步提升复杂场景下基于深度特征匹配的目标跟踪算法的跟踪性能,接下来将从如下几个方面进行研究与完善:

(1) 本论文提出的三种改进算法从再检测机制、网络结构、引导训练和模板更新等方面进行改进。虽然,提出的算法使跟踪性能得到了有效提升,但仍然存在一些局限性,例如,单一尺度的再检测网络并不能较好地满足算法高重叠率

的要求,需要在网络训练时融入多尺度的训练样本或在推理时融入多尺度的搜索策略;有限的尺度变换因子并不能较好地应对跟踪过程中目标的尺寸变化,需要在算法中融入更准确、更鲁棒的多尺度计算策略;采用的模板更新策略比较传统,需要进一步结合深度学习以应对更严峻的挑战等。

(2) 本论文提出的算法都是针对短期跟踪任务,而现实中大部分跟踪任务都需要具备长期跟踪能力。与短期跟踪任务不同,长期跟踪需要算法具备良好的记忆能力和较大的搜索区域。然而,在跟踪算法中引入记忆模块会增加算法的复杂度,扩大跟踪算法的搜索区域会使算法的计算负担明显加重。目前,随着目标检测算法日趋成熟,其不仅在深度学习的帮助下极大地提升了检测性能,还有效提升了检测速度并达到了实时检测的需求。因此,如何将目标跟踪算法与目标检测算法进行结合,从而有效提升跟踪算法的长期跟踪能力,仍是一个十分重要且极具研究意义的方向。

(3) 本论文主要开展单目标跟踪算法的研究,而现实场景中通常需要稳定跟踪多个目标。与单目标跟踪算法不同,多目标跟踪算法不仅需要实时检测出当前帧中的所有目标,还要针对前后帧的目标信息建立数据相关性,这无疑会在时间和空间两个方面增加开销。因此,如何将单目标跟踪算法进行改进,构建具有良好跟踪性能的多目标跟踪算法,仍然是一个值得关注的研究方向。