Robust and Accurate Generic Visual Object Tracking Using Deep Neural Networks in Unconstrained Environments

by

Javad Khaghani

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Signal and Image Processing

Department of Electrical and Computer Engineering
University of Alberta

© Javad Khaghani, 2021

Abstract

The availability of affordable cameras and video-sharing platforms have provided a massive amount of low-cost videos. Automatic tracking of objects of interest in these videos is the essential step for complex visual analyses. As a fundamental computer vision task, Visual Object Tracking aims at accurately (and efficiently) locating a target in an arbitrary video, given an initial bounding box in the first frame. While the state-of-the-art deep trackers provide promising results, they still suffer from performance degradation in challenging scenarios including small targets, occlusion, and viewpoint change. Also, estimating the axis-aligned bounding box enclosing the target cannot provide the full details about its boundaries. Moreover, the performance of tracker relies on its well-crafted modules, typically consisting of manually-designed network architectures to boost the performance. In this thesis, first, a context-aware IoU-guided tracker is proposed that exploits a multitask two-stream network and an offline reference proposal generation strategy to improve the accuracy for tracking class-agnostic small objects from aerial videos of medium to high altitudes. Then, a two-stage segmentation tracker to provide better semantically interpretation of target in videos is developed. Finally, a novel cell-level differentiable architecture search with early stopping is introduced into Siamese tracking framework to automate the network design of the tracking module, aiming to adapt backbone features to the objective of network. Extensive experimental evaluations on widely used generic and aerial visual tracking benchmarks demonstrate the effectiveness of the proposed methods.

Preface

The research conducted towards this thesis is a collaboration between Javad Khaghani and Seyed Mojtaba Marvasti-Zadeh, which is led by professor Li Cheng at the University of Alberta.

Chapter 3 of this thesis is written based on the paper published as *Seyed Mojtaba Marvasti-Zadeh, Javad Khaghani, Hossein Ghanei-Yakhdan, Shohreh Kasaei and Li Cheng "COMET: Context-Aware IoU-Guided Network for Small Object Tracking." ACCV (2020)* where the first two authors contributed equally to the work as stated in the original paper.

Chapter 5 is provided based on the manuscript which will be published as *Seyed Mojtaba Marvasti-Zadeh, Javad Khaghani, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei "CHASE: Robust Visual Tracking via Cell-Level Differentiable Neural Architecture Search." BMVC (2021)*, and the first two authors have equal contribution for this research as shown in the publication.

Chapter 4 & the last part of chapter 3 present two visual object trackers participated in VOT-ST2020 & VisDrone-SOT2020 challenges, respectively. Both of these trackers are developed in collaborative projects between Javad and Mojtaba.

Acknowledgements

I would first like to express my sincere gratitude to my supervisor, professor Li Cheng, for his continuous support throughout my Master's studies. His guidance, advice, and feedback helped me to conduct this research successfully.

I would also like to thank the members of my thesis committee, Dr. Xingyu Li & Dr. Di Niu, for taking the time to read this thesis and provide comments and suggestions. I would also thank our collaborators, especially Seyed Mojtaba Marvasti-Zadeh. Moreover, I appreciate the help of *Industry Sandbox and AI Computing* (ISAIC) for providing high-performance GPUs for conducting my research.

Last but not least, I would like to thank my family for their support throughout my whole life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Visual Object Tracking is a fundamental computer vision task with the goal of predicting the state of a target in video frames, given the initial bounding box in the first frame. This problem has many applications in real world scenarios including intelligent vehicles, robotics, behavior analysis, human-computer interaction, and automatic surveillance [1, 2]. Over the years, *Visual Object Tracking* community tried to obtain robust & accurate predictions for the state of class-agnostic targets using efficient methods (i.e., ideally the trackers run beyond real-time). However, as shown in Fig. 1.1, tracking arbitrary objects in the wild contains several challenges including occlusion, small objects, and out-of-view just to name a few [3, 4]. On the other hand, the recent success of deep neural networks in different computer vision tasks including object classification [5], detection [6], and semantic segmentation [7] proves that these methods can achieve state-of-the-art performance in a wide range of applications, outperforming traditional computer vision methods with a large margin of performance gain.

Motivated by that, this thesis studies *Visual Object Tracking* using deep neural networks with a focus on developing robust & accurate trackers for unconstrained environments. The target is considered in generic and model-free settings. In other words, at the inference time, no prior knowledge about the class-agnostic target is available, except the bounding box fitted on that in the first frame of test video. Also,

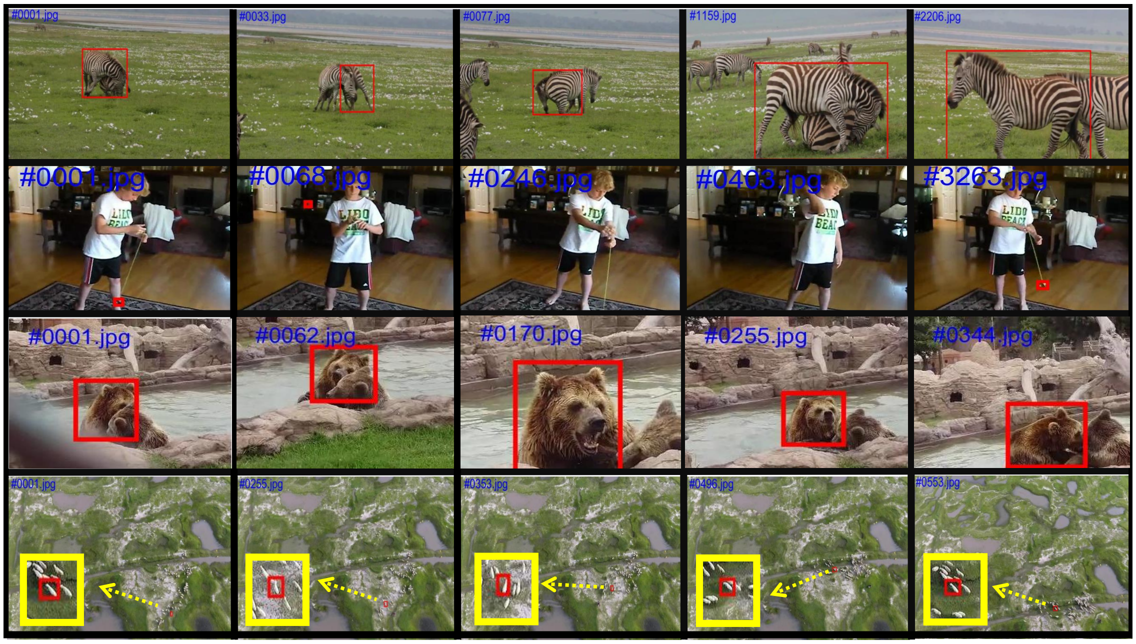

Figure 1.1: Example sequences from LaSOT [8], TrackingNet [9], and VisDrone-SOT2019 [10, 11] tracking benchmarks. With the bounding box of target in the first frame, the goal of visual tracking is to predict the box fitted on the object in the subsequent frames to the end. Considering the widely used *One-Pass Evaluation* (OPE) protocol, we do not re-initialize tracker after failure. For each frame, the red box and blue text show the ground truth box and frame number, respectively. The first (last) column shows the first (last) frame for each sequence, and the intermediate frames contain several tracking challenges including occlusion, fast motion, out-of-view, similar objects, small object, and camera motion. Note that the ground truth bounding box for frames with out-of-view challenge is not shown. For better visualization of the last sequence, the magnified window around the target is shown inside a yellow box. Best-viewed in zooming-in.

tracking is done in a casual manner, i.e., the tracker does not have access to future frames of video. Moreover, the focus of this research is on single object tracking in videos captured by single RGB camera.

Considering these assumptions, a family of the existing trackers learn an online classifier using the ground truth information at the first frame of video. Then, this classifier discriminates the target from background and other objects in next frames and they update the classifier through the video. Another category of methods use an offline trained Siamese deep neural network to localize the target at the test frames by measuring the similarity between the reference template and test patches. To have an unbiased evaluation of trackers as well as providing large-scale data for training deep trackers, several standard datasets and benchmarks (e.g., LaSOT [8], GOT-10k [12], and TrackingNet [9]) have been introduced. Accordingly, a comprehensive general background on widely used methods and benchmarks for *Visual Object Tracking* has been provided in chapter 2 of this thesis. Also, the literature exclusively related to each of the proposed methods has been only discussed in the corresponding chapter. While the recent trackers (see Sec. 2.1) have achieved high-quality results, their accuracy & robustness, notably for the challenging scenarios including small objects, low resolution, and similar objects, can be enhanced. Accordingly, to propose the COMET & LTCOMET (see chapter 3), DESTINE (see chapter 4), and CHASE (see chapter 5) trackers, *Visual Object Tracking* from the following aspects is studied:

1) Size of target: The size of objects can be medium/large or small. For small objects, latent representations extracted from deep layers of neural networks cannot provide enough information for appearance modeling. As a result, we should consider dedicated strategies to obtain robust & accurate model of small targets. Moreover, small object scenarios usually happen in aerial videos, i.e., videos mainly captured by UAVs, which introduce other challenges including frequent viewpoint change & extreme camera motions to the tracking problem. Following that, the *COMET: Context-Aware IoU-Guided Network for Small Object Tracking* tracker published in ACCV 2020 is proposed which mainly benefits from the multitask two-stream network and the offline reference proposal generation strategy. Also, a modified version of this tracker, named LTCOMET, has been developed to participate in VisDrone-SOT2020 challenge. These trackers are discussed in more details in chapter 3 of this thesis.

2) The predicted state of target: In *Visual Object Tracking*, predicting the axis-aligned bounding box enclosing the target is popular. However, this bounding box does not precisely represent the boundaries of object, notably for deformable ones with elongated parts (e.g., cats with long tail). Hence, *Visual Object Tracking* researchers seek to predict more accurate states of target, e.g., binary mask or rotated box instead of axis aliened box. Predicting the mask of target`s` in frames of video, given the `mask`

for the first frame is studied in another computer vision task, called semi-supervised *Video Object Segmentation* (VOS). Following this trend in tracking community, the two-stage *Adaptive Visual Tracking and Instance Segmentation* (DESTINE) tracker is developed to participate in the famous VOT-ST2020 challenge. This segmentation tracker is explained in chapter 4.

3) Procedure for designing the neural network: Visual tracking community attempts to obtain robust target representations using manually designed neural networks. However, the ability of deep neural networks to automatically learn robust feature extraction in an end-to-end manner is highly affected by the architecture of the network. Accordingly, obtaining well-performing deep trackers using manual architecture search is time consuming and depends on prior experience. As a result, automatic design of the best network architectures for visual trackers, considering the recent advances in *AutoML* and *Neural Architecture Search* (NAS) [13–15] is a promising research to explore. Motivated by this, the *CHASE: Robust Visual Tracking via Cell-Level Differentiable Neural Architecture Search* tracker is introduced which is accepted to BMVC 2021. This tracker incorporates the proposed cell-level differentiable architecture search mechanism with early stopping to adapt backbone features to the objective of Siamese tracking networks. A detailed explanation on CHASE is included in chapter 5.

Finally, a summary of the proposed trackers and future directions is provided in chapter 6. As one of the main future works, we have done unpublished studies on animal tracking and applications of computer vision for animal studies. As a result of this research, a zebrafish larvae segmentation tracking dataset with zebrafish-specific visual attributes will be provided, in collaboration with *Guan Zhen* from the *Institute of Molecular and Cell Biology* (IMCB), A*STAR. Also, a comprehensive literature review on computer vision tools for animals will be submitted. Please refer to section 6.2 for more information on these projects.

Chapter 2

Background

In this chapter, a general background on the existing literature for *Visual Object Tracking* using deep neural networks is provided. However, the dedicated related literature for each proposed tracker has been exclusively reviewed in the corresponding chapter. This chapter is structured as follows. First, in Sec. 2.1, the existing methods for generic single object tracking in RGB videos using online classification-based and Siamese-based networks are discussed. Then, an overview of the existing tracking datasets and evaluation metrics is provided in Sec. 2.2.

2.1 Visual Tracking Methods

Online classifier-based and Siamese network-based trackers are the main two approaches for single object visual tracking using deep convolutional neural networks.

2.1.1 Online Classifier-Based Trackers

Some of the well-performing trackers efficiently learn an online classifier based on the information of target provided in the first frame of test video (i.e., no offline training required) to discriminate the target from background and distractors in the next frames. Through the video, the classifier is refined based on the prediction of previous frames [16]. MDNet [17] consists of a shared sub-network with sequence specific classifier branches. During offline training, the shared sub-network along with

sequence specific branches are trained using video sequences. In online tracking, they train the new classifier branch which receives input from the pre-trained shared part using the test video. SANet [18] uses *Recurrent Neural Networks* (RNNs) to integrate the structural information of target into the MDNet pipeline. ATOM [19] considers an online classification network & a two stream network for robustly localizing the target & accurately estimating its scale, respectively. The classification network consists of two convolutional layers which receive backbone features from `Block4` of ResNet-18 [20]. The convolutional layers are trained efficiently by integrating conjugate gradient computing into PyTorch's autograd tools to predict a response map by solving Gauss-Newton problem in online tracking phase. They also incorporate pre-existing practical techniques (e.g., hard negative sampling for handling distractors) to robustly track the target. To estimate the scale of target, they generate object proposals based on dimension of target in the previous frame and prediction of online classifier at the current frame. Then, they refine the proposals using an integration of IoU-Net [21] into a Y-shape network; leveraging the proposed IoU modulation vector to encode target related information of reference frame into test branch. The incorporated two stream IoU-Net network is trained on large-scale datasets in offline phase, while the classification network will be learned in online tracking phase using the test video.
As a major subset of online classifier-based trackers, *Discriminative Correlation Filter* (DCF)-based trackers [22] learn a template model in online manner using a non-linear ridge regression problem which is solved using circular correlation. Despite their robustness & efficiency, they suffer from accuracy limitations as they usually use a simple and rapid multi-scale search to estimate the scale of target.

2.1.2 Siamese Network-Based Trackers

Siamese networks are Y-shape networks which learn to measure the similarity between a reference image (i.e., exemplar image) with the candidate target regions in the test frame (i.e., search regions), resulting in a dense response map for target localization.

These networks provide a good trade-off between performance and computational complexity. First, they extract target-specific features for tracking by transferring powerful backbone features to the objective of tracking using lightweight modules. To limit the number of parameters, the parameters of lightweight modules are shared between the reference & test branches. Then, to measure the similarity between template & search features, regular or depthwise cross-correlation between the latent representations of exemplar and search regions are widely used [23–25]. The response map can only classify the target from background and similar instances, not providing enough information about the dimensions of object. As a result, we also need an scale estimation part which usually considers smooth scale variations in adjacent frames. For instance, the previous works incorporate bounding box regression using *Region Proposal Networks* (RPN) [23, 24], refining the bounding box from previous frame [21, 26], or a simple multi-scale search [27]. Siamese-based trackers are trained offline on pairs of samples, benefiting from large-scale image [28–30] & video-based (see Sec. 2.2) datasets. However, during online tracking phase, they usually consider a static template model based on the first frame and localize the target at the current frame without any online fine-tuning. Accordingly, these networks are usually good at accurately estimating the scale of target, while they suffer from robustness issues in presence of challenging scenarios including distractors & similar objects and novel object classes due to the lack of online adaptation for most of them.

As one of the first Y-Shaped networks, *Generic object Tracking using regression networks* (GoTurn) [31] directly regresses the whole box of target for the test frame. *Fully Convolutional Siamese networks* (SiamFC) [27] introduces tracking as similarity learning between the reference and test branches to obtain the position of target. Later, CFNet [32] introduced a closed-form solution to train a convolutional filter in an end-to-end manner, exploiting both correlation filters and Siamese advantages. Following the efforts in improving the performance of Siamese trackers, *Siamese Region Proposal Network* (SiamRPN) [23] formulates tracking as local one-shot learning by integrating RPN [33] into two-stream networks for bounding box refinement, resulting in better target estimation. Many later Siamese trackers consider this method as their baseline and introduced new modules and strategies for enhancing the performance [24, 25, 34–36]. DaSiamRPN [34] exploits semantic backgrounds, distractor suppression, and local-to-global search window towards learning robust features, online adaptation, and handling long-term scenarios. SiamDW [35] designs deeper and wider networks by considering various units and backbone networks to benefit from state-of-the-art network architectures. *Siamese Cascaded RPN* (CRPN) [36] consists of multiple RPNs that perform stage-by-stage classification and localization. SiamRPN++ [25] proposes a ResNet-based Siamese tracker that exploits layer-wise & depthwise aggregations and uses spatial-aware sampling strategy to train a deeper network. SiamMask [24] tracker develops a two-stage segmentation tracker by integrating class agnostic binary segmentation to Siamese architecture.

Recently, researchers have started to introduce online learning into Siamese framework to increase the robustness of tracking novel targets against distractors. DaSiamRPN[34] incorporates incremental learning using a distractor-aware module for inference time. DSiam [37] utilizes a fast transformation learning module for learning the target appearance & suppressing the background in online phase. UpdateNet [38] learns to update the template model for Siamese trackers based on a combination of the template model from first frame, current frame, and accumulated template. GradNet [39] uses the discriminative information in gradient to update the template model.

DiMP [26] tracker introduces a model predictor, consisting of filter initializer & optimizer into the Siamese framework for target classification. First, they use a single 3×3 convolutional block to transfer the appearance features extracted from ResNet [20] backbone for tracking purpose for both reference and test branches. Then, they use a 4×4 PrROI pooling [21] layer over the latent representation from reference branch to obtain an initialization of the template model. After that, they introduce a filter optimization step to adapt this initial template model to the appearance of new targets in a few-shot settings using a learned discriminative loss. They measure the similarity between the obtained template model and features from test branch using cross-correlation, resulting in a score map. DiMP uses the same IoU-Net [21]-based network as ATOM [19] for target estimation. Finally, they train the whole network consisting of Y-shape classification and target estimation networks in an end-to-end manner using a multi-task discriminative loss function. Hence, we can consider DiMP as a deep Siamese- & DCF-based method. During online tracking, they use the filter optimization step considering previous online learning strategies in tracking literature (e.g., hard negative sampling). In this way, they can use the information available in large-scale datasets in offline phase and adapt the model to the appearance of new target for tracking in a few-shot manner. Moreover, the ground truth boxes in large-scale tracking datasets might contain noisy samples, e.g., datasets consider different strategies for annotating deformable objects with elongated parts (e.g., animals with long tails). Pr-DiMP [40] introduces probabilistic regression into DiMP [26] pipeline to predict the conditional probability of target box and handle the noisy ground truth boxes in tracking benchmarks. They also introduce Super-DiMP tracker which combines the strength of DiMP [26], PrDiMP [40], and engineering techniques to increase the performance.

Object-aware anchor-free networks (Ocean) [41] tracker, proposed by Microsoft research group, is the first tracking method to leverage anchor-free approach for *Visual Object Tracking*. Anchor-based methods are trained using positive samples with high *Intersection over Union* (IoU) with the ground truth box in offline phase which results in drifting the tracker by error accumulation in online tracking. By leveraging the anchor-free idea, inspired from FCOS [42] detection method, Ocean can regress the full state of target in a single network and rectify weak predictions resulting in robust & accurate tracking. To further improve the robustness using online learning, they integrate the model prediction approach of DiMP [26] into their method, and introduce Ocean-Online tracker.

The idea of using spatial & channel attention mechanisms [43] to enhance the performance of trackers is previously used in a few works including our COMET [44]. SiamAttn [45] proposes *Deformable Siamese Attention* (DSA) module which leverages the cross-attention mechanism to learn the mutual information from both branches as well as the self-attention to highlight the target-related spatial & channel information. However, they require the depthwise cross-correlation to combine the information of reference and test branches. Very recently, researchers have started to integrate transformers into Siamese-like networks for *Visual Object Tracking*. *Transformer Meets Tracker* [46] introduces modified transformers (i.e., without fully-connected feed-forward layers) into Super-DiMP and SiamFC architectures, achieving promising performance gain while training for 50 epochs. They integrate the separated encoder & decoder of the modified transformer into the reference & test branches to reinforce the target-related features and consider pixel-wise temporal correspondence between branches. However, they still use cross-correlation to measure the similarity of reference and test branches. *Transformer Tracking* (TransT) [47] claims that the local linear matching nature of cross-correlation results in falling into local optimum and neglecting semantic information while those information is required for precise scale estimation. To address this issue, they integrate transformers into Siamese framework to fuse the latent representations of reference & test branches using attention mechanism, and predict the full state of target (i.e., position & scale) using a single network. While they obtain state-of-the-art performance, their method is trained for 1,000 epochs.

2.2 Visual Tracking Datasets

Providing standard benchmarks to fairly compare different visual trackers has a long history in literature of tracking [48–52]. By emerging the deep convolutional neural networks in tracking community [4], researchers started to release large-scale annotated video-based datasets for offline training of high-performance deep trackers. In this section, we will review the generic and aerial bounding box tracking datasets and benchmarks which are used for training and evaluation of the proposed methods of this thesis. Note that the datasets for segmentation tracking and *Video Object Segmentation* are discussed in related chapter (see Sec. 4.2). There is also a brief discussion on long-term tracking datasets in Sec. 3.2.3.

The utilized datasets in this thesis consist of GOT-10k [12], LaSOT [8], TrackingNet [9], NfS [52], UAV-123 [53], UAVDT [54, 55], VisDrone-SOT2019 [10, 11], and Small-90 [56]. For inference, all of them use the *One-Pass Evaluation* (OPE) protocol, i.e., the trackers is initialized with the bounding box of first frame and is run to the end of sequence without resetting after losing the target. The incorporated evaluation metrics for the performance are mainly precision & success, i.e., ratio of frames with *Center Location Error* (CLE) & *Intersection over Union* (IoU) between predictions and ground truths thresholded at predefined values (e.g., less than 20 pixels for precision & higher than 0.5 or 0.75 for IoU). Some benchmarks utilize normalized precision to address the dependency of regular precision metric on the scale of target and resolution of frame [8, 9]. The *Area Under Curve* (AUC) metric for success plot which is equivalent to computing the *Average Overlap* (AO) is also widely used. For evaluating the efficiency, *Frame Per Second* (FPS) is dominant.

GOT-10k [12]: This is a high-diversity large-scale short-term tracking benchmark with 10K videos and 1.5 million bounding boxes. On average, the length of each clip is 15 seconds. The training and test object classes have zero overlap (except for the *person* class) to evaluate the generalization of methods. To avoid human bias, 563 object classes and 87 motion forms based on the semantic hierarchy of WordNet have been used. They also provide motion classes, object visible ratios and absence indicators labels. To handle the class imbalance issues in evaluation, they consider the *mean Average Overlap* (mAO) and *mean Success Rate* (mSR) instead of original AO and SR metrics. The ground truth for the test set is private (except the first frame) and evaluation is done on 180 sequences.

LaSOT [8]: This is a challenging large-scale tracking benchmark with 1.4K long videos and 3.5 million frames which are manually annotated. The data is divided into the same number of sequences for each of 70 classes. On average, each video contains 2.5K frames with target disappearance/ reappearance (i.e., long-term) scenarios. In this thesis, we have used the official protocol II for splitting data into the training and test sets. Accordingly, the training set contains 1.12K videos with 2.83 million frames, and the test set consists of 280 sequences with 690k frames. Precision, success, and normalized precision are used as the evaluation metrics. They also annotate 14 visual attributes including *Illumination Variation* (IV), *Full Occlusion* (FOC), *Partial Occlusion* (POC), *Deformation* (DEF), *Motion Blur* (MB), *Fast Motion* (FM), *Scale Variation* (SV), *Camera Motion* (CM), *Rotation* (ROT), *Background Clutter* (BC), *Low Resolution* (LR), *Viewpoint Change* (VC), *Out-of-View* (OV) and *Aspect Ratio Change* (ARC).

TrackingNet [9]: This in-the-wild large-scale benchmark contains more than 30K videos and 14.4 million frames of 27 target classes which are gathered from YouTube-BB [29] and annotated automatically using tracker. They also provide 15 visual attributes. The test set includes 511 videos with private ground truth boxes (except for the first frame) and the same distributions of classes for training & test sets. They use precision, success, and normalized precision for evaluation.

NfS [52]: This dataset provides 100 annotated sequences with a high framerate of 240 *FPS*.

UAV-123 [53]: This is a challenging aerial view tracking dataset consisting of 123 videos, 113K frames, and 9 classes of targets captured from a low-altitude perspective. They have annotated bounding boxes as well as 12 visual attributes of *Background Clutter* (BC), *Fast Motion* (FM), *Aspect Ratio Change* (ARC), *Illumination Variation* (IV), *Camera Motion* (CM), *Out-of-View* (OV), *Partial Occlusion* (POC), *Full Occlusion* (FOC), *Scale Variation* (SV), *Low Resolution* (LR), *Similar Object* (SOB), *Viewpoint Change* (VC).

UAVDT [54, 55]: This UAV dataset contains 100 videos for object detection, single object tracking, and multiple object tracking. They annotate this data with 840K bounding boxes which are provided manually. For single object tracking, they provide a testing set containing 50 UAV sequences, without any training data. Success and precision are used as evaluating metrics. 10 visual attributes consisting of *Background Clutter* (BC), *Camera Rotation* (CR), *Object Rotation* (OR), *Small Object* (SO), *Illumination Variation* (IV), *Object Blur* (OB), *Scale Variation* (SV), *Large Occlusion* (LO), *Viewpoint Change* (VC), and *Fast Motion* (FM), 3 scene condition attributes (i.e., Weather Condition, Flying Altitude, and Camera View) , and a sequence duration attribute are also provided.

VisDrone-SOT2019 [10]: This dataset contains 167 videos with 189K frames & manually labeled bounding boxes, splitting into training (86 sequences), validation (11 sequences), and testing sets. Then the testing data is divided into two overlapping sets of test-dev (35 sequences) with released ground truth for development, and test-challenge (60 sequences including 25 long-term scenarios) with private boxes for the corresponding competition. The data is captured using various drone models under different weather and lighting conditions. Success and precision metrics are used for evaluating the performance. They have also annotated 12 visual attributes consisting of *Full Occlusion* (FOC), *Partial Occlusion* (POC), *Aspect Ratio Change* (ARC), *Background Clutter* (BC), *Fast Motion* (FM), *Camera Motion* (CM), *Illumination Variation* (IV), *Scale Variation* (SV), *Low Resolution* (LR), *Out-of-View* (OV), *Small Object* (SO), and *Viewpoint Change* (VC).

Small-90 [56]: Small-90 has incorporated small object sequences of various existing datasets including UAV-123 [53], OTB [48], and TC-128 [51].

Chapter 3

COMET

In this chapter, first, the challenges and motivations of tracking small objects from aerial videos are introduced. Then, the related works including small object detection, small object tracking, and long-term tracking are discussed. After that, the *COMET: Context-Aware IoU-Guided Network for Small Object Tracking* [44] method published in *Asian Conference on Computer Vision* (ACCV) 2020 is reviewed. Also, the extended version of the tracker, named LTCOMET [11], participated in *Vision Meets Drone- Single Object Tracking* (VisDrone-SOT) 2020 challenge [11] is introduced. Accordingly, some parts of this chapter are written based on the published COMET paper [44].

3.1 Introduction

Aerial *Visual Object Tracking* can be considered in two different scenarios: 1) Videos captured in low-altitudes to track medium or large objects in surveillance videos with limited viewing angle, and 2) Videos captured in medium- (30$\sim$70 meters) and high-altitudes (>70 meters) to consider tiny objects in videos mostly captured by UAV. As you see in Fig. 3.1, most objects in the first category (captured from low-altitude aerial views (10$\sim$30 meters)) are medium/large-sized and provide sufficient information for appearance modeling. However, the second one aims to track targets with low resolution involving complicated scenarios including tiny targets, drastic cam-

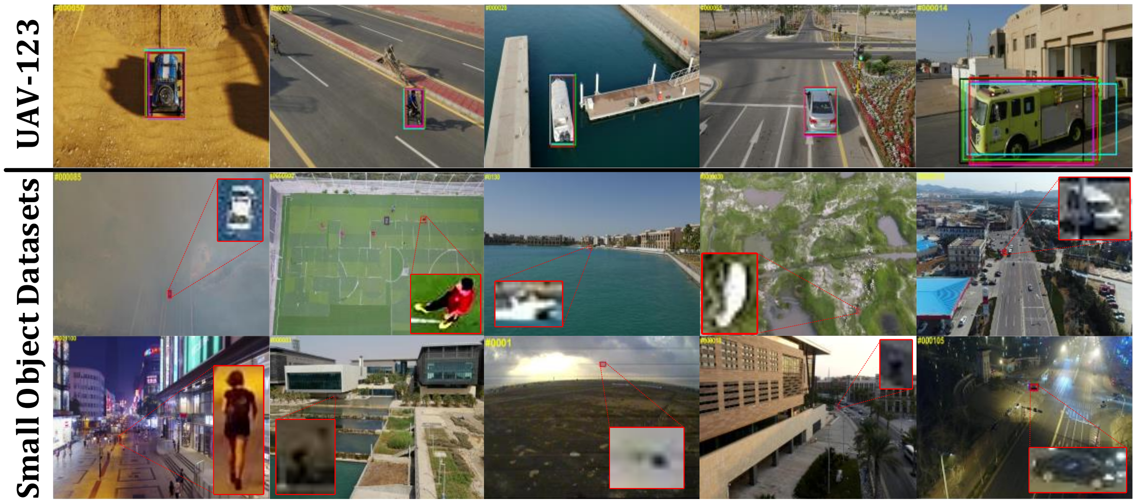

Figure 3.1: Examples to compare low-altitudes and medium/high-altitudes aerial tracking. The first row represents the size of most targets in UAV-123 [53] dataset, which captured from 10∼30 meters. However, some examples of small object tracking scenarios in UAVDT [54], VisDrone-2019 [10], and Small-90 [56] datasets are shown in last two rows.

era viewpoint changes & rotations, wide aerial view, and severe camera & objection movements. In most cases, it is arduous even for humans to track tiny objects in the presence of complex background as a consequence of limited pixels of objects.

The existing state-of-the-art generic visual trackers have satisfactory performance in the first scenario, while their results are not convincing for the later settings. Moreover, to the best of our knowledge at the time of our paper submission, there were no existing tracking research with convincing performance on recently released tiny object tracking benchmarks (i.e., VisDrone [10, 57], UAVDT [54, 55], and Small-90 [56]). Only a few DCF-based aerial trackers (discussed in the 3.2.2) exist which consider efficiency as the main objective by sacrificing the performance.

Motivated by this, we studied the problem of tracking a class-agnostic small target from aerial videos of medium to high altitudes. We attributed the performance degradation of state-of-the-art trackers on these videos to neglecting any special strategies to handle their novel challenges. Then, considering ATOM [19] tracker as the baseline method, we introduce our contributions into the scale-estimation network of ATOM [19] to narrow the gap between the state-of-the-art generic trackers & aerial ones.

Our COMET benefits from a two stream network architecture trained using multi-task loss function and an offline proposal generation strategy. The main performance gain of our approach results from multi-scale feature learning and attention modules to enhance target related information. Moreover, the offline proposal generation strategy from reference frame aims at helping network with providing context to generalize on object and its part during offline training. Finally, the multi-task loss helps the network to consider both accuracy & robustness during offline training.

The contributions of the paper are summarized as the following two folds.

1) Offline Proposal Generation Strategy: In offline training, the proposed method generates limited high-quality proposals from the reference frame. The proposed strategy provides context information and helps the network to learn target and its parts. Therefore, it successfully handles occlusions and viewpoint changes in challenging aerial scenarios. Furthermore, it is just used in offline training to impose no extra computational complexity for online tracking.

2) Multitask Two-Stream Network: COMET utilizes a multitask two-stream network to deal with challenges in small object tracking. First, the network fuses aggregated multi-scale spatial features with semantic ones to provide rich features. Second, it utilizes lightweight spatial and channel attention modules to focus on more relevant information for small object tracking. Third, the network optimizes a proposed multitask loss function to consider both accuracy and robustness.

Extensive experimental analysis are performed to compare the proposed tracker with state-of-the-art methods on the well-known small object benchmarks, namely UAVDT [54, 55], VisDrone-2019 [10, 11], and Small-90 [56]. The results demonstrate the effectiveness of COMET [44] for small object tracking. We also show that while the main focus of this research is on small objects, the proposed strategies can enhance the performance of baseline tracker on ground-view generic object tracking benchmarks [8, 12]. Moreover, we have extended our tracker to LTCOMET [11] by integrating strategies for long-term tracking & light enhancement and providing more augmentations into its pipeline to enhance the performance on the VisDrone-2019-test-challenge set.

3.2 Related Works

In this section, we will review the literature exclusively related to COMET by summarizing the existing efforts on small object detection, small object tracking, and long-term tracking. Please refer to the chapter 2 for general background information on *Visual Object Tracking*.

3.2.1 Small Object Detection

The literature for object detection introduces some strategies for enhancing the performance for small objects. *Single Shot multi-box Detector* (SSD) [58] uses different levels of backbone features depending on the size of the target. *Discriminative Single Shot Detector* (DSSD) [59] uses deconvolution layers to enhance the spatial resolution of deep features considering context information for small object detection. *Multi-scale Deconvolutional Single Shot Detector* (MDSSD) [60] enhances the performance of small object detection using multi-scale deconvolution fusion modules. Also, [61] utilizes multi-scale feature concatenation and attention mechanisms to enhance small object detection using context information. SCRDet [62] uses SF-Net and MDA-Net as feature fusion and attention module, respectively. Furthermore, other well-known detectors (e.g., YOLO-v3 [63]) exploit the same ideas, such as multi-scale feature pyramid networks, to alleviate their poor accuracy for small objects.

3.2.2 Small Object Tracking

Developing specific methods for small object tracking from aerial view is still in progress, and there are limited algorithms for solving existing challenges. Indeed, most of the existing trackers prioritize efficiency to performance by developing *Discriminative Correlation Filters* (DCF)-based trackers. To restrict the alternation rate of response map, *Aberrance Repressed Correlation Filter* (ARCF) [64] uses a cropping matrix and regularization term. *Boundary Effect aware Visual Tracker* (BEVT) [65] addresses the boundary effect issue of correlation filters to achieve a robust aerial tracker. They introduce a background learning strategy, learning the response map by comparing the scores of adjacent frames, and using multi-layer backbone features. *Keyfilter-aware* [66] tracker utilizes key-filters to prevent filter corruption and learn context information. To improve the quality of training set, *Time Slot-based Distillation* [67] (TSD) algorithm adaptively scores historical samples by a cooperative energy minimization function. It also accelerates this process by discarding low-score samples. AutoTrack [68] adaptively learns a spatio-temporal regularization term to avoid using the predefined parameters which is popular in correlation filters. The local spatial and global temporal terms help the tracker to focus on important object parts and update the learning rate, respectively. The performance of these DCF-based trackers are lower than the state-of-the-art trackers as they have focused on efficiency for deploying the trackers on UAVs.

3.2.3 Long-Term Tracking

In real-world scenarios, objects can have disappearance & reappearance in the scene through long videos. In short-term single object tracking, researchers usually consider a local window around the prediction of previous frame, and search this windows to find the target in the current frame. This assumption can be invalidated when object has very fast movements, full occlusion, out-of-view, or camera is moving. Compared with ground-view videos, these scenarios more frequently happen in videos captured by UAVs. Consequently, we need a global re-detection strategy when these scenarios happen.

DaSiamRPN [34] introduces an iterative local-to-global search by enlarging the search window with a constant step size once the predicted target score is low. This simple strategy helps DaSiamRPN to run at 110 FPS on long-term videos. SPLT [69] introduces a real-time long-term tracker using perusal and skimming modules for local tracking and global re-detection, respectively. Perusal consists of a SiamRPN [23] to generate local proposals and a verifier to measure the cosine similarity between the feature embeddings of target instance and generated proposals. They switch to global search if the highest measured similarity is lower than a threshold. Then, skimming module is responsible for rapidly selecting the best global proposals from a large set of sliding windows. Short-term trackers consider strong priors on smoothness of changes in position & scale of target in nearby frames. To have a multi-scale target search over the whole frame without such strong priors, GlobalTrack [70] uses a two-stage object detector guided by the target instance. Siam R-CNN [71] integrates faster R-CNN into Siamese framework to develop a tracking by re-detection method. Also, they propose a *Tracklet Dynamic Programming Algorithm* (TDPA) which leverages the initial template and target box in previous frame for obtaining a robust long-term tracker. Besides, they introduce a hard example mining strategy to obtain negative samples conditioned on reference target from other videos. Long-term trackers are mainly developed by integrating global re-detection into existing short-term trackers. Short-term trackers use the discriminative cues obtained from confidence map to decide when and how to update the tracker in online phase. However, the experiments show that the predicted response map is not always reliable. Due to the high uncertainty in long-term videos, researchers prefer to build their long-term trackers based on short-term ones that do not consider any online learning. To address this issue, [72] introduces a meta-updater implemented using cascaded LSTM to combine the sequence of geometric & appearance cues with the widely used discriminative ones to make decisions for online updating of short-term trackers. Moreover, they introduce a complete long-term tracking pipeline consisting of local tracker (ATOM [19] and SiamMask [24]), proposed meta-updater, global detection (faster R-CNN [33]), proposal generation (SiamRPN [23]), and verifier (RTMDNet [73]) methods.

Researchers use the famous *Visual Object Tracking- Long-term Tracking* (VOT-LT) [3, 74, 75] challenge to introduce & evaluate their developed long-term trackers before publishing them as regular papers. Moreover, some standard datasets including OxUvA [76] and TLP [77] for long-term tracking are introduced. Besides, the LaSOT [8] dataset contains long videos which are widely used to evaluate the robustness of short-term trackers in challenging scenarios. On the other hand, due to the importance of long-term scenarios in aerial tracking, previous works have introduced UAV20L [53] & VisDrone-test-challenge [10, 11] datasets for this task. While UAV20L is a subset of UAV123 [53] with publicly available ground truth boxes, the VisDrone-test-challenge [10, 11] set with private ground truth boxes has been used to evaluate the state-of-the-art aerial trackers in VisDrone competition. Later in this chapter (see Sec. 3.4.4), we will introduce the long-term extension of COMET participated in VisDrone-SOT2020 [11], and compare its performance to the other participants.

3.3 Method

A key motivation of COMET is to solve the issues discussed in Sec. 3.1 by adapting small object detection schemes into the network architecture for tracking purposes. The graphical abstract of proposed offline training and online tracking is shown in Fig. 3.2. The proposed framework mainly consists of an offline proposal generation strategy and a two-stream multitask network, which consists of lightweight individual modules for small object tracking. Also, the proposed proposal generation strategy helps the network to learn a generalized target model, handle occlusion, and viewpoint change with the aid of context information. This strategy is just applied to offline training of the network to avoid extra computational burden in online tracking.

3.3.1 Offline Proposal Generation Strategy

The eventual goal of proposal generation strategies is to provide a set of candidate detection regions, which are possible locations of objects. There are various category-dependent strategies for proposal generation [21, 58, 78]. For instance, IoU-Net [21] augments the ground truth instead of using *Region Proposal Networks* (RPNs) to

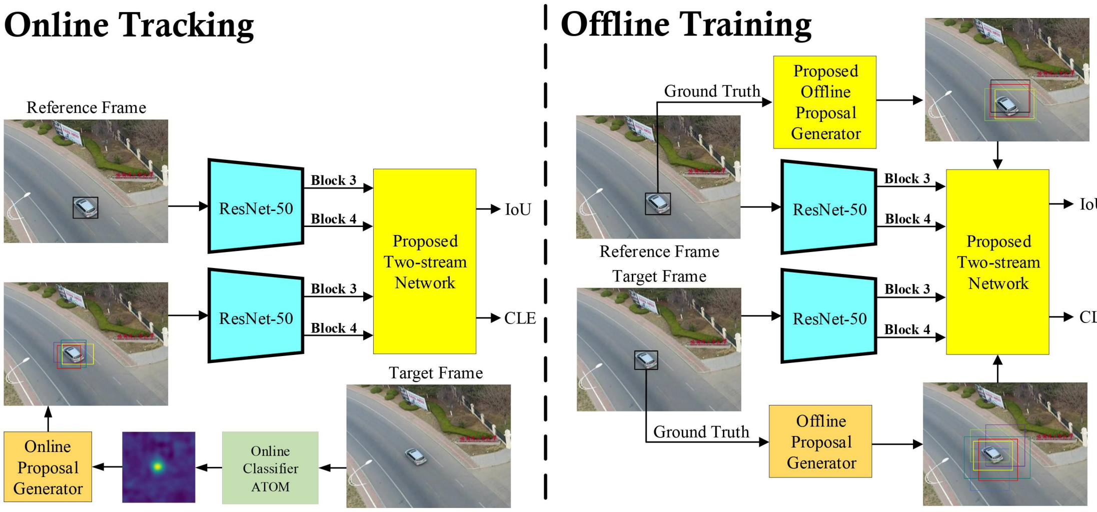

Figure 3.2: Overview of the proposed method (COMET) in offline training and online tracking phases.

provide better performance and robustness to the network. Also, ATOM [19] uses a proposal generation strategy similar to [21] with a modulation vector to integrate target-specific information into its network.

Motivated by IoU-Net [21] and ATOM [19], an offline proposal generation strategy is proposed to extract the context information of target from the reference frame. The ATOM tracker generates N target proposals from the test frame ($\mathcal{P}_{t+\zeta}$), given the target location in that frame ($\mathcal{G}_{t+\zeta}$). Jittered ground truth locations in offline training produce the target proposals. But, the estimated locations achieved by a simple two-layer classification network will be jittered in online tracking. The test proposals are generated according to $IoU_{Gt+\zeta} \triangleq IoU(\mathcal{G}_{t+\zeta}, \mathcal{P}_{t+\zeta}) \geqslant \mathcal{T}_1$. Then, a network is trained to predict IoU values (IoU_{pred}) between $\mathcal{P}_{t+\zeta}$ and object, given the bounding box of the target in the reference frame ($\mathcal{G}_t$). Finally, the designed network in ATOM minimizes the mean squared error of $IoU_{G_{t+\zeta}}$ and IoU_{pred}.

In this work, the proposed strategy also provides target patches with background supporters from the reference frame (denoted as $\mathcal{P}_t$) to solve the challenging problems of small object tracking. Besides $\mathcal{G}_t$, the proposed method exploits $\mathcal{P}_t$ just in offline training. Using context features and target parts will assist the proposed network in handling occlusion and viewpoint change problems for small objects. For simplicity, we will describe the proposed offline proposal generation strategy with the process of IoU-prediction. However, the proposed network predicts both IoU and *Center Location Error* (CLE) of test proposals with target, simultaneously.

An overview of the process of offline proposal generation for IoU-prediction is shown in Algorithm 1. The proposed strategy generates $(N/2) - 1$ target proposals from the reference frame, which are generated as $IoU_{Gt} \triangleq IoU(\mathcal{G}_t, \mathcal{P}_t) \geqslant \mathcal{T}_2$. Note that it considers $\mathcal{T}_2 > \mathcal{T}_1$ to prevent drift toward visual distractors. The proposed tracker exploits this information (especially in challenging scenarios involving occlusion and viewpoint change) to avoid confusion during target tracking. The $\mathcal{P}_t$ and $\mathcal{G}_t$ are passed through the reference branch of the proposed network, simultaneously. In this work, an extended modulation vector has been introduced to provide the representations of the target and its parts into the network. That is a set of modulation vectors that each vector encoded the information of one reference proposal. To compute IoU-prediction, the features of the test patch should be modulated by the features of the target and its parts. It means that the IoU-prediction of N test proposals is computed per each reference proposal. Thus, there will be $N^2/2$ IoU predictions. Instead of computing $N/2$ times of N IoU-predictions, the extended modulation vector allows the computation of $N/2$ groups of N IoU-predictions at once. Therefore, the network predicts $N/2$ groups of IoU-predictions by minimizing the mean squared error of each group compared to $IoU_{Gt+\zeta}$. During online tracking, COMET does not generate $\mathcal{P}_t$ and just uses the $\mathcal{G}_t$ to predict one group of IoU-predictions for generated $\mathcal{P}_{t+\zeta}$. Therefore, the proposed strategy will not impose extra computational complexity in online tracking.

3.3.2 Multitask Two-Stream Network

Tracking small objects from aerial view involves extra difficulties such as clarity of target appearance, fast viewpoint change, or drastic rotations besides existing tracking challenges. This part aims to design an architecture that handles the problems

Algorithm 1 : Offline Proposal Generation

Notations: Bounding box $\mathcal{B}$ ($\mathcal{G}_{t+\zeta}$ for a test frame or $\mathcal{G}_t$ for a reference frame), IoU threshold $\mathcal{T}$ ($\mathcal{T}_1$ for a test frame or $\mathcal{T}_2$ for a reference frame), Number of proposals $\mathbb{N}$ (N for a test frame or $(N/2) - 1$ for a reference frame), Iteration number (ii), Maximum iteration (max_{ii}), A Gaussian distribution with zero-mean ($\mu = 0$) and randomly selected variance Σ_r ($\mathcal{N}$), Bounding box proposals generated by a Gaussian jittering $\mathcal{P}$ ($\mathcal{P}_{t+\zeta}$ for a test frame or $\mathcal{P}_t$ for a reference frame)

```
Input: $\mathcal{B}$, $\mathcal{T}$, $\mathbb{N}$, $\Sigma_r$, $max_{ii}$
Output: $\mathcal{P}$
for $i = 1 : \mathbb{N}$ do
    $ii = 0$,
    do
        $\mathcal{P}[i] = \mathcal{B} + \mathcal{N}(\mu, \Sigma_r)$,
        $ii = ii + 1$,
    while $(IoU(\mathcal{B}, \mathcal{P}[i]) < \mathcal{T})$ and $(ii < max_{ii})$;
end
return $\mathcal{P}$
```

of small object tracking by considering recent advances in small object detection. Inspired by [19, 21, 43, 62, 79], a two-stream network is proposed (see Fig. 3.3), which consists of multi-scale processing and aggregation of features, the fusion of hierarchical information, spatial attention module, and channel attention module. Also, the proposed network seeks to maximize the IoU between estimated bounding boxes and the object while it minimizes their location distance. Hence, it exploits a multitask loss function, which is optimized to consider both the accuracy and robustness of the estimated bounding boxes. In the following, the proposed architecture and the role of the main components are described.

The proposed network has adopted ResNet-50 [20] to provide backbone features for reference and test branches. Following small object detection methods, features from `Block3` and `Block4` of ResNet-50 are just extracted to exploit both spatial and semantic features while controlling the number of parameters [62, 80]. Then, the proposed network employs a *Multi-Scale Aggregation and Fusion* (MSAF) module. It processes spatial information via the InceptionV3 module [81] to perform factorized asymmetric convolutions on target regions. This low-cost multi-scale processing helps the network to approximate optimal filters that are proper for small object tracking. Also, semantic features are passed through the convolution and deconvolution layers to be refined and resized for feature fusion. The resulted hierarchical information is fused by an element-wise addition of the spatial and semantic feature maps. After feature fusion, the number of channels is reduced by 1×1 convolution layers to limit

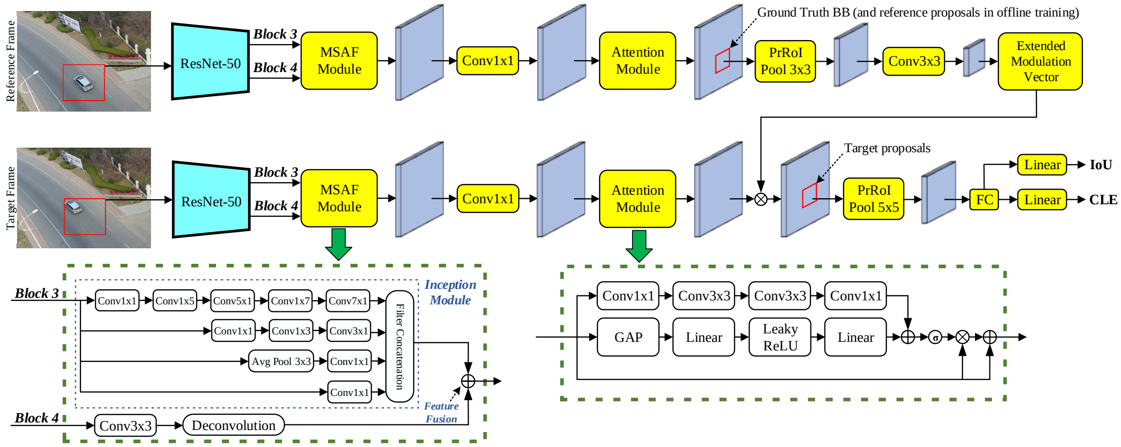

Figure 3.3: Overview of the proposed two-stream network. MSAF denotes multi-scale aggregation and fusion module, which utilizes the InceptionV3 module in its top branch. For deconvolution block, a 3×3 kernel with a stride of 2, input padding of 1, dilation value of 1, and output padding of 1 is used. After each convolution/fully-connected block, batch normalization and leaky ReLU are applied. Extended modulation vector allows COMET to learn targets and their parts in offline training. Also, the fully-connected block, global average pooling, and linear layer are denoted as the FC, GAP, and linear, respectively.

the network parameters. Exploring multi-scale features helps the COMET for small objects that may contain less than 0.01% pixels of a frame.

Next, the proposed network utilizes the *Bottleneck Attention Module* (BAM) [43], which has a lightweight and simple architecture. It emphasizes target-related spatial and channel information and suppresses distractors and redundant information, which are common in aerial images [62]. The BAM includes channel attention, spatial attention, and identity shortcut connection branches. In this work, the SENet [82] is employed as the channel attention branch, which uses *Global Average Pooling* (GAP) and a multi-layer perceptron to find the optimal combination of channels. The spatial attention module utilizes dilated convolutions to increase the receptive field. Lastly, the identity shortcut connection helps for better gradient flow.

After that, the proposed method generates proposals from the test frame. Also, it uses the proposed proposal generation strategy to extract the bounding boxes from the target and its parts from the reference frame in offline training. These generated bounding boxes are applied to the resulted feature maps and fed into a *Precise Re-*

gion of Interest (PrRoI) Pooling layer [21], which is differentiable w.r.t. bounding box coordinates. The network uses a convolutional layer with a 3×3 kernel to convert the PrRoI output to target appearance coefficients. Target coefficients are expanded and multiplied with the features of test patch to merge the information of the target and its parts into the test branch. That is, applying target-specific information into the test branch by the extended modulation vector. Then, the test proposals ($\mathcal{P}_{t+\zeta}$) are applied to the features of the test branch and fed to a 5×5 PrRoI pooling. Finally, the proposed network simultaneously predicts IoU and CLE of test proposals by optimizing a multitask loss function as $\mathcal{L}_{Net} = \mathcal{L}_{IoU} + \lambda\mathcal{L}_{CLE}$, where the $\mathcal{L}_{IoU}$, $\mathcal{L}_{CLE}$, and λ represent the loss function for IoU-prediction head, loss function for the CLE-prediction head, and balancing hyper-parameter for loss functions, respectively. By denoting i-th IoU- and CLE-prediction values as $IoU^{(i)}$ and $CLE^{(i)}$, the loss functions are defined as

$$\mathcal{L}_{IoU} = \frac{1}{N}\sum_{i=1}^{N}(IoU^{(i)}_{G_{t+\zeta}} - IoU^{(i)}_{pred})^2, \tag{3.1}$$

$$\mathcal{L}_{CLE} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}(CLE^{(i)}_{G_{t+\zeta}} - CLE^{(i)}_{pred})^2 & |(CLE^{(i)}_{G_{t+\zeta}} - CLE^{(i)}_{pred}| < 1 \\ \frac{1}{N}\sum_{i=1}^{N}|(CLE^{(i)}_{G_{t+\zeta}} - CLE^{(i)}_{pred})| - \frac{1}{2} & otherwise \end{cases}, \tag{3.2}$$

where the $CLE_{G_{t+\zeta}} = (\Delta x_{G_{t+\zeta}}/width_{G_{t+\zeta}}, \Delta y_{G_{t+\zeta}}/height_{G_{t+\zeta}})$ is the normalized distance between the center location of $\mathcal{P}_{t+\zeta}$ and $\mathcal{G}_{t+\zeta}$. For example, $\Delta x_{G_{t+\zeta}}$ is calculated as $x_{G_{t+\zeta}} - x_{P_{t+\zeta}}$. Also, the CLE_{pred} (and IoU_{pred}) represents the predicted CLE (and the predicted IoU) between bounding box estimations ($\mathcal{G}_{t+\zeta}$) and target, given an initial bounding box in the reference frame. In offline training, the proposed network optimizes the loss function to learn how to predict the target bounding box from the pattern of proposals generation.

In online tracking, the target bounding box from the first frame (similar to [19, 23–25]) and also target proposals in the test frame are passed through the network. As a result, there is just one group of CLE-prediction as well as IoU-prediction to avoid more computational complexity. In this phase, the aim is to maximize the IoU-

```
Algorithm 2 : Online Tracking
Notations: Input sequence ($\mathcal{S}$), Sequence length ($T$), Current frame ($t$), Rough estimation of bounding box ($\mathcal{B}_t^e$), Generated test proposals ($\mathcal{B}_t^p$), Concatenated bounding boxes ($\mathcal{B}_t^c$), Bounding box prediction ($\mathcal{B}_t^{pred}$), Step size ($\beta$), Number of refinements ($n$), Online classification network ($\mathbb{Net}_{online}^{ATOM}$), Scale and center jittering ($Jitt$) with random factors, Network predictions ($IoU$ and $CLE$)
Input: $\mathcal{S} = \{I_0, I_1, ..., I_T\}$, $\mathcal{B}_0 = \{x_0, y_0, w_0, h_0\}$
Output: $\mathcal{B}_t^{pred}$, $t \in \{1, ..., T\}$
for $t = 1 : T$ do
    $\mathcal{B}_t^e = \mathbb{Net}_{online}^{ATOM}(I_t)$
    $\mathcal{B}_t^p = Jitt(\mathcal{B}_t^e)$
    $\mathcal{B}_t^c = Concat(\mathcal{B}_t^e, \mathcal{B}_t^p)$
    for $i = 1 : n$ do
        $IoU, CLE = \text{FeedForward}(I_0, I_t, \mathcal{B}_0, \mathcal{B}_t^c)$
        $\mathbf{grad}_{\mathcal{B}_t^c}^{IoU} = [\frac{\partial IoU}{\partial x}, \frac{\partial IoU}{\partial y}, \frac{\partial IoU}{\partial w}, \frac{\partial IoU}{\partial h}]$
        $\mathcal{B}_t^c \leftarrow \mathcal{B}_t^c + \beta \times [\frac{\partial IoU}{\partial x}.w, \frac{\partial IoU}{\partial y}.h, \frac{\partial IoU}{\partial w}.w, \frac{\partial IoU}{\partial h}.h]$
        $\mathbf{grad}_{\mathcal{B}_t^c}^{CLE} = [\frac{\partial CLE}{\partial x}, \frac{\partial CLE}{\partial y}, \frac{\partial CLE}{\partial w}, \frac{\partial CLE}{\partial h}]$
        $\mathcal{B}_t^c \leftarrow \mathcal{B}_t^c - \beta \times [\frac{\partial CLE}{\partial x}.w, \frac{\partial CLE}{\partial y}.h, \frac{\partial CLE}{\partial w}, \frac{\partial CLE}{\partial h}]$
    end
    $\mathcal{B}_t^{K\times 4} \leftarrow$ Select $K$ best $\mathcal{B}_t^c$ w.r.t. IoU-scores
    $\mathcal{B}_t^{pred} = Avg(\mathcal{B}_t^{K\times 4})$
end
return $\mathcal{B}_t^{pred}$
```

prediction of test proposals using the gradient ascent algorithm and also to minimize its CLE-prediction using the gradient descent algorithm. Algorithm 2 describes the process of online tracking in detail. This algorithm shows how the inputs are passed through the network, and bounding box coordinates are updated based on scaled back-propagated gradients. While the IoU-gradients are scaled up with bounding box sizes to optimize in a log-scaled domain, just x and y coordinates of test bounding boxes are scaled up for CLE-gradients. It experimentally achieved better results compared with the scaling process for IoU-gradients. The intuitive reason is that the network has learned the normalized location differences between bounding box estimations and target bounding box. That is, the CLE-prediction is responsible for accurate localization, whereas the IoU-prediction determines the bounding box aspect ratio. After refining the test proposals ($N = 10$ for online phase) for $n = 5$ times, the proposed method selects the $K = 3$ best bounding boxes and uses the average of these predictions based on IoU-scores as the final target bounding box.

3.4 Empirical Evaluation

In this section, first, the proposed method is compared with the baseline ATOM [19] on the test sets of large-scale LaSOT [8] and GOT-10k [12] datasets. Then, as the main aim, the proposed tracker is evaluated on state-of-the-art benchmarks for small object tracking from aerial view: VisDrone-2019-test-dev [10], UAVDT [54], and Small-90 [56]. Although the Small-90 dataset includes the challenging videos of the UAV-123 dataset with small objects, the experimental results on the UAV-123 [53] dataset (low-altitude UAV dataset (10∼30 meters)) are also presented. However, the UAV-123 dataset lacks varieties in small objects, camera motions, and real scenes [55]. Moreover, traditional tracking datasets do not consist of challenges such as tiny objects, significant viewpoint changes, camera motion, and high density from aerial views. For these reasons and our focus on tracking small objects on videos captured from medium- & high-altitudes, the proposed tracker (COMET) is evaluated on related benchmarks to demonstrate the motivation and major effectiveness for small object tracking.

Experiments have been conducted three times, and the average results are reported. The details about employed visual attributes is provided in Sec. 2.2. The trackers are compared in terms of precision [48], success (or *Success Rate* (SR)) [12, 48], *normalized Area-Under-Curve* (AUC), and *Average Overlap* (AO) [12] metrics by standard benchmarks with default thresholds. In the following, implementation details, ablation analysis, and state-of-the-art comparisons of COMET are presented.

3.4.1 Implementation Details

For offline proposal generation, hyper-parameters are set to $N = 16$ (test proposals number, $(N/2) = 8$ (seven reference proposal numbers plus reference ground truth)), $\mathcal{T}_1 = 0.1$, $\mathcal{T}_2 = 0.8$, $\lambda = 4$, and image sample pairs randomly selected from videos with a maximum gap of 50 frames ($\zeta = 50$). Flipping and color jittering are used for data

Table 3.1: Ablation analysis of COMET considering different components and feature fusion operations on UAVDT dataset.

Metric	COMET	A1	A2	A3	A4	A5
Precision	88.7	87.2	85.2	83.6	88	85.3
Success	81	78	76.9	73.5	80.4	77.2

augmentation of the reference patch. The values for IoU and CLE are normalized to the range of $[-1, 1]$.

The maximum iteration number max_{ii} for proposal generation is 200 for reference proposals and 20 for test proposals. The weights of the backbone network are frozen, and other weights are initialized using [83]. The training splits are extracted from the official training set (protocol II) of LaSOT [8], training set of GOT-10K [12], NfS [52], and training set of VisDrone-2019 [10] datasets. Moreover, the validation splits of VisDrone-2019 and GOT-10K datasets have been used in the training phase. To train in an end-to-end fashion, the ADAM optimizer [84] is used with an initial learning rate of 10^{-4}, weight decay of 10^{-5}, and decay factor 0.2 per 15 epochs. The proposed network trained for 60 epochs with a batch size of 64 and 64000 sampled videos per epoch. Also, the proposed tracker has been implemented using PyTorch, and the evaluations performed on an Nvidia Tesla V100 GPU with 16 GB RAM. Finally, the parameters of the online classification network are set as the ATOM [19].

3.4.2 Ablation Analysis of COMET

A systematic ablation study on individual components of the proposed tracker has been conducted on the UAVDT dataset [55] (see Table 3.1). It includes three different versions of the proposed network consisting of the networks without 1) "CLE-head", 2) "CLE-head and reference proposals generation", and 3) "CLE-head, reference pro-

Table 3.2: Overall & attribute-based evaluations of COMET on the test sets of LaSOT & GOT-10k.

Tracker	LaSOT (AUC metric)															GOT-10k		
	Overall	*IV*	*POC*	*DEF*	*MB*	*CM*	*ROT*	*BC*	*VC*	*SV*	*FOC*	*FM*	*OV*	*LR*	*ARC*	*AO*	$SR_{0.5}$	$SR_{0.75}$
COMET	54.2	57.8	50	56.2	53.2	57.5	53.5	48.7	51.1	53.9	46.3	44.2	46.2	46.8	52.2	59.6	70.6	44.9
ATOM	51.8	56.1	48.3	51.4	49.7	56.4	48.9	45.1	47.4	51.5	42.8	43.3	44.2	44.7	50.5	55.6	63.4	40.2

posals generation, and attention module", referred to as A1, A2, and A3, respectively. Moreover, two other different feature fusion operations have been investigated, namely features multiplication (A4) and features concatenation (A5), compared to the element-wise addition of feature maps in the MSAF module (see Fig 3.3).

These experiments demonstrate the effectiveness of each component on tracking performance results, while the proposed method has achieved 88.7% and 81% in terms of precision and success rates, respectively. According to these results, the attention module, reference proposal generation strategy, and CLE-head have improved the average of success and precision rates up to 2.5%, 1.55%, and 2.25%, respectively. Besides, comparing results of feature fusion operations demonstrate that the element-wise addition has provided the average of precision and success rates up to 0.65% and 3.6% compared to A4 and A5, respectively. Also, the benefit of feature addition previously has been proved in other methods such as [19]. Finally, the proposed tracker is compared with the baseline tracker [19] on the test sets of two large-scale generic object tracking benchmarks, namely LaSOT [8] and GOT-10k [12]. Table 3.2 demonstrates that the COMET also considerably improves the performance of the ATOM [19] on traditional visual tracking datasets.

3.4.3 State-of-the-art Comparison of COMET

For quantitative comparison, COMET is compared with state-of-the-art visual trackers including AutoTrack [68], ATOM [19], DiMP-50 [26], PrDiMP-50 [40], Ocean-online [41], SiamRPN++ [25], SiamMask [24], DaSiamRPN [34], SiamDW [35], CREST [85], MDNet [17], PTAV [86], ECO [87], and MCPF [88] on aerial tracking datasets. Fig. 3.4 shows the achieved results in terms of precision and success plots [48]. According to these results, COMET outperforms top-performing visual trackers on three available challenging small object tracking datasets (i.e., UAVDT, VisDrone-2019-test-dev and Small-90) as well as the UAV-123 dataset. For instance, COMET has outperformed the SiamRPN++ and DiMP-50 trackers by 4.4% and 3.2% in terms of average precision metric, and 3.3% and 3% in terms of average success metric on all datasets, respectively. Besides, it outperforms the PrDiMP and Ocean-online up to 3.3% and 5.4% in average precision metric, and 3.6% and 5.3% in average success metric on the small object tracking datasets. Compared to the baseline ATOM tracker, COMET has improved the average precision rate up to 10.6%, 7.2% and 0.8%, while it increased the average success rate up to 11.2%, 7.1% and 2.9% on the UAVDT, VisDrone-2019-test-dev and Small-90 datasets, respectively. Although COMET slightly outperforms ATOM on the UAV-123 (see Fig. 3.1), it achieved up to 6.2% and 7% improvements compared to it in terms of average precision and success metrics on small object tracking datasets.

These results are mainly owed to the proposed proposal generation strategy and effective modules, which makes the network focus on relevant target (and its parts) information and context information. Furthermore, COMET runs at 24 *FPS*, while the average speeds of other trackers on the referred machine are indicated in Table 3.3. This satisfactory speed has been originated from considering different proposal generation strategies for offline & online procedures and employing lightweight modules in the proposed architecture. The COMET has been evaluated according to various attributes of small object tracking scenarios to investigate its strengths and weaknesses. Table 3.4 and Table 3.5 present the attribute-based comparison of visual trackers. These tables demonstrate that the COMET can successfully handle challenging scenarios for small object tracking purposes. For instance, compared with the DiMP-50, SiamRPN++, SiamMask, PrDiMP & Ocean-online, COMET achieves improvements up to 9.5%, 7.4%, 4.5%, 1.8% & 7.7% for small object attribute, and 4.4%, 2.6%, 5.3%, 3.6% & 5.1% for viewpoint change attribute, respectively. While

Table 3.3: Average speed (*FPS*) of COMET compared with the state-of-the-art trackers on UAVDT dataset.

	COMET	ATOM	SiamRPN++	DiMP-50	SiamMask	ECO	PrDiMP-50
Speed	24	30	32	33	42	35	22

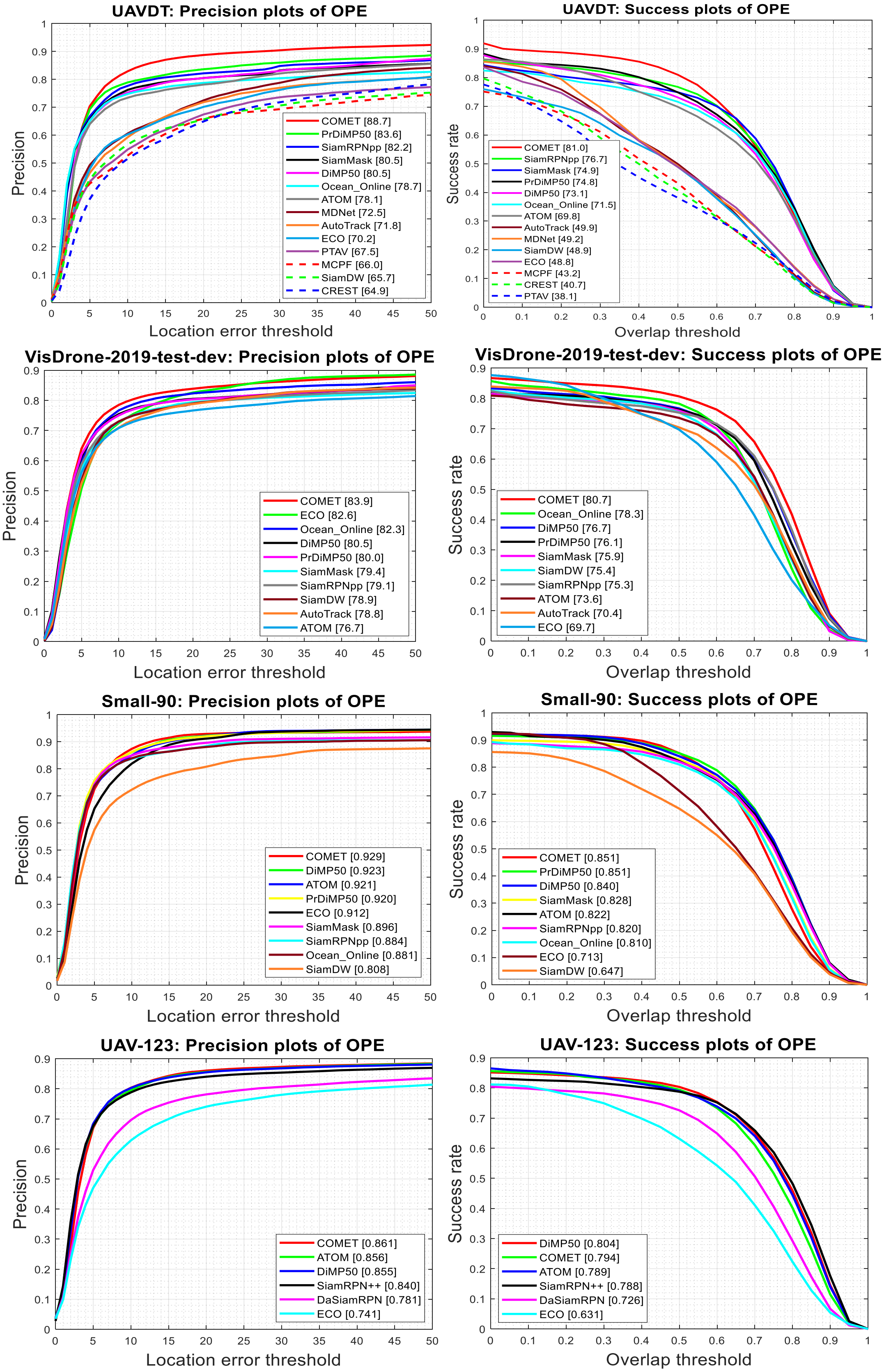

Figure 3.4: Overall precision and success comparisons of the proposed method (COMET) with state-of-the-art tracking methods on UAVDT, VisDrone-2019-test-dev, Small-90, and UAV-123 datasets.

Table 3.4: Attribute-based results of COMET compared with the state-of-the-art trackers in terms of accuracy metric on UAVDT dataset [First, second, and third methods are shown in color].

Tracker	BC	CM	OM	SO	IV	OB	SV	LO	LT
COMET	83.8	86.1	90.6	90.9	88.5	87.7	90.2	79.6	96
ATOM	70.1	77.2	73.4	80.6	80.8	74.9	73	66	91.7
SiamRPN++	74.9	75.9	80.4	83.5	89.7	89.4	80.1	66.6	84.9
SiamMask	71.6	76.7	77.8	86.7	86.4	86	77.3	60.1	93.8
DiMP-50	71.1	80.3	75.8	81.4	84.3	79	76.1	68.6	100
PrDiMP-50	74.4	79.7	82.7	84.1	83.8	83.1	84.7	98.6	73.2
Ocean-online	69.7	72.3	76.2	83.2	87.8	85.6	74.5	83.3	62.5

Table 3.5: Attribute-based results of COMET compared with the state-of-the-art trackers in terms of AUC metric on VisDrone-2019-test-dev dataset [First, second, and third methods are shown in color].

Tracker	Overall	ARC	BC	CM	FM	FOC	IV	LR	OV	POC	SOB	SV	VC
COMET	64.5	64.2	43.4	62.6	64.9	56.7	65.5	41.8	75.9	62.1	42.8	65.8	70.4
ATOM	57.1	52.3	36.7	56.4	52.3	48.8	63.3	31.2	63	51.9	35.6	55.4	61.3
SiamRPN++	59.9	58.9	41.2	58.7	61.8	55.1	63.5	36.4	69.3	58.8	39.6	59.9	67.8
DiMP-50	60.8	54.5	40.6	60.6	62	55.8	63.6	32.7	62.4	56.8	39.8	59.7	66
SiamMask	58.1	57.8	38.5	57.2	60.8	49	56.6	46.5	67.5	52.9	37	59.4	65.1
PrDiMP-50	59.8	58.6	41.1	58	57.5	57	64.2	31.8	67.7	61.2	37.4	58.3	66.8
Ocean-online	59.4	61.1	46.3	59.2	55.3	53	56.6	47.7	66.8	53.4	45.8	62.1	65.3

the performance still can be improved based on IV, OB, LR, LO, and LT attributes, COMET outperforms ATOM by a margin up to 7.7%, 12.8%, 10.6%, 13.6%, and 4.3% on these attributes, respectively.

The qualitative comparisons of visual trackers on UAVDT [54, 55] are shown in Fig. 3.5, in which the videos have been selected for more clarity. According to the first row of Fig. 3.5, COMET successfully models small objects on-the-fly considering complicated aerial view scenarios. Also, it provides promising results when the aspect ratio of target significantly changes. Examples of occurring out-of-view and occlusion are shown in the next rows of Fig. 3.5. By considering target parts and context information, COMET properly handles these problems existing potential distractors. More qualitative results on arbitrary YouTube videos are shown in Fig. 3.6. Please refer to the supplementary material of our COMET paper [44] for the video

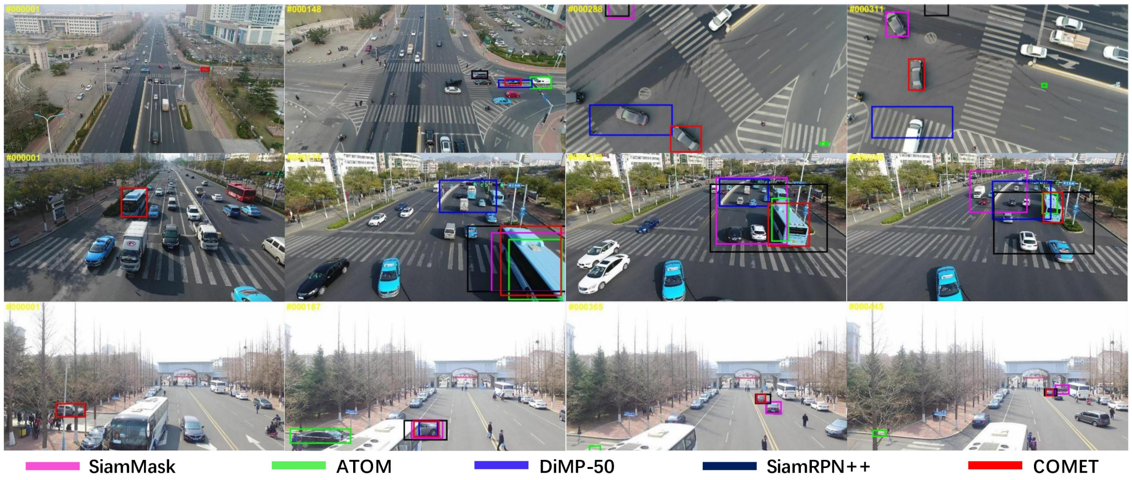

Figure 3.5: Qualitative comparison of the proposed COMET tracker with state-of-the-art tracking methods on S1202, S0602, and S0801 video sequences from UAVDT dataset (top to bottom row, respectively). Best viewed in color.

containing more examples.

3.4.4 Extending COMET to LTCOMET

Short-term trackers search the local window around the previous position of target to localize it in the current frame. However, this assumption is not valid for the videos with object disappearance (e.g., full occlusion & out-of-view) and videos with frames for which local window searching is insufficient (e.g., drastic camera motions or fast object movement especially for small targets). These scenarios frequently happen in aerial videos. Accordingly, the VisDrone-SOT challenge have included long-term sequences in the competition testing set (i.e., VisDrone-test-challenge), starting from 2019 [10, 11]. Motivated by this, we are interested in extending our COMET to globally search the test frame once one of long-term scenarios happen. In doing so, the local tracker (i.e. COMET) should find out that the target is not inside the local window. Then, switch to global search and identify potential targets. Finally, verify which instance is our target and restart the tracker from that instance.

The architecture of our new tracker, called *Long-Term COMET* (LTCOMET), which participated in VisDrone-SOT2020 challenge, is the same as COMET [44] with-

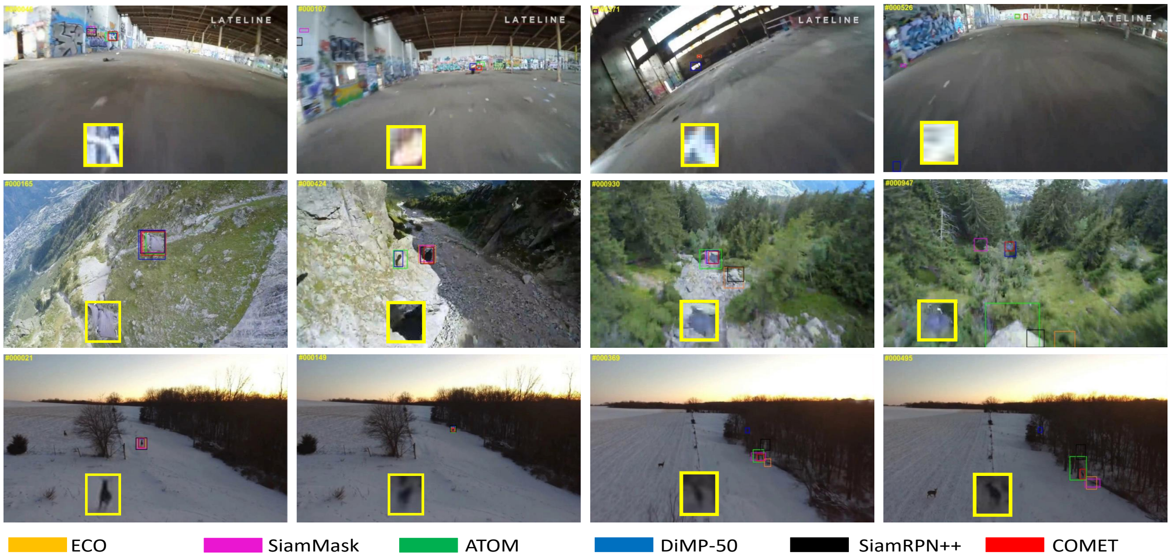

Figure 3.6: Qualitative evaluation of the proposed COMET tracker compared with state-of-the-art visual tracking methods on three YouTube videos. For better visualisation, the magnified target is shown inside a yellow box. Best viewed in color.

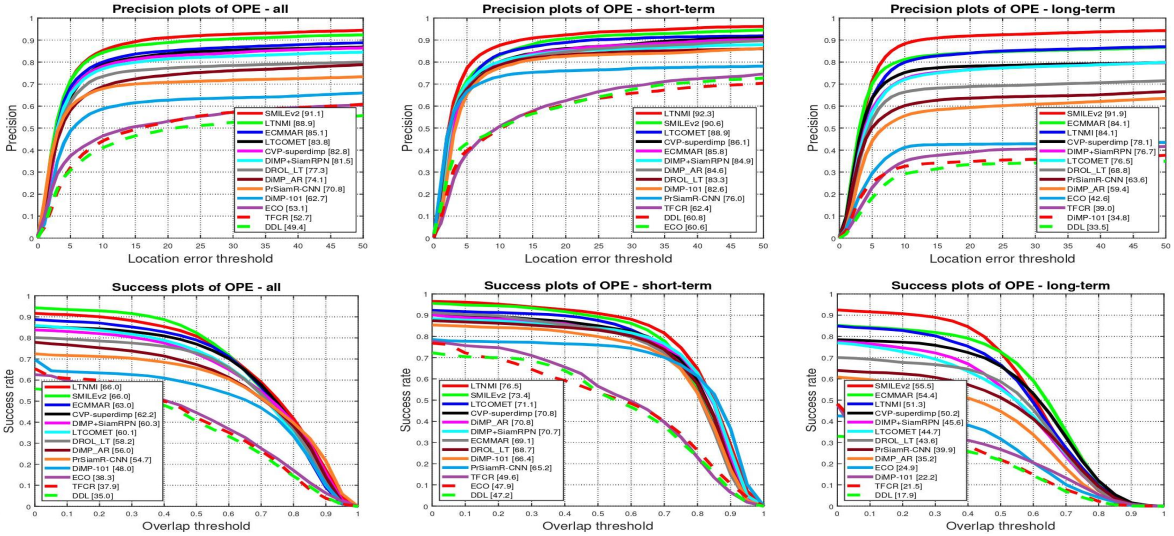

Figure 3.7: The precision and success results of participant trackers including LTCOMET in VisDrone-SOT2020 challenge [11]. Evaluation is done on VisDrone-test-challenge set which includes both short-term and long-term tracking sequences. From left to right, precision and success results on all, short-term, and long-term sequences, respectively.

out using channel reduction after the *Multi-Scale Aggregation and Fusion* (MSAFs) modules. The disappearance of target from local window has been *guessed* using thresholding the confidence score. For global search, we utilize a sliding window search. Finally, we use the online network of our local tracker (i.e. COMET) which is not updated during disappearance as the verifier. In doing so, the obtained global windows are compared to the template model of target before disappearance, and the tracker is restarted from the potential target with confidence score higher than a predefined threshold (i.e. 0.9). However, the tracker might drift towards distractors/ similar objects after restarting from global detection. Moreover, the appearance of target might be different after returning back to the scene. Hence, the model obtained before disappearance might not be reliable.

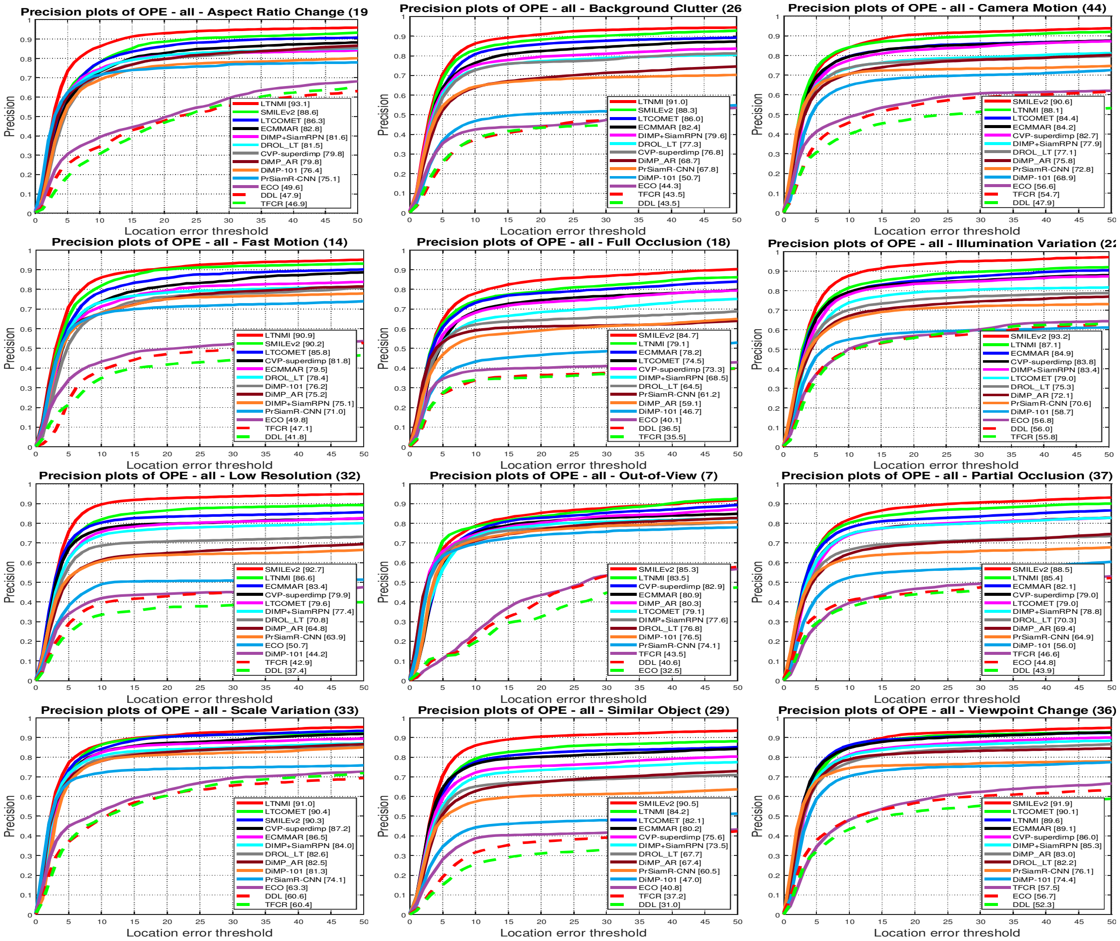

Figure 3.8: The precision results of participant trackers including LTCOMET in VisDrone-SOT2020 challenge [11] based on different visual attributes on VisDrone-test-challenge set. The number of sequences for each attribute is shown in the title of the plot. Best-viewed in color and zooming-in.

The competition test set for VisDrone challenge contains other challenges includ-

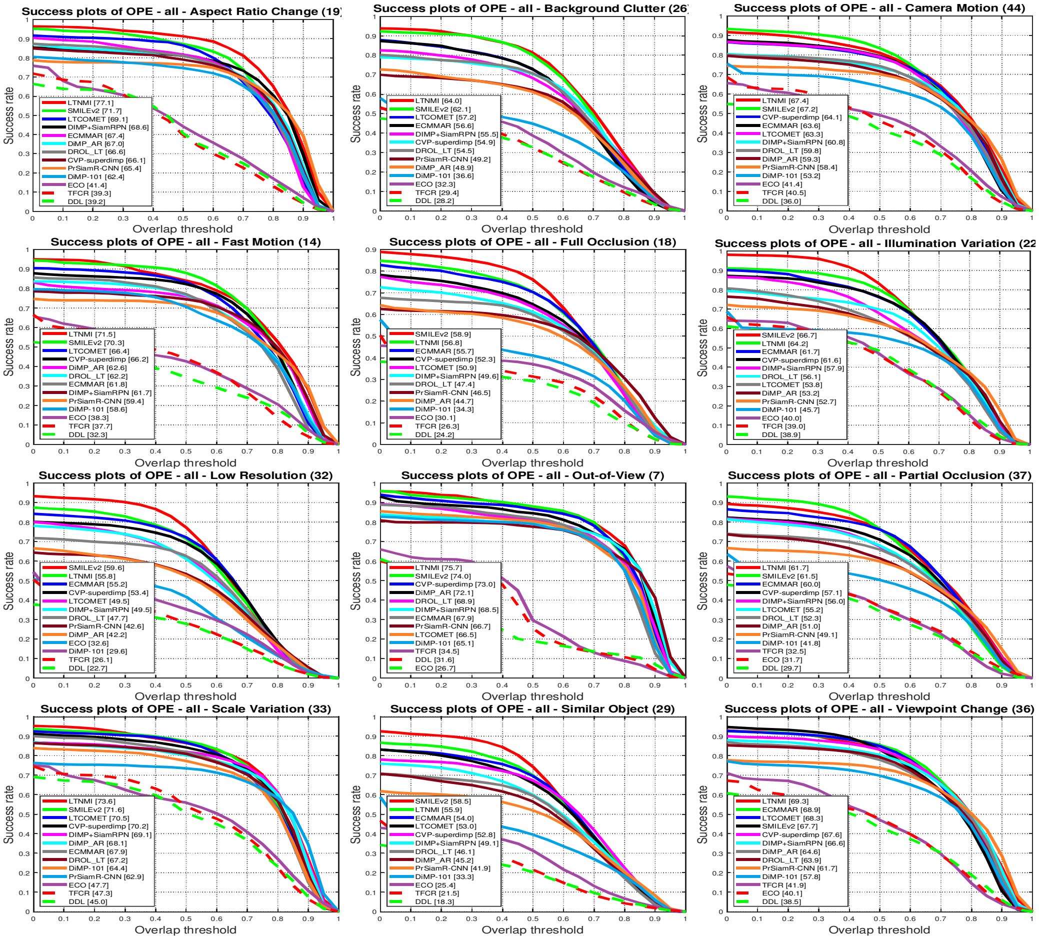

Figure 3.9: The success results of participant trackers including LTCOMET in VisDrone-SOT2020 challenge [11] based on different visual attributes on VisDrone-test-challenge set. The number of sequences for each attribute is shown in the title of the plot. Best-viewed in color and zooming-in.

ing illumination variation, low-resolution, distortion, different weather conditions, and videos captured at night. Hence, we preprocess the input test frames using the *Kindling the Darkness* (KinD) [89] and *Photo-realistic Cascading Residual Network* (PCARN) [90] for light adjustment & degradation removal and obtaining high-resolution patches, respectively. Also, more data augmentations (including photometric and geometric distortions) over the first frame of video are used to provide a diverse initial set for training the online network of COMET [44] at the start of each sequence.

The evaluation results of the participant trackers in VisDrone-SOT2020 [11] are shown in Fig. 3.7. Our LTCOMET tracker is the third top performer tracker on short-term sequences in terms of precision & success, while we ranked sixth for long-term sequences. The main reason is that the other top performer methods use ensemble of state-of-the-art short-term trackers (e.g., PrDimP [40], DiMP [26], SiamRPN++ [25]), long-term & multi-object trackers and detection modules (e.g., Siam R-CNN [71], SORT_MOT [91], and faster R-CNN [33]), and auxiliary inputs (e.g., estimated optical flow) to enhance the performance while we have integrated a simple global search strategy into our COMET short-term tracker. On all the sequences, we ranked fourth & sixth in terms of precision & success, respectively. The results based on different visual attributes can be found in Fig. 3.8 and Fig. 3.9 in terms of precision and success, respectively.

3.5 Summary

A context-aware IoU-guided tracker proposed that includes an offline reference proposal generation strategy and a two-stream multitask network. It aims to track small objects in videos captured from medium- and high-altitude aerial views. First, an introduced proposal generation strategy provides context information for the proposed network to learn the target and its parts. This strategy effectively helps the network to handle occlusion and viewpoint change in high-density videos with a broad view angle in which only some parts of the target are visible. Moreover, the proposed network exploits multi-scale feature aggregation and attention modules to learn multi-scale features and prevent visual distractors. Finally, the proposed multitask loss function accurately estimates the target region by maximizing IoU and minimizing CLE between the predicted box and object. Experimental results on four state-of-the-art aerial view tracking datasets and remarkable performance of the proposed tracker demonstrate the motivation and effectiveness of proposed components for small object tracking purposes. An extended version of the proposed tracker is developed for enhancing the robustness in long-term scenarios.

Chapter 4

DESTINE

In this chapter, first, segmentation tracking and *Video Object Segmentation* (VOS) computer vision tasks are introduced. Then, the existing methods and benchmarks for these two tasks are reviewed. Finally, the *Adaptive Visual Tracking and Instance Segmentation* (DESTINE) tracker, participated in the famous *Visual Object Tracking* (VOT) 2020 challenge [3] is presented.

4.1 Introduction

The goal of segmentation tracking is to predict the binary mask of the target in each frame of video with having its ground truth box in the first frame as input. Compared with regular bounding box tracking, predicting this pixel-level representation provides better understanding of detailed object state through the video at the cost of sacrificing the computational resource.

Segmentation tracking is related to another computer vision task, called *Video Object Segmentation* (VOS) which is studied in both unsupervised and semi-supervised settings in literature. The former aims at predicting the binary mask of class-agnostic salient objects in all frames of video, while the later takes the mask of targets in the first frame as input to predict accurate targets' mask at the subsequent frames. The use of other levels of supervision for VOS resulted in interactive and weakly-supervised VOS methods [92].

VOS in semi-supervised settings is highly related to segmentation tracking, however the minor differences between these two tasks are 1) For VOS, the benchmark datasets [93–95] contain targets with complex shapes/edges, but the videos are less challenging compared with tracking benchmark videos. For instance, VOS considers very short videos containing fewer distractors & the primary target objects usually occupy majority pixels of image, 2) VOS is usually considered in multi-object settings, 3) Most of the existing VOS methods are really slow while efficiency is an important criteria for *Visual Object Tracking* community, and 4) In tracking community, even for segmentation trackers, the supervision for first frame is usually considered as bounding box of target. However, in semi-supervised VOS, the mask of targets are provided.

Accordingly, VOS methods are usually slow, not fulfilling the real-time requirement for tracking community. However, for scenarios that bounding box tracking is not sufficient (e.g., highly deformable & articulated objects), the need to obtain the precise mask of target is more pronounced. Considering the recent interest of *Visual Object Tracking* (VOT) community in VOS task, we can expect the rapid merging of these two problems which will result in more accurate trackers & more robust real-time VOS methods in the next few years. Following this, we are interested in developing a segmentation tracker using the existing efforts in VOT & VOS literature. In Sec. 4.2, we will review the existing notable methods and datasets for VOS & segmentation tracking. Then, we will introduce our proposed DESTINE tracker, and will review the results of VOT-ST2020 challenge in Sec. 4.3.

4.2 Semi-Supervised VOS/ Segmentation Tracking

4.2.1 Existing Methods

Semi-supervised VOS methods [92] can be classified into three main groups. *Propagation-based* [96, 97] methods utilize the spatio-temporal consistency in adjacent frames to propagate the initial ground truth mask of targets in the first frame to the other frames of video. However, these methods have difficulties with handling temporal discontinuities. *Fine-tuning-based* [98–100] methods fine-tune a pre-trained segmentation network on the appearance of the targets and their augmentations in the first frame of test video. Then, they predict the segmentation mask of targets in the subsequent frames using the new weights without any temporal information. This approach is computationally expensive and suffers from over-fitting to the appearance of targets in the first frame. Also, due to the difference between the distribution of training samples for pre-training and training at inference time, online learning does not work as expected. Utilizing meta-learning to address these issues have been explored in some of the recent works [101, 102]. Finally, *matching-based* methods use the ground truth mask for the first frame as well as the predicted masks from previous frames to obtain the mask for the current frame. Siamese architecture [103], memory networks [104–106], and distance maps [107] have been used by matching-based methods to provide promising results & quite efficiency. Note that some of the existing methods combine different ideas or extend online learning to other frames to increase the performance.

MaskTrack [108] combines the propagation-based & fine-tuning-based VOS approaches by training a network from scratch in inference phase which uses the predicted mask of previous frame & optical flow as auxiliary inputs. DIPNet [109] introduces a two-stage method where they obtain a coarse mask using *Dynamic Identity Propagation* in the first stage. Then, in the second stage, they utilize a *Spatial Instance Segmenta-*

tion which is fine-tuned on the first frame of test video at inference time to obtain the detailed mask guided by the mask obtained in the previous stage. To learn to update the segmentation model for matching-based VOS, [106] proposes an episodic graph memory network with fixed memory size and learnable read & write controllers.

Incorporating the recent advances of *Visual Object Tracking* (e.g., Siamese networks, online classification networks, and discriminative loss functions) to develop a new group of robust, accurate, and fast VOS/segmentation trackers is also studied in recent works [102, 103]. This effort got accelerated by substituting the ground truth bounding box of targets with binary masks for the famous *Visual Object Tracking* (VOT) challenge, starting from 2020 [3]. A simple approach to develop a two-stage segmentation tracker is to utilize a weakly supervised segmentation method which receives bounding box prediction from a regular tracker as the input. Following that, SiamMask [24] is one of the celebrated efforts in tracking community which tries to solve both bounding box & segmentation tracking using a unified two-stage network. Very recently, [110] considers spatio-temporal consistencies in weakly supervised segmentation to provide promising results. On the other hand, the *Discriminative Single Shot Tracker* (D3S) [111] aims at developing a one-stage segmentation tracker which is trained on VOS datasets. Then, they obtain the bounding box representation from the predicted masks. Interestingly, they achieve state-of-the-art or competitive performance on tracking benchmarks, while they only use VOS training datasets which are less challenging compared with tracking benchmarks. Their method consists of refining [112] a rough mask obtained from concatenation of *Geometrically Constrained Euclidean Model* (GEM) and *Geometrically Invariant Model* (GIM) which are obtained from ATOM [19] tracker and Video Match [103] method, respectively. Also, the refinement is done following the Sharp Mask [112] method to combine feature maps from different layers of ResNet backbone with the coarse mask to obtain the final detailed mask. Inspired by the recent discriminative visual trackers [19], FRTM-VOS [102] introduces a novel two-stage VOS method which the first stage is a shallow network consisting of two convolutional layers trained in an online manner using discrimnative loss to predict a coarse mask of targets. Then, they use a segmentation refinement network which is trained only in offline phase to convert the prediction of previous stage to accurate mask. However, our experiments show that while the shallow network of FRTM-VOS [102] uses the discriminative loss & online learning, it suffers from robustness issues compared with the state-of-the-art discriminative visual object trackers.

4.2.2 Datasets

YouTube-VOS [93]: This large-scale dataset contains 4.453K YouTube videos of $3 \sim 6$ seconds long, splitting into 3.471K, 474, and 508 clips for training, validation, and test sets. While the ground truth masks for the first two sets are publicly available for all the frames, the test set with released first frame's ground truth mask has been used for evaluating the participant VOS methods in corresponding competition. This dataset covers 94 different categories including humans, animals, vehicles, common objects, and accessories. The samples are collected from the YouTube-8M video classification dataset.

DAVIS 2017 [94, 95]: Densely-Annotated Video Segmentation 2017 provides 150 sequences splitting into 60, 30, 30, and 30 clips for training, validation, test-dev, and test-challenge sets with multiple foreground objects in the video. The ground truth masks for all the frames of the first two sets are released, but the second two sets with released first frame's ground truth are used for the competition. A preliminary version of this dataset with 50 videos and single foreground object per video was presented in DAVIS 2016. [113]. For evaluating vos methods, region similarity (IoU/Jaccard index) & contour accuracy (F measure) between the prediction and ground truth are used. [113]

VOT-ST 2020 [3]: The VOT-ST2020 introduces the first segmentation tracking dataset which contains 60 challenging sequences with released first frame's ground truth mask, and the evaluation is done using the official Python toolkit. Compared with the previous year's competitions [74, 75], ground truth masks are used instead of (rotated) bounding boxes, and the Matlab toolkit is replaced by the new Python toolkit. This toolkit implements the new evaluation protocol, i.e., instead of resetting the tracker after failure which results in unfair evaluation, they divide each sequence to anchors. Then, they redefine robustness, accuracy, and *Expected Average Overlap* (EAO) for this protocol.

4.3 Method and Results

Motivated by the recent advances in VOT & VOS community [3, 19, 24, 26, 102], we are interested in developing a robust and accurate segmentation tracker to participate in VOT-ST2020 challenge. We utilize the robustness of visual object trackers and accuracy of VOS methods to develop a single object segmentation tracker. Accordingly, our proposed DESTINE tracker is a two-stage method consisting of axis-aligned bounding box estimation and mask prediction, respectively.

For the first-stage we combine the predictions of two state-of-the-art trackers, running in parallel at inference time, which the main robust tracker switches to the auxiliary accurate tracker when the *Intersection over Union* (IoU) & *normalized L1-Distance* (nL1D) between the prediction of two trackers meets the predefined thresholds (e.g. $IoU > 0.8$ & $nL1D < 0.2$). In this way, we prioritize robustness to accuracy and switch to more accurate tracker in simple frames. Intuitively, we used DiMP-50 [26] as the main tracker and SiamRPN++ [25] as the auxiliary one, but decided to change the auxiliary tracker to ATOM [19] as this modification resulted in better performance on validation data.

For the second stage, we use the modified SiamMask [24] to provide a coarse binary mask of the predicted box in previous stage. Then, the refinement network of FRTM-VOS [102] runs over the predictions of SiamMask. We observed that while

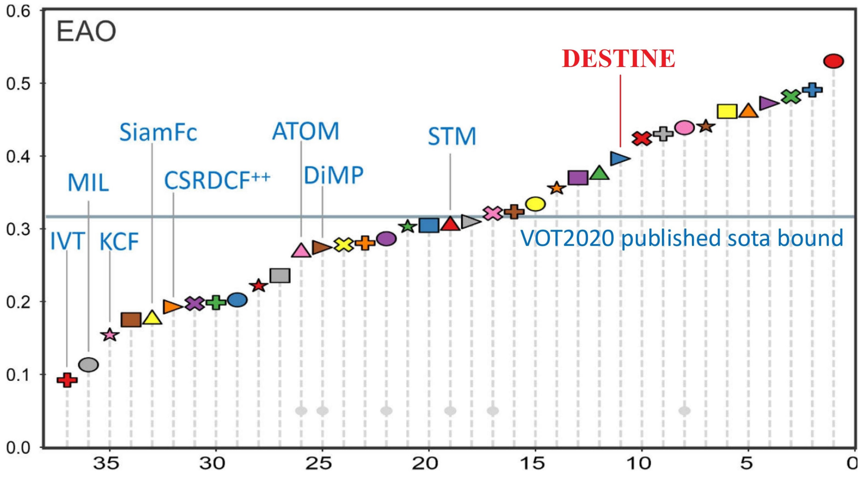

Figure 4.1: Results of participant trackers including DESTINE in VOT-ST2020 challenge [3] in terms of Expected Average Overlap (EAO).

the refinement network provides very accurate masks for normal targets, it misses the boundary pixels of small & fast moving targets. Accordingly, if the ratio of foreground pixels between the two masks is lower than a predefined threshold (e.g., $ratio < 0.43$), we use the output of SiamMask as the final prediction.

Visual Object Tracking- Short-term Tracking (VOT-ST) challenge has started to use the segmentation mask annotation instead of bounding box, starting from 2020 [3]. They also introduce a new evaluation metric and integrate this new metric into a newly proposed Python evaluation toolkit. Each participant should run the tracker over the publicly available datasets using the official toolkit and share the packed results with the VOT challenge organizers. The VOT organizers evaluated the top-5 trackers on a sequestered dataset to avoid over-fitting. Following these steps, as shown in Fig. 4.1, our DESTINE ranked 11th among 37 participant trackers in VOT-ST2020 baseline in terms of *Expected Average Overlap* (EAO). The RPT ranked as the first tracker by combining ATOM [19], RepPoints [114], and D3S [111] methods. Also, three different versions of Ocean [41] tracker, i.e., OceanPlus, fastOcean, and Ocean ranked 2nd, 6th, and 9th, respectively.

4.4 Summary

A two-stage segmentation tracker consisting of an axis-aligned bounding box estimation and mask prediction is developed. First, DiMP-50 [26] is used as the main tracker switching to ATOM [19] when IoU and normalized L1-Distance between the predictions meet the predefined thresholds. Then, SiamMask [24] segments the predicted box of previous stage and the refinement network of FRTM-VOS [102] is run over that. Finally, DESTINE selects the best target mask according to the ratio of foreground pixels for two mask predictions.

Chapter 5

CHASE

In this chapter, first, the motivations of automatic architecture search for obtaining a well-performing visual object tracker are discussed. Then, the literature of *Neural Architecture Search* (NAS), with a focus on differentiable NAS has been reviewed. Finally, the *CHASE: Robust Visual Tracking via Cell-Level Differentiable Neural Architecture Search* [115] method; accepted in British Machine Vision Conference (BMVC) 2021 has been presented. Accordingly, this chapter is mainly based on the CHASE paper [115].

5.1 Introduction

Many efforts have been done to find the well-performing Siamese networks for *Visual Object Tracking* [4, 116, 117]. These network architectures are manually designed by extensive trial & errors of computer vision experts. However, this approach is time-consuming, depends on prior knowledge, and some good luck. Moreover, it is biased towards human priors with no guarantees achieving the highest effectiveness. Accordingly, automatically searching the best modules for Siamese networks becomes a promising research problem to explore. Following the recent advances of AutoML [13] in machine learning & computer vision applications, researchers have shown a great interest in automatic design of neural architectures, i.e., *Neural Architecture Search* (NAS). Within the computer vision community, prior works have well-explored NAS for classification task [14, 15]. There are also a handful of research works exploring the potential of NAS for other applications including object detection [118, 119], semantic segmentation [120, 121], image super-resolution [122], and cell segmentation [123]. However, to the best of our knowledge, the potential of NAS has not been explored for *Visual Object Tracking*, yet. Very recently, the LightTrack [124] uses evolutionary search to obtain lightweight architectures for resource-limited hardware platforms. Also, it uses single-path uniform sampling and lightweight building blocks to achieve more compact architectures and reduce the computational costs. However, single-path sampling decouples the optimizations of the weights and architecture parameters of the super-net, leading to large-variance to the optimization process and tendency to a non-complex structure [119]. The LightTrack [124] has inherited the limitations of evolutionary algorithms as well as single-pass search approaches. Furthermore, it searches within a limited search space and stacks the basic blocks to construct the final architecture.

In contrast, the aim of this work is to automatically discover the best architecture block (or cell) that adapts large-scale trained backbone features to the objectives of Siamese tracking networks. It modifies DARTS [125] that provides interesting advantages such as weight-sharing, gradient-based search, efficiency, and simplicity to have better generalization. However, the primary differences include (i) cell-level NAS instead of searching stacked cells together, (ii) integrating cell-level NAS into Siamese framework especially beneficial for visual tracking, (iii) employing operation-level Dropout without hand-crafted constraints in [126, 127], and (iv) proposing an early-stopping strategy for searching procedure to address the over-fitting problem and multiple retraining from scratch to select the best cell.

The proposed approach (CHASE) takes advantage of the 2nd-order DARTS by learning a cell into Siamese tracking networks. This is contrary to prior works such as [125–131] searching for multiple stacked cells in CNN/RNN architectures using the simple 1st-order DARTS with lower performance. The CHASE provides a simple, efficient, and generalizable approach considering visual tracking purposes, i.e., high performance and speed. Besides, DARTS-based methods require searching on a small proxy dataset and transferring the architecture blocks to the large-scale target task to address the high GPU memory consumption issues. However, the CHASE performs a cell-level architecture search, which allows directly utilizing a large-scale tracking dataset. Last but not least, this work does not apply any post-processing after the network search to restrict the number of skip-connections in contrast to [126, 127]. In fact, it removes prior heuristics since the proposed early-stopping provides a performance-aware cell derivation strategy during the searching phase. It exploits a hold-out sample set for validating the generalization of the best cell. Thus, it finds the saturated searching point to address the over-fitting problem and the performance gap between the search and evaluation phases [127], and then it can select the best cell without requiring multiple retraining from scratch. Finally, the effectiveness of NAS exploitation and its generalization is validated by employing three versions of DARTS [125, 130] and integrating the proposed approach into two visual trackers [26, 40]. In summary, the main contributions are as follows:

- A novel cell-level differentiable architecture search mechanism is proposed to automate the network design of the tracking module during offline training. It is effectively integrated into Siamese tracking network architectures to directly optimize a cell on a large-scale tracking dataset. Our approach is simple, efficient, and easy to be incorporated into existing Siamese trackers for improving performance. This idea can also be used in other computer vision tasks.

- An early-stopping strategy is proposed to improve the generalization performance of selected cell architecture. This simple yet effective performance-aware cell derivation strategy finds the best cell during the searching phase without requiring inefficient multiple re-training from scratch.

- Extensive experimental evaluations on five widely used visual tracking benchmarks demonstrate the superior performance of the proposed approach. Moreover, it is practically shown to boost the overall performance when applied to existing baselines.

5.2 Neural Architecture Search

The design of the deep neural network regulates its ability to automatically learn feature extraction. Accordingly, many researchers in the computer vision community have focused on the architecture engineering of networks [132]. However, the manual design of networks cannot guarantee to suggest the optimal architectures on the target datasets. Motivated by that, the computer vision community has indicated an increasing interest in *Neural Architecture Search* (NAS) to automate the design of neural networks. The *search space*, *search strategy*, and *performance evaluation strategy* are the main components of NAS where the search strategy aims at picking the best candidate architecture from the search space based on the estimated performance of candidates [14, 15]. The search space plays the prominent role in determining the final architecture selected by the NAS methods. By designing an appropriate search space for the target application, even the random search strategy can provide satisfactory results. However, developing more complex and effective search strategies is required. Generally, based on the diversified search strategies, NAS methods can be divided into four major groups of evolutionary algorithm (EA)-based, Reinforcement Learning (RL)-based, Bayesian Optimization (BO)-based, and gradient-based algorithms [14, 15]. While the EA-based algorithms were introduced many years ago [133], the majority of practical techniques to obtain the best deep neural networks using NAS has been developed under RL-based paradigm. However, the required computational resource for most of RL-based methods is not accessible for ordinary users. Recently, gradient-based methods introduced efficiency into NAS, while they achieve competitive performance to the state-of-the-art RL-based NAS methods.

More specifically, the celebrated *Differentiable Architecture Search* (DARTS) method introduces a continuous relaxation of discrete architecture representation. To search the best architecture, they consider a stack of candidate normal & reduction cells and each cell is represented as a *Directed Acyclic Graph* (DAG). Each edge is a mixture of all the candidate operations, and connects the intermediate nodes to all the previous nodes; forming a complete DAG in forward direction. By doing so, they share the parameters between all the candidate architectures, and use the Softmax of mixing parameters of operations as the continuous representation of architecture. Then, they optimize mixing parameters of operations & weight parameters of network using a bi-level optimization approach, by considering the 1st- & 2nd-order approximation-based approaches according to the calculation of architecture gradient where the 2nd-order one leads to better performance but lower search speed. Finally, to obtain the best subgraph (i.e., the preferable or ideally optimal network), they use hard pruning over values of mixture weights.

DARTS increases the efficiency at the cost of tremendous memory usage, leading to impractical settings for very large search spaces. Due to the high memory usage, they cannot search directly on the large-scale datasets, which make them use limited data as the proxy dataset for searching phase. This causes performance gap as we use a stack of the best normal and reduction cells in final training phase on large-scale target datasets. Consequently, DARTS suffers from (i) the performance gap between the search & evaluation phases [126, 127], (ii) repeating blocks restriction [134], (iii) performance collapse [128, 129, 131] due to the model over-fitting, (iv) degenerate architectures [131], (v) aggregation of skip connections [126, 127, 130], and (vi) requiring multiple re-training from scratch.

Since ICLR 2019 that DARTS method is published, many methods have been introduced to address its issues. The *Progressive DARTS* (PDARTS) [126] gradually increases the network depth assisted by the search space approximation and regularization. The ProxylessNAS [134] proposes learning architectures on large-scale datasets, path-level pruning, and latency regularization loss to address repeating blocks restriction, GPU memory consumption, and hardware limitations. The DARTS+ [128] proposes an early stopping paradigm with hand-crafted constraints to avoid the performance collapse of DARTS due to the model over-fitting in the search phase. To improve the robustness, the RobustDARTS [131] investigates the failure cases of DARTS causing degenerate architectures with inferior performance. It introduces an adaptive regularization and early stopping criterion with the dominant Hessian eigenvalue of validation loss. The DARTS- [129] proposes an indicator-free approach to handle the performance collapse & search instability of DARTS. It distinguishes two roles of skip connections (i.e., stabilization of super-net training & candidate operation) by an auxiliary skip connection between every two nodes. Finally, the Fair-DARTS [130] proposes the collaborative competition approach and auxiliary loss to address the aggregation of skip connections & discretization discrepancy problems, respectively. Most DARTS-based methods (e.g., [126–131]) employ the 1st-order DARTS to reduce computational complexity, allowing the search procedure on some stacked cells. The 2st-order DARTS fully exploits training and validation information and converges to a better local optimum. This work integrates a modified cell-level 2nd-order DARTS into the Siamese framework to track visual targets. The proposed early-stopping strategy and operation-level Dropout [126, 127] without any constraints are exploited to address the over-fitting problem, test-validation performance gap, and the best cell architecture selection.

5.3 Proposed Approach: CHASE

The primary motivation is to automatically adapt the robust features extracted from the backbone to the tracking objective by a computational cell (see Fig. 5.1). Hence, this work exploits a modified version of DARTS [125] that forms an ordered *Directed Acyclic Graph* (DAG) with $\mathcal{N}$ nodes as its computational cell, which is learned through architecture search procedure. The CHASE learns a cell integrated into a Siamese tracking architecture to avoid dramatically affecting the computational complexity

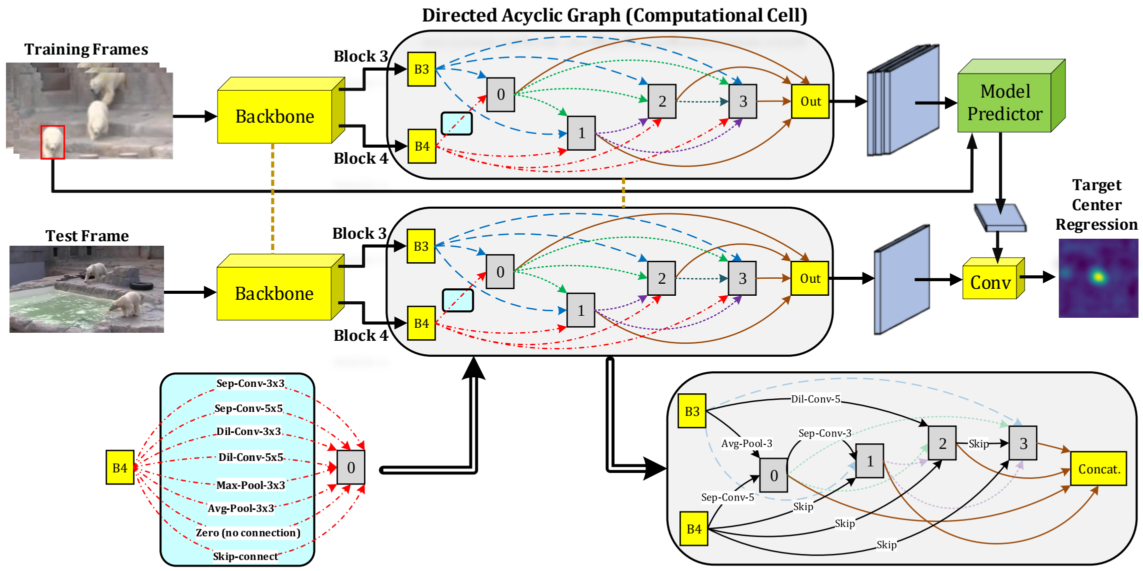

Figure 5.1: An overview of the proposed CHASE tracker. Cell-level NAS is integrated into the TCR network (with Siamese structure) of the baseline tracker [40] to adapt backbone features extracted from `Block3` and `Block4` to the network's objective. First, a computational cell is formed in searching phase in which each edge (dashed line) is a mixture of candidate operations (shown as a blue box for one edge), each intermediate node is connected to all the previous nodes, and the output node is the concatenation of intermediate nodes (shown by brown solid lines). The objective of this phase is to find the optimal sub-graph (i.e., the best cell shown at the bottom-right) by jointly optimizing the weights and architecture parameters of the cell. Then, in training phase, the computational cell is replaced by the best cell, and the whole pipeline is trained from scratch. Finally, the network is used in evaluating phase for visual tracking.

and tracking speed. PrDiMP [40] is used as the baseline to demonstrate the effectiveness of the proposed approach for visual tracking. It includes the *Target Center Regression* (TCR) & *Bounding Box Regression* (BBR) networks, while it predicts the conditional probability density to minimize the *Kullback-Leiber* (KL) divergence between the predictions and label distribution (see [40] for more details). The CHASE tracker replaces additional convolutional blocks after the backbone with a DAG to find the best operations and node connections.

5.3.1 Cell-Level NAS for Visual Tracking

In this section, DARTS is adapted to a Siamese tracking network to move toward our objectives and critical aspects of visual tracking. In proposed approach, the computational cell has two input nodes and four intermediate nodes. The CHASE fuses multi-level deep features extracted from `Block3` & `Block4` of ResNet-50 [20] in designing the cell, according to their importance for visual tracking [4, 116]. Given a feature map $\mathcal{X}^{(i)}$ at node i, the corresponding latent representation at intermediate node j is computed as $\mathcal{X}^{(j)} = \sum_{i<j} \mathfrak{p}^{(i,j)}(\mathcal{X}^{(i)})$, where $\mathfrak{p}^{(i,j)}$ stands for candidate operations (from a predefined set $\mathcal{P} = \{\mathfrak{p}_1^{(i,j)}, \mathfrak{p}_2^{(i,j)}, ..., \mathfrak{p}_{\mathcal{M}}^{(i,j)}\}$ in the search space) on edge $\zeta^{(i,j)}$. Since the DARTS tends to aggregate skip connections due to the rapid error decay during its optimization [128, 130], the CHASE employs the operation-level Dropout without constraints in [126, 127] with an initial rate τ, which gradually decays during the search procedure. The CHASE does not control the number of skip connections to preserve flexibility in cell design and improve training stability. To relax the problem into a continuous search space, the mixed output for $\zeta^{(i,j)}$ is calculated by

$$\bar{\mathfrak{p}}^{(i,j)}(\mathcal{X}) = \sum_{\mathfrak{p}\in\mathcal{P}} \frac{exp(\alpha_{\mathfrak{p}}^{(i,j)})}{\sum_{\hat{\mathfrak{p}}\in\mathcal{P}} exp(\alpha_{\hat{\mathfrak{p}}}^{(i,j)})} \mathfrak{p}(\mathcal{X}), \tag{5.1}$$

in which $\alpha_{\mathfrak{p}}^{(i,j)}$ is the operation mixing weight associated with the operation $\mathfrak{p}$ between nodes i and j. By doing so, the cell architecture search converts into the learning of parameters $\alpha = \{\alpha_1^{(i,j)}, \alpha_2^{(i,j)}, ..., \alpha_{\mathcal{M}}^{(i,j)}\}$. To jointly learn network parameters ($\mathcal{W}$) and architecture parameters (α), the *gradient descent* (GD) algorithm is used to minimize the training ($\mathcal{L}_{tr}$) and validation losses ($\mathcal{L}_{val}$) by performing the bi-level optimization problem

$$\min_{\alpha} \quad \mathcal{L}_{val}(\mathcal{W}^*(\alpha), \alpha) \tag{5.2}$$

$$\text{s.t.} \quad \mathcal{W}^*(\alpha) = \arg\min_{\mathcal{W}} \mathcal{L}_{tr}(\mathcal{W}, \alpha). \tag{5.3}$$

To avoid expensive inner optimization, the DARTS reduces the evaluation of architecture gradient by applying the finite difference approximation as

$$\nabla_{\alpha}\mathcal{L}_{val}(\mathcal{W}^*(\alpha), \alpha) \tag{5.4}$$

$$\approx \nabla_{\alpha}\mathcal{L}_{val}(\mathcal{W}', \alpha) - \eta \frac{\nabla_{\alpha}\mathcal{L}_{tr}(\mathcal{W}^+, \alpha) - \nabla_{\alpha}\mathcal{L}_{tr}(\mathcal{W}^-, \alpha)}{2\epsilon} \tag{5.5}$$

where $\mathcal{W}' = \mathcal{W} - \eta \nabla_{\mathcal{W}} \mathcal{L}_{tr}(\mathcal{W}, \alpha)$, $\mathcal{W}^{\pm} = \mathcal{W} \pm \epsilon \nabla_{\mathcal{W}'} \mathcal{L}_{val}(\mathcal{W}', \alpha)$. Also η and ϵ are the learning rate for a step of inner optimization and a small scalar, respectively. Accordingly, the 2nd-order approximation of DARTS requires two forward passes for $\mathcal{W}$ and two backward passes for α, contrary to the 1st-order DARTS requiring one forward pass for each one.

The 1st-order DARTS provides the ability to search an architecture by stacking multiple cells according to its simplicity and low complexity, e.g., [126–131]. Although differentiable NAS aims at minimizing the validation loss to find optimal architectures, the 1st-order DARTS cannot guarantee that the validation loss is sufficiently small due to ignoring the optimization on fully-trained weights $\mathcal{W}^*(\alpha)$. The 2nd-order DARTS embeds the training loss in updating architecture parameters. Hence, it achieves more stability and higher performance than the 1st-order DARTS by fully exploiting training & validation information and converging to a better local optimum. However, it increases the computational complexity not efficient for optimizing stacked cells. The CHASE enjoys the modified 2nd-order DARTS according to learning one cell that adapts large-scale trained backbone features to the tracking objectives. Moreover, the DARTS [125] suffers from some problems including i) deriving the best discrete architecture with the best validation performance by re-training top-k architectures ($k = 4$) from scratch, and ii) the performance collapse and over-fitting problems on the validation set, resulting in poor generalization on test datasets. To address these challenges, the proposed CHASE focuses on cell-level search and proposes an early stopping strategy to address the over-fitting problem and multiple re-training from scratch.

5.3.2 Early Stopping

To alleviate the test-validation gap of DARTS, prior works (e.g., [128, 131]) impose strong early stopping priors or extra computing costs. However, these methods run several times and re-train each best architecture from scratch to select the final one.

This work performs a performance-aware cell derivation by the proposed early stopping strategy to address these limitations simultaneously. In particular, generic visual tracking seeks to learn target models generalizable to various appearance changes and real-world challenging scenarios. Hence, the proposed strategy introduces a hold-out sample set represented for generalization validation. Note that the CHASE never uses test sets for this purpose. While the CHASE respectively optimizes $\mathcal{W}$ and α on the training and validation sets, it calculates the hold-out loss ($\mathcal{L}_{ho}$) of mixture operations. Then, it derives the best cell architecture at the minimum hold-out loss on the hold-out set by $\mathfrak{p}_o^{(i,j)} = \arg\max_{\mathfrak{p}\in\mathcal{P}} \alpha_{\mathfrak{p}}^{(i,j)}$. This search-stage cell selection originates from the reduced discrepancies between the continuous cell encoding and the derived discrete cell due to the searching one cell using the proposed modified 2nd-order DARTS, resulting in no several re-training requirements from scratch. That is, the CHASE finds the best cell during the searching phase and then trains it from scratch once.

5.4 Empirical Experiments

Herein, the implementation details of the proposed approach, ablation analysis, and tracking results of the best cell architecture on benchmark datasets are reported.

5.4.1 Implementation Details

The backbone consists of ResNet-50 architecture [20] initialized with the pre-trained Image-Net [28] weights. The offline experiments comprise the searching and training phases. The proposed CHASE tracker is implemented in PyTorch and runs 23 *FPS* on a single Nvidia Tesla V100 GPU with 16GB RAM. Except for the following details, the rest of the hyper-parameters are set to the ones in [40]. The test sets are never utilized in searching or training phases.

Searching Phase

In this phase, the cell architecture is searched by the modified 2nd-order DARTS. The cell includes 14 edges and 7 nodes (2 input, 4 intermediate, and 1 output), which the output node is obtained by depthwise concatenation of intermediate nodes. The standard DARTS search space is employed to exploit the maximum number of nodes & edges allowing in a cell, which provides the highest flexibility in cell design. The candidate operations include 3×3 & 5×5 separable convolutions, 3×3 & 5×5 dilated convolutions, 3×3 max pooling, 3×3 average pooling, zero (no connection), and skip connection (i.e., $\mathcal{M} = 8$). The CHASE applies operation-level Dropout, which its rate starts from $\tau = 0.6$ and gradually decayed to the last epoch. In contrast to [126, 127], the CHASE fairly explores all operations, considering the importance of skip-connections on the evaluation accuracy and architecture stability.

The training set of the TrackingNet dataset [9] is divided into two subsets for optimizing the weights of network ($\mathcal{W}$) & encoding weights of architecture (α) on the training ($\mathcal{L}_{tr}$) & validation ($\mathcal{L}_{val}$) sets, respectively. Besides, the training sets of GOT-10k [12] and LaSOT [8] datasets are used as the hold-out set ($\mathcal{L}_{ho}$) to specify the best architecture among three runs (with different random seeds) and select the final cell architecture based on their performance. Based on the training tricks of NAS in [135], the backbone and BBR parameters are frozen during architecture search, while the architecture parameters are started to optimize after 10 epochs. It is more critical for the proposed approach to calculate reliable 2nd-order gradients of architecture parameters built on 1st-order ones of network weights. The proposed CHASE provides better initialization of candidate operations directly impacting the optimization procedure of architecture parameters. Thus, it provides fair competition between weight-free operations with other ones and helps effective learning of architecture parameters, leading to performance improvement, acceleration, and avoiding getting stuck into bad local optima. The network is trained for at most 70 epochs with a batch size of 10, similar to the baseline [40]. However, the proposed approach stops the training procedure based on the proposed early-stopping strategy (epoch 41 for CHASE). The Adam optimizer [84] is used to learn network and architecture parameters. The initial learning rate is 0.001 for optimizing $\mathcal{W}$ with the cosine annealing scheduler. The maximum iteration numbers are 15K, 15K, and 5K for training, validation, and early-stopping procedures. The search phase takes about 41 (18) hours for the second (first) order DARTS method using the TrackingNet dataset on a Nvidia Tesla V100 GPU with 16GB RAM.

Training Phase

In contrast to prior works (e.g., [125–129, 131]), the CHASE just trains the best model selected in searching phase from scratch. In this phase, computational cell is replaced by the best cell architecture, and the whole network (including backbone, TCR, and BBR) is jointly trained from scratch for 70 epochs. The TCR and BBR layers are initialized with random weights ignoring the weights during the searching phase. For the training phase, the training sets of LaSOT [8], TrackingNet [9], GOT-10k [12], and COCO [30] datasets are used, similar to the baseline [40]. Also, other hyper-parameters are set as in the baseline tracker [40].

Evaluating Phase

After offline training phases, the proposed CHASE tracker is evaluated on test splits of generic and aerial visual tracking datasets, namely GOT-10k [12], TrackingNet [9], LaSOT [8], UAV-123 [53], and VisDrone-2019-test-dev [10]. In the online phase, all procedures and settings are the same as [40].

5.4.2 Ablation Analysis of CHASE

In this section, a systematic ablation analysis on the GOT-10k dataset [12] is conducted to validate the effectiveness of various search spaces and methods. It includes the cells derived by the 1) 1st-order DARTS (CHASE-D1), 2) Fair-DARTS [130]

(CHASE-FD), and 3) proposed approach (CHASE-PrDiMP or CHASE). Besides, the CHASE is integrated into the DiMP tracker [26] (CHASE-DiMP), demonstrating the generalization of the proposed approach for visual tracking. Furthermore, three versions of the proposed approach are investigated, including the CHASE with 1) fully segregated datasets in searching & training phases (CHASE-S/T), 2) a search space consisting of two intermediate nodes (CHASE-2N), and 3) a search space without weightless candidate operations (CHASE-WO). The comparison results are reported in Table 5.1 regarding the derived cells shown in Fig. 5.2.

Accordingly, the CHASE-D1 derives a cell dominated by weight-free operations (i.e., skip and pooling operations), and there is no connection between intermediate nodes resulting in a shallow architecture. The CHASE-FD employs the Fair-DARTS [130], which utilizes the Sigmoid activation function and an auxiliary loss to address exclusive competition of skip-connections and discretization discrepancy. Nonetheless, the CHASE outperforms the CHASE-D1 & CHASE-FD up to 3.6% and 2.1% in terms of *Average Overlap* (AO) metric, respectively. Conventional DARTS-based methods (with stacked cell networks for image classification) search a network archi-

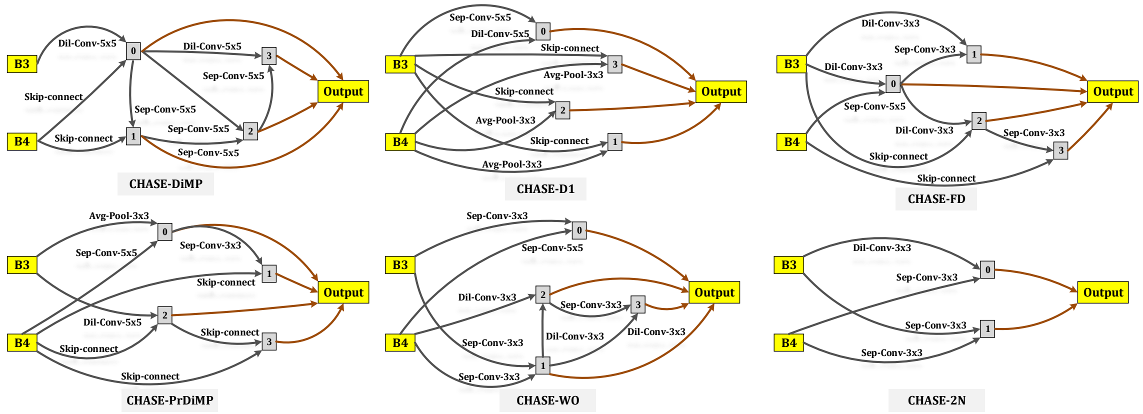

Figure 5.2: Best cell architectures derived by CHASE-DiMP (modified 2nd-order DARTS), CHASE-D1 (1st-order DARTS), CHASE-FD (Fair-DARTS), CHASE-PrDiMP (modified 2nd-order DARTS), CHASE-WO (modified 2nd-order DARTS without weightless operations), and CHASE-2N (modified 2nd-order DARTS with two intermediate nodes). B3 and B4 are the input latent representations (from `Block3` & `Block4` of Resnet50 [20], respectively). Also, 0, 1, 2, 3 are the intermediate nodes, and the output is the depthwise concatenation of intermediate nodes.

Table 5.1: Ablation analysis of CHASE on GOT-10k dataset [12].

Metric	DiMP [26]	**CHASE-DiMP**	PrDiMP [40]	CHASE-D1	CHASE-FD	**CHASE**	CHASE-2N	CHASE-WO	CHASE-S/T
$\mathrm{SR}_{0.75}$ (↑)	49.2	**51.1**	54.3	54.8	56.1	**56.5**	51.4	45.9	56.1
$\mathrm{SR}_{0.5}$ (↑)	71.7	**75.3**	73.8	76.7	76.8	**78.8**	76.5	71.5	76.3
AO (↑)	61.1	**63.6**	63.4	64.9	65.6	**67.0**	64.2	60.7	65.6

tecture on a small proxy dataset (e.g., CIFAR-10) and then transfer it to a large-scale target dataset (e.g., ImageNet) to alleviate high memory consumption [134]. However, the proposed approach can enjoy searching on the large-scale TrackingNet dataset by its cell-level search. Hence, the CHASE uses the large-scale TrackingNet dataset in both searching & training phases outperforming the CHASE-S/T up to 1.4% in terms of AO metric. Except for CHASE-S/T, all CHASE-versions have been searched and trained on similar datasets.

While the CHASE employs the standard DARTS search space to have more design flexibility via the maximum number of nodes & edges allowing in a cell, the CHASE-2N and CHASE-WO represent search spaces with limited node numbers (i.e., two intermediate nodes) and removed weightless candidate operations (i.e., pooling, zero, & skip connect), respectively. According to the results, the CHASE has improved the performance of CHASE-2N & CHASE-WO up to 2.8% & 6.3% in terms of AO metric, respectively. These results demonstrate prior heuristics and limited search space dramatically affect architecture design and tracking performance. For instance, the intuitive reason in the case of CHASE-WO is that removing weightless operations (particularly skip-connections) has been led to instability in cell design and accuracy degradation. Besides, the node restriction results in shallow cell architecture and limited performance improvement. The computational cells derived by the CHASE-PrDiMP confirm selecting various operations regarding objective function, increasing the depth as necessary, and preventing over-fitting and performance collapse problems. Finally, the proposed approach is integrated into the DiMP tracker [26] minimizing an L^2-based discriminative learning loss to train its network to investigate the generalization to different objective functions. The proposed approach

Table 5.2: State-of-the-art comparison results of CHASE on GOT-10k [12], LaSOT [8], TrackingNet [9], UAV-123 [53], VisDrone-2019-test-dev [10] datasets.

Trackers	GOT-10k			LaSOT			TrackingNet			UAV-123		VisDrone-2019-test-dev	
	AO (↑)	$SR_{0.5}$ (↑)	$SR_{0.75}$ (↑)	AUC (↑)	Norm. Prec. (↑)	Prec. (↑)	AUC (↑)	Norm. Prec. (↑)	Prec. (↑)	$SR_{0.5}$ (↑)	Prec. (↑)	AUC (↑)	Prec. (↑)
CHASE	**67.0**	**78.8**	**56.5**	**61.7**	**71.1**	**62.9**	**76.8**	**82.5**	**71.8**	**83.9**	**88.2**	**61.7**	82.0
LightTrack [124]	62.3	72.6	-	-	-	56.1	73.3	78.9	70.8	-	-	-	-
PrDiMP-50 [40]	63.4	73.8	**54.3**	**59.8**	**68.8**	**60.8**	**75.8**	81.6	70.4	**82.7**	**87.4**	59.8	79.7
Ocean [41]	61.1	72.1	47.3	56.0	65.1	56.6	-	-	-	-	-	59.4	82.3
D3S [111]	59.7	67.6	46.2	-	-	-	72.8	76.8	66.4	-	-	-	-
ROAM++ [140]	46.5	53.2	23.6	44.7	-	44.5	67.0	75.4	62.3	-	-	-	-
SiamAttn [45]	-	-	-	56.0	64.8	-	75.2	81.7	-	79.4	84.5	-	-
KYS [138]	63.6	75.1	51.5	55.4	63.3	-	74.0	80.0	68.8	-	-	-	-
DiMP-50 [26]	61.1	71.7	49.2	56.9	65.0	56.7	74.0	80.1	68.7	80.4	85.5	60.8	80.5
SiamCAR [141]	56.9	67.0	41.5	50.7	60.0	51.0	-	-	-	77.3	81.3	-	-
SiamBAN [142]	-	-	-	51.4	59.8	52.1	-	-	-	77.4	83.3	-	-
MAML [143]	-	-	-	52.3	-	-	75.7	**82.2**	**72.5**	-	-	-	-
ATOM [19]	55.6	63.4	40.2	51.5	57.6	50.5	70.3	77.1	64.8	78.9	85.6	57.1	76.7
SiamRPN++ [25]	51.8	61.8	32.5	49.6	56.9	-	73.3	80.0	69.4	78.8	84.0	59.9	79.1
DCFST [136]	**63.8**	**75.3**	49.8	-	-	-	75.2	80.9	70.0	-	-	-	-
COMET [44]	59.6	70.6	44.9	54.2	-	-	-	-	-	79.4	86.1	**64.5**	**83.9**
SiamFC++ [137]	59.5	69.5	47.9	54.4	62.3	54.7	75.4	80.0	70.5	-	-	-	-
SiamMask [24]	51.4	58.7	36.6	-	-	-	72.5	77.8	66.4	-	-	58.1	79.4
DaSiamRPN [34]	-	-	-	-	-	-	63.8	73.3	-	72.6	78.1	-	-
ECO [87]	31.6	30.9	11.1	32.4	33.8	30.1	55.4	61.8	49.2	63.1	74.1	55.9	**82.6**

outperforms the DiMP tracker [26] up to 2.5% in terms of the AO and up to 3.6% in terms of *Success Rate* (SR) at the overlap threshold of 0.5. At last, the best-performing tracker, CHASE, is selected to be compared with recent trackers in the next section.

5.4.3 State-of-the-art Comparison of CHASE

In this section, the state-of-the-art evaluations are performed on five large-scale visual tracking benchmarks and the proposed CHASE tracker is compared with various state-of-the-art visual trackers, namely ECO [87], SiamMask [24], DaSiamRPN [34], SiamRPN++ [25], ATOM [19], DCFST [136], COMET [44], SiamFC++ [137], DiMP-50 [26], PrDiMP-50 [40], KYS [138], SiamAttn [45], MAML [139], ROAM++ [140], SiamCAR [141], SiamBAN [142], D3S [111], Ocean [41], and LightTrack [124].

GOT-10k [12]: As noted in Sec. 2.2, the object classes of evaluation and training set of GOT-10K has no overlap. Hence, this dataset is usually used for studying the transferability of proposed approaches for tracking unseen targets. Therefore, the proposed CHASE uses its training set as one of the hold-out sets to early-stop the cell searching phase. The comparison results presented in Table 5.2 show that the

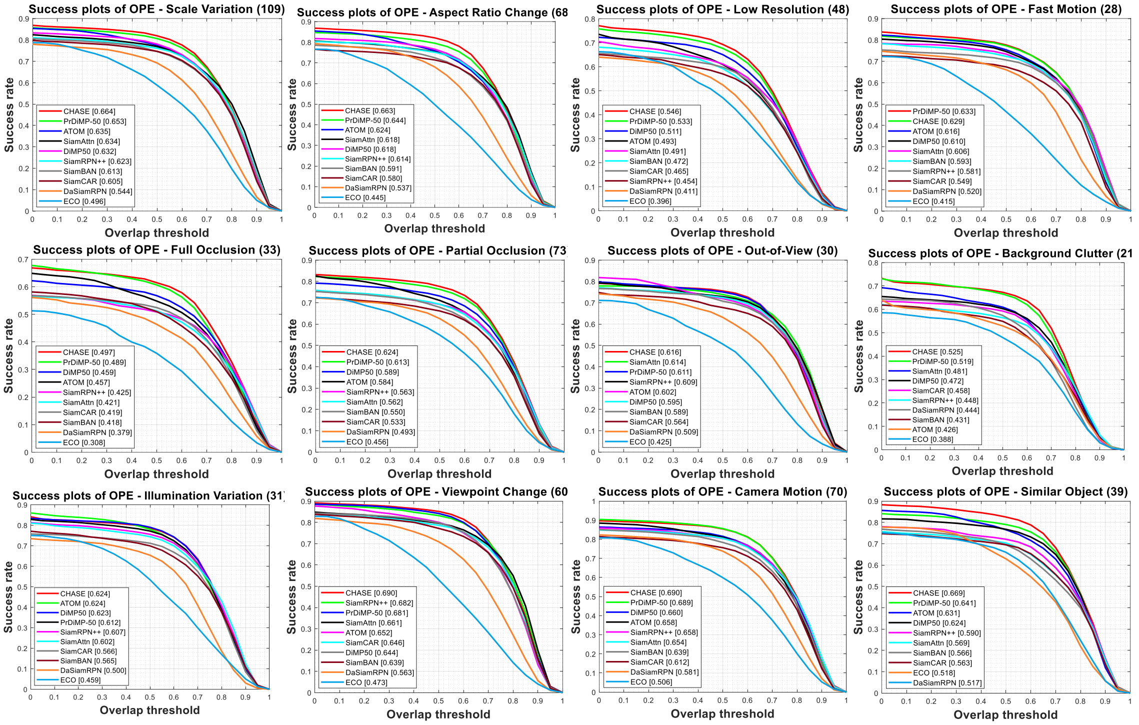

Figure 5.3: Attribute-based comparisons of the proposed method (CHASE) on UAV-123 dataset [53] in terms of AUC metric.

CHASE outperforms the baseline up to 3.6%, 5%, and 2.2% in terms of AO and SR at overlap thresholds of 0.5 and 0.75, respectively. Besides, the CHASE has achieved better results (4.7% in AO, 6.2% in $\text{SR}_{0.5}$) compared with the LightTrack [124].

LaSOT [8]: This datasets contains long-term tracking scenarios (see Sec. 2.2 for more details). Therefore, it appropriately indicates the robustness of short-term trackers in real-world situations. For this reason, the proposed tracker uses its training set as the second dataset of hold-out set in the searching phase. As shown in Table 5.2, the CHASE improves the baseline results [40] by a margin of 1.9%, 2.3%, and 2.1% in terms of *Area Under Curve* (AUC), normalized precision, and precision, respectively.

TrackingNet [9]: From Table 5.2, the MAML tracker [143] has close results (better in precision metric) compared with the proposed tracker since it employs a modern object detector (i.e., FCOS [42]) and online domain adaptation to enhance discriminating target from non-target regions. However, the proposed CHASE tracker has achieved better results in terms of AUC and normalized precision, and it has improved the baseline results by a margin of 1% in AUC and 1.4% in precision metric.

UAV-123 [53]: According to the results in Table 5.2, the proposed CHASE tracker outperforms the state-of-the-art visual trackers but also the baseline tracker [40] up to 1.2% and 0.8% in terms of success and precision rate metrics.

VisDrone-2019-test-dev [10]: This aerial view testing data contains challenging scenarios such as abrupt camera motion, tiny targets, fast view-point change, and day/night conditions. Compared with the baseline [40], the results of the CHASE tracker have improved up to 1.9% in AUC and 2.3% in precision rate. The COMET [44] has obtained the best results employing the training set of VisDrone for its offline training and accurately designed modules for small object tracking.

To provide a detailed evaluation in challenging scenarios, additional attribute-based results are presented in Fig. 5.3. Accordingly, the proposed CHASE tracker outperforms all the recent trackers based on the AUC metric on the aerial tracking dataset of UAV-123 [53].

5.5 Summary

A novel cell-level differentiable architecture search mechanism is proposed. To address the inherent limitations of differentiable architecture search, the modified second-order DARTS by early stopping to mitigate the over-fitting and performance collapse issues and operation-level dropout without any post-processing is introduced. The approach is simple, efficient, and easy to be integrated into existing visual trackers. Extensive experiments demonstrate the effectiveness of the proposed approach, as well as noticeable performance improvement when working with different existing trackers.

Chapter 6

Conclusions & Future Works

6.1 Conclusions

In this thesis, *Visual Object Tracking* using deep neural networks in unconstrained environments is studied. While the state-of-the-art object trackers provide fascinating performance over most of the publicly available videos with generic targets [4, 116, 117], they still have issues with handling challenging scenarios including small objects, camera motion, deformable objects, and object blur just to name a few. Moreover, the network architecture for the state-of-the-art trackers which benefit from deep neural networks is obtained manually using many trial and errors. Motivated by these, three research projects on small object tracking, segmentation tracking, and automatic designing of deep tracking networks are conducted.

To enhance the accuracy of existing trackers for small objects in videos captured by UAVs from medium to high altitudes, an offline proposal generation strategy and a Y-shape multi-task network are introduced. The proposal generation strategy aims at helping the network to learn target estimation using context information by providing target & its parts as the template model during offline training. The importance of this strategy is more pronounced in scenarios which only parts of objects are visible (e.g., occlusion and viewpoint change). Besides, the multi-task network employs the MSAF and attention modules to aggregate the refined & resized multi-scale features and highlight target-related information considering robustness and accuracy of predictions. Extensive experimental evaluations on object tracking benchmarks show the effectiveness of proposed strategies, maintaining state-of-the-art performance on all the existing aerial small/tiny object tracking benchmarks. Finally, a simple global search strategy is considered to enhance the robustness of our tracker in long-term scenarios.

Moreover, by developing a two-stage method consisting of bounding box tracking and segmentation steps, segmentation tracking is studied. The first part takes advantage of fusing the prediction of two trackers running in parallel to obtain the axis-aligned box fitted on the target. Then, the segmentation step obtains a coarse binary mask of the target from the predicted box and refines it to obtain the final mask.

Finally, a novel cell-level differentiable *Neural Architecture Search* (NAS) method to adapt the pre-trained backbone features to the objective of visual tracking for Siamese neural networks is proposed. The introduced cell-level NAS benefits from modified second-order DARTS and a proposed early-stopping strategy using a hold-out set to address the over-fitting, performance collapse, and requiring multiple retraining from scratch issues of differentiable NAS methods. Extensive experimental evaluations by integrating various differentiable NAS methods into Siamese framework for different tracking objectives demonstrate the noticeable performance improvement, and generalization of the proposed method.

6.2 Future Works

At the end of this study, we believe that the following areas in *Visual Object Tracking* and computer vision should be explored in the next few years.

Transformers were originally introduced for natural language processing using attention concept [144]. Recently, researchers have started to introduce transformers for different computer vision applications [145, 146] including object classification [147, 148], detection [149–151], semantic segmentation [152, 153], action recognition [154], and object tracking [46, 47]. Specifically for tracking, transformers can be used to provide pixel-level correspondence in temporal domain. For instance, they are already used to combine pixel-level information in the template model with search patch instead of widely used cross-correlation operation [47]. However, the convergence of original transformers is slow (e.g., TransT [47] tracker is trained for 1,000 epochs). Considering the recent advances of NAS methods, an interesting research direction to explore is using NAS to obtain modified lightweight transformer architectures for different computer vision tasks including *Visual Object Tracking*. However, we should find a solution for addressing memory limitation issues.

Also, the *Visual Object Tracking* community is showing interest in tracking the binary mask of target instead of the regular axis-aligned bounding box. To move a step forward, the goal will be tracking the 3D reconstruction of targets in publicly available videos. At this moment, there are a few published researches on 3D tracking of specific classes of targets including humans [155–157] and four-legged animals [158] by leveraging the prior knowledge. However, solving this problem in class-agnostic manner is highly challenging due to the importance of prior knowledge for 3D reconstruction. Using the idea of structure-from-motion [159] can be another solution, but 3D tracking should be solved in real-time to meet the expectations of tracking community. While assuming rigid-body transformations, the *Track to Reconstruct and Reconstruct to Track* [160] method is an inspiring research to solve this problem for autonomous driving applications. We believe that combining segmentation trackers with self-supervised monocular depth estimation [161, 162] can be potential solution to this problem.

Finally, we believe that computer vision tools will be widely used towards low-cost automated analysis of animals in the next few years. Accordingly, we are collaborating with *Guan Zhen* from *Institute of Molecular and Cell Biology* (IMCB), A*STAR to provide a zebrafish larvae segmentation tracking dataset with zebrafish-specific visual attributes (e.g., food, low resolution). Our DESTINE [3] tracker has been used to provide the initial coarse masks. Due to the fast movements of transparent zebrafish larvae, annotating the precise binary mask of zebrafish for the boundaries & tail is impossible. Accordingly, a human annotator is asked to refine the obtained coarse mask in order to provide the tri-mask ground truth which will be validated by zebrafish experts. Providing the ground truth for more than half of the total 18K frames has been already done. Also, we are providing a comprehensive literature review on applications of computer vision in animal studies. In this work, we have studied existing computer vision researches/tools for animal detection, tracking, segmentation, and pose- & shape-estimation with various real world applications including but not limited to using camera traps & UAVs for wildlife monitoring [163, 164], tracking & locomotion analysis of animals towards behavior & health monitoring [165, 166], concealed/ camouflaged object detection [167–169], transferring human keypoints to animals using limited annotated samples [170, 171], developing parametric shape models for animals [172–174], and obtaining species-specific shapes from fine-grained image collections [175, 176].

Bibliography

[1] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1442–1468, 2014.

[2] R. Pflugfelder, "An in-depth analysis of visual tracking with siamese neural networks," *arXiv preprint arXiv:1707.00569*, 2017.

[3] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, M. Danelljan, L. Č. Zajc, A. Lukežič, O. Drbohlav, L. He, Y. Zhang, S. Yan, J. Yang, G. Fernández, and et al., "The eighth visual object tracking VOT2020 challenge results," in *Proc. ECCVW*, 2020, pp. 547–601.

[4] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *IEEE Trans. Intell Transp Syst*, pp. 1–26, 2021. DOI: 10.1109/TITS.2020.3046478.

[5] X. Shen, "A survey of object classification and detection based on 2d/3d data," *arXiv preprint arXiv:1905.12683*, 2019.

[6] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International journal of computer vision*, vol. 128, no. 2, pp. 261–318, 2020.

[7] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[8] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE CVPR*, 2019.

[9] M. Müller, A. Bibi, S. Giancola, S. Alsubaihi, and B. Ghanem, "TrackingNet: A large-scale dataset and benchmark for object tracking in the wild," in *Proc. ECCV*, 2018, pp. 310–327.

[10] D. Du, P. Zhu, L. Wen, X. Bian, H. Ling, and et al., "VisDrone-SOT2019: The Vision Meets Drone Single Object Tracking Challenge Results," in *Proc. ICCVW*, 2019.

[11] H. Fan, L. Wen, D. Du, P. Zhu, Q. Hu, H. Ling, M. Shah, B. Wang, B. Dong, D. Yuan, and et al., "Visdrone-sot2020: The vision meets drone single object tracking challenge results," in *Proc. ICCVW*, Springer, 2020, pp. 728–749.

[12] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, 2021.

[13] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106 622, 2021, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.106622. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120307516.

[14] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[15] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–34, 2021.

[16] M. Y. Abbass, K.-C. Kwon, N. Kim, S. A. Abdelwahab, F. E. A. El-Samie, and A. A. Khalaf, "A survey on online learning for visual tracking," *The Visual Computer*, vol. 37, pp. 993–1014, 2021.

[17] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE CVPR*, 2016, pp. 4293–4302.

[18] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *Proc. IEEE CVPRW*, 2017, pp. 2217–2224.

[19] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE CVPR*, 2019.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[21] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proc. IEEE ECCV*, 2018, pp. 816–832.

[22] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE CVPR*, 2010, pp. 2544–2550.

[23] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE CVPR*, 2018, pp. 8971–8980.

[24] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE CVPR*, 2019.

[25] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," in *Proc. IEEE CVPR*, 2019.

[26] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. ICCV*, 2019, pp. 6181–6190.

[27] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *Proc. ECCV*, 2016, pp. 850–865.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[29] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video," in *Proc. IEEE CVPR*, 2017, pp. 7464–7473.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.

[31] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. ECCV*, 2016, pp. 749–765.

[32] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE CVPR*, 2017, pp. 5000–5008.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[34] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. ECCV*, vol. 11213 LNCS, 2018, pp. 103–119.

[35] Z. Zhang and H. Peng, *Deeper and wider Siamese networks for real-time visual tracking*, 2019. eprint: arXiv:1901.01660. [Online]. Available: http://arxiv.org/abs/1901.01660.

[36] H. Fan and H. Ling, *Siamese cascaded region proposal networks for real-time visual tracking*, 2018. eprint: arXiv:1812.06148. [Online]. Available: http://arxiv.org/abs/1812.06148.

[37] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic Siamese network for visual object tracking," in *Proc. IEEE ICCV*, 2017, pp. 1781–1789.

[38] L. Zhang, A. Gonzalez-Garcia, J. v. d. Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for siamese trackers," in *Proc. IEEE ICCV*, 2019.

[39] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "Gradnet: Gradient-guided network for visual object tracking," in *Proc. IEEE ICCV*, 2019.

[40] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE CVPR*, 2020.

[41] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, “Ocean: Object-aware anchor-free tracking,” in *Proc. ECCV*, 2020.

[42] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in *Proc. ICCV*, 2019.

[43] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, “BAM: Bottleneck attention module,” in *Proc. BMVC*, 2018, pp. 147–161.

[44] S. M. Marvasti-Zadeh, J. Khaghani, H. Ghanei-Yakhdan, S. Kasaei, and L. Cheng, “COMET: Context-aware IoU-guided network for small object tracking,” in *Proc. ACCV*, 2020.

[45] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, “Deformable siamese attention networks for visual object tracking,” in *Proc. IEEE CVPR*, 2020.

[46] N. Wang, W. Zhou, J. Wang, and H. Li, “Transformer meets tracker: Exploiting temporal context for robust visual tracking,” in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[47] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, “Transformer tracking,” in *CVPR*, 2021.

[48] Y. Wu, J. Lim, and M. Yang, “Object tracking benchmark,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.

[49] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, and et al., “The visual object tracking VOT2013 challenge results,” in *Proc. ICCV*, 2013, pp. 98–111.

[50] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, and et al., “The visual object tracking VOT2014 challenge results,” in *Proc. ECCV*, 2015, pp. 191–217.

[51] P. Liang, E. Blasch, and H. Ling, “Encoding color information for visual tracking: Algorithms and benchmark,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, 2015.

[52] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, “Need for speed: A benchmark for higher frame rate object tracking,” in *Proc. IEEE ICCV*, 2017, pp. 1134–1143.

[53] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for UAV tracking,” in *Proc. ECCV*, 2016, pp. 445–461.

[54] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, “The unmanned aerial vehicle benchmark: Object detection and tracking,” in *Proc. ECCV*, 2018, pp. 375–391.

[55] H. Yu, G. Li, W. Zhang, and et al., “The unmanned aerial vehicle benchmark: Object detection, tracking and baseline,” *Int. J. Comput. Vis.*, 2019.

[56] C. Liu, W. Ding, J. Yang, and et al., “Aggregation signature for small object tracking,” *IEEE Trans. Image Processing*, vol. 29, pp. 1738–1747, 2020.

[57] P. Zhu, L. Wen, D. Du, and et al., “VisDrone-VDT2018: The vision meets drone video detection and tracking challenge results,” in *Proc. ECCVW*, 2018, pp. 496–518.

[58] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in *Proc. ECCV*, 2016, pp. 21–37.

[59] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. Berg, *DSSD: Deconvolutional single shot detector*, 2017. eprint: arXiv:1701.06659. [Online]. Available: https://arxiv.org/abs/1701.06659.

[60] L. Cui, R. Ma, P. Lv, X. Jiang, Z. Gao, B. Zhou, and M. Xu, *MDSSD: Multi-scale deconvolutional single shot detector for small objects*, 2018. eprint: arXiv:1805.07009v3. [Online]. Available: https://arxiv.org/abs/1805.07009v3.

[61] J. Lim, M. Astrid, H. Yoon, and S. Lee, *Small object detection using context and attention*, 2019. eprint: arXiv:1912.06319v2. [Online]. Available: https://arxiv.org/abs/1912.06319v2.

[62] X. Yang, J. Yang, J. Yan, Y. Zhang, T. Zhang, Z. Guo, X. Sun, and K. Fu, “SCRDet: Towards more robust detection for small, cluttered and rotated objects,” in *Proc. IEEE ICCV*, 2019.

[63] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, 2018. eprint: arXiv:1804.02767. [Online]. Available: https://arxiv.org/abs/1804.02767.

[64] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, “Learning aberrance repressed correlation filters for real-time UAV tracking,” in *Proc. IEEE ICCV*, 2019, pp. 2891–2900.

[65] C. Fu, Z. Huang, Y. Li, R. Duan, and P. Lu, “Boundary effect-aware visual tracking for UAV with online enhanced background learning and multi-frame consensus verification,” in *Proc. IROS*, 2019, pp. 4415–4422.

[66] Y. Li, C. Fu, Z. Huang, Y. Zhang, and J. Pan, “Keyfilter-aware real-time uav object tracking,” in *Proc. ICRA*, 2020.

[67] F. Li, C. Fu, F. Lin, Y. Li, and P. Lu, “Training-set distillation for real-time UAV object tracking,” in *Proc. ICRA*, 2020, pp. 1–7.

[68] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, “AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization,” in *Proc. IEEE CVPR*, 2020.

[69] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, “’skimming-perusal’ tracking: A framework for real-time and robust long-term tracking,” in *Proc. IEEE ICCV*, 2019.

[70] L. Huang, X. Zhao, and K. Huang, “Globaltrack: A simple and strong baseline for long-term tracking,” in *Proc. AAAI*, 2020.

[71] P. Voigtlaender, J. Luiten, P. H. Torr, and B. Leibe, “Siam r-cnn: Visual tracking by re-detection,” in *Proc. IEEE CVPR*, 2020.

[72] K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, and X. Yang, “High-performance long-term tracking with meta-updater,” in *Proc. IEEE CVPR*, 2020.

[73] I. Jung, J. Son, M. Baek, and B. Han, “Real-time MDNet,” in *Proc. ECCV*, 2018, pp. 89–104.

[74] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, and et al., “The sixth visual object tracking VOT2018 challenge results,” in *Proc. ECCVW*, 2019, pp. 3–53.

[75] M. Kristan and et al., “The seventh visual object tracking VOT2019 challenge results,” in *Proc. ICCVW*, 2019.

[76] J. Valmadre, L. Bertinetto, J. F. Henriques, R. Tao, A. Vedaldi, A. W. Smeulders, P. H. Torr, and E. Gavves, “Long-term tracking in the wild: A benchmark,” in *Proc. ECCV*, vol. 11207 LNCS, 2018, pp. 692–707.

[77] A. Moudgil and V. Gandhi, “Long-term visual object tracking benchmark,” in *Proc. ICCV*, 2018, pp. 629–645.

[78] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE CVPR*, 2014, pp. 580–587.

[79] J.-S. Lim, M. Astrid, H.-J. Yoon, and S.-I. Lee, *Small object detection using context and attention*, 2019. eprint: arXiv:1912.06319v2. [Online]. Available: https://arxiv.org/abs/1912.06319.

[80] K. Tong, Y. Wu, and F. Zhou, “Recent advances in small object detection based on deep learning: A review,” *Image and Vision Computing*, vol. 97, 2020.

[81] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. CVPR*, 2016, pp. 2818–2826.

[82] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. DOI: 10.1109/TPAMI.2019.2913372.

[83] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification,” in *Proc. ICCV*, 2015, pp. 1026–1034.

[84] D. P. Kingma and J. Ba, “ADAM: A method for stochastic optimization,” in *Proc. ICLR*, 2014.

[85] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M. H. Yang, “CREST: Convolutional residual learning for visual tracking,” in *Proc. ICCV*, 2017, pp. 2574–2583.

[86] H. Fan and H.Ling, “Parallel tracking and verifying,” *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4130–4144, 2019.

[87] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE CVPR*, 2017, pp. 6931–6939.

[88] T. Zhang, C. Xu, and M. H. Yang, "Multi-task correlation particle filter for robust object tracking," in *Proc. IEEE CVPR*, 2017, pp. 4819–4827.

[89] Y. Zhang, J. Zhang, and X. Guo, "Kindling the darkness: A practical low-light image enhancer," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, pp. 1632–1640.

[90] N. Ahn, B. Kang, and K.-A. Sohn, "Efficient deep neural network for photo-realistic image super-resolution," *arXiv preprint arXiv:1903.02240*, 2019.

[91] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, IEEE, 2016, pp. 3464–3468.

[92] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 4, pp. 1–47, 2020.

[93] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, "Youtube-vos: A large-scale video object segmentation benchmark," *arXiv preprint arXiv:1809.03327*, 2018.

[94] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset, "The 2018 davis challenge on video object segmentation," *arXiv preprint arXiv:1803.00557*, 2018.

[95] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[96] Y. Zhang, Z. Wu, H. Peng, and S. Lin, "A transductive approach for video object segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6949–6958.

[97] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele, "Lucid data dreaming for video object segmentation," *International Journal of Computer Vision*, vol. 127, no. 9, pp. 1175–1197, 2019.

[98] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "Video object segmentation without temporal information," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 6, pp. 1515–1530, 2018.

[99] P. Voigtlaender and B. Leibe, "Online adaptation of convolutional neural networks for video object segmentation," *BMVC*, 2017.

[100] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, "One-shot video object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 221–230.

[101] T. Meinhardt and L. Leal-Taixe, “Make one-shot video object segmentation efficient again,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[102] A. Robinson, F. J. Lawin, M. Danelljan, F. S. Khan, and M. Felsberg, “Learning fast and robust target models for video object segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[103] Y.-T. Hu, J.-B. Huang, and A. G. Schwing, “Videomatch: Matching based video object segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 54–70.

[104] H. Seong, S. W. Oh, J.-Y. Lee, S. Lee, S. Lee, and E. Kim, “Hierarchical memory matching network for video object segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 889–12 898.

[105] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim, “Video object segmentation using space-time memory networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9226–9235.

[106] X. Lu, W. Wang, M. Danelljan, T. Zhou, J. Shen, and L. Van Gool, “Video object segmentation with episodic graph memory networks,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, Springer, 2020, pp. 661–679.

[107] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, “Feelvos: Fast end-to-end embedding learning for video object segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9481–9490.

[108] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung, “Learning video object segmentation from static images,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2663–2672.

[109] P. Hu, J. Liu, G. Wang, V. Ablavsky, K. Saenko, and S. Sclaroff, “Dipnet: Dynamic identity propagation network for video object segmentation,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1904–1913.

[110] B. Zhao, G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, “Generating masks from boxes by mining spatio-temporal consistencies in videos,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 13 556–13 566.

[111] A. Lukezic, J. Matas, and M. Kristan, “D3S - A discriminative single shot segmentation tracker,” in *Proc. IEEE CVPR*, 2020.

[112] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *European conference on computer vision*, Springer, 2016, pp. 75–91.

[113] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 724–732.

[114] Z. Yang, S. Liu, H. Hu, L. Wang, and S. Lin, “Reppoints: Point set representation for object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9657–9666.

[115] S. M. Marvasti-Zadeh, J. Khaghani, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, “Chase: Robust visual tracking via cell-level differentiable neural architecture search,” *BMVC*, 2021.

[116] P. Li, D. Wang, L. Wang, and H. Lu, “Deep visual tracking: Review and experimental comparison,” *Pattern Recognit.*, vol. 76, pp. 323–338, 2018.

[117] R. Pflugfelder, *An in-depth analysis of visual tracking with Siamese neural networks*, 2017. eprint: arXiv:1707.00569. [Online]. Available: http://arxiv.org/abs/1707.00569.

[118] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, “DetNAS: Backbone search for object detection,” in *Proc. NeurIPS*, 2019.

[119] N. Wang, Y. Gao, H. Chen, P. Wang, Z. Tian, C. Shen, and Y. Zhang, “NAS-FCOS: Fast neural architecture search for object detection,” in *Proc. IEEE CVPR*, 2020.

[120] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A. L. Yuille, and L. Fei-Fei, “Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation,” in *Proc. IEEE CVPR*, 2019.

[121] V. Nekrasov, H. Chen, C. Shen, and I. Reid, “Fast neural architecture search of compact semantic segmentation models via auxiliary cells,” in *Proc. IEEE CVPR*, 2019.

[122] X. Yang, J. Fan, C. Wu, D. Zhou, and T. Li, “Nasmamsr: A fast image super-resolution network based on neural architecture search and multiple attention mechanism,” *Multimedia Systems*, pp. 1–14, 2021.

[123] Y. Zhu and E. Meijering, “Automatic improvement of deep learning-based cell segmentation in time-lapse microscopy by neural architecture search,” *Bioinformatics*, 2021.

[124] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, *LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search*, 2021. eprint: arXiv:2104.14545. [Online]. Available: http://arXiv:2104.14545.

[125] H. Liu, K. Simonyan, and Y. Yang, “DARTS: Differentiable architecture search,” in *Proc. ICLR*, 2019.

[126] X. Chen, L. Xie, J. Wu, and Q. Tian, “Progressive differentiable architecture search: Bridging the depth gap between search and evaluation,” in *Proc. ICCV*, 2019, pp. 1294–1303.

[127] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive DARTS: Bridging the optimization gap for nas in the wild," *Int J Comput Vis*, vol. 129, 638–655, 2021.

[128] L. Hanwen, S. Zhang, J. Sun, X. He, W. Huang, K. Zhuang, and Z. Li, *DARTS+: Improved differentiable architecture search with early stopping*, 2020. eprint: arXiv:1909.06035. [Online]. Available: http://arXiv:1909.06035.

[129] X. Chu, X. Wang, B. Zhang, S. Lu, X. Wei, and J. Yan, "DARTS-: Robustly stepping out of performance collapse without indicators," in *Proc. ICLR*, 2021.

[130] X. Chu, T. Zhou, B. Zhang, and J. Li, "Fair DARTS: Eliminating unfair advantages in differentiable architecture search," in *Proc. ECCV*, 2020.

[131] A. Zela, T. Elsken, T. Saikia, Y. Marrakchi, T. Brox, and F. Hutter, "Understanding and robustifying differentiable architecture search," in *Proc. ICLR*, 2020.

[132] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, 2020.

[133] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, 1999.

[134] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," in *Proc. ICLR*, 2019.

[135] H. Xu, L. Yao, W. Zhang, X. Liang, and Z. Li, "Auto-FPN: Automatic network architecture adaptation for object detection beyond classification," in *Proc. ICCV*, 2019.

[136] L. Zheng, M. Tang, Y. Chen, J. Wang, and H. Lu, "Learning feature embeddings for discriminant model based tracking," in *Proc. ECCV*, 2020.

[137] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI*, 2020, pp. 12 549–12 556.

[138] G. Bhat, M. Danelljan, L. Van Gool, and T. Radu, "Know your surroundings: Exploiting scene information for object tracking," in *Proc. ECCV*, 2020.

[139] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. ICML*, 2017, pp. 1126–1135.

[140] T. Yang, P. Xu, R. Hu, H. Chai, and A. B. Chan, "ROAM: Recurrently optimizing tracking model," in *Proc. IEEE CVPR*, 2020.

[141] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE CVPR*, 2020.

[142] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE CVPR*, 2020.

[143] G. Wang, C. Luo, X. Sun, Z. Xiong, and W. Zeng, “Tracking by instance detection: A meta-learning approach,” in *Proc. IEEE CVPR*, 2020.

[144] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[145] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” *arXiv preprint arXiv:2101.01169*, 2021.

[146] *Transformer-in-vision*, https://github.com/Yangzhangcst/Transformer-in-Computer-Vision, Accessed: 2021-07-11.

[147] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.

[148] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, “Understanding robustness of transformers for image classification,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[149] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European Conference on Computer Vision*, Springer, 2020, pp. 213–229.

[150] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” *ICLR*, 2021.

[151] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, and L. Zhang, “Dynamic detr: End-to-end object detection with dynamic attention,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2988–2997.

[152] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, “Segmenter: Transformer for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[153] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[154] R. Girdhar, J. a. Carreira, C. Doersch, and A. Zisserman, “Video Action Transformer Network,” in *CVPR*, 2019.

[155] S. Yuan, X. Li, and Y. Fang, “Deeptracking-net: 3d tracking with unsupervised learning of continuous flow,” *arXiv preprint arXiv:2006.13848*, 2020.

[156] M. Kocabas, N. Athanasiou, and M. J. Black, “Vibe: Video inference for human body pose and shape estimation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[157] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, “Learning 3d human dynamics from video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5614–5623.

[158] B. Biggs, T. Roddick, A. Fitzgibbon, and R. Cipolla, “Creatures great and smal: Recovering the shape and motion of animals from video,” in *Asian Conference on Computer Vision*, Springer, 2018, pp. 3–19.

[159] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[160] J. Luiten, T. Fischer, and B. Leibe, “Track to reconstruct and reconstruct to track,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1803–1810, 2020.

[161] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, “Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8977–8986.

[162] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, “Unsupervised monocular depth learning in dynamic scenes,” *arXiv preprint arXiv:2010.16404*, 2020.

[163] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, “Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning,” *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, E5716–E5725, 2018.

[164] N. Rey, M. Volpi, S. Joost, and D. Tuia, “Detecting animals in african savanna with uavs and the crowds,” *Remote Sensing of Environment*, vol. 200, pp. 341–351, 2017.

[165] V. Štih, L. Petrucco, A. M. Kist, and R. Portugues, “Stytra: An open-source, integrated system for stimulation, tracking and closed-loop behavioral experiments,” *PLoS computational biology*, vol. 15, no. 4, e1006699, 2019.

[166] L. Bergamini, S. Pini, A. Simoni, R. Vezzani, S. Calderara, R. B. Eath, and R. B. Fisher, “Extracting accurate long-term behavior changes from a large pig dataset,” in *16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2021*, SciTePress, 2021, pp. 524–533.

[167] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, “Camouflaged object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2777–2787.

[168] Y. Lyu, J. Zhang, Y. Dai, A. Li, B. Liu, N. Barnes, and D.-P. Fan, “Simultaneously localize, segment and rank the camouflaged objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[169] D. Fan, G. Ji, M. Cheng, and L. Shao, “Concealed object detection,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, 5555, ISSN: 1939-3539.

[170] T. Nath*, A. Mathis*, A. C. Chen, A. Patel, M. Bethge, and M. W. Mathis, “Using deeplabcut for 3d markerless pose estimation across species and behaviors,” *Nature Protocols*, 2019.

[171] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge, “Deeplabcut: Markerless pose estimation of user-defined body parts with deep learning,” *Nature Neuroscience*, 2018.

[172] B. Biggs, O. Boyne, J. Charles, A. Fitzgibbon, and R. Cipolla, “Who left the dogs out? 3d animal reconstruction with expectation maximization in the loop,” in *Proc. ECCV*, 2020.

[173] Y. Wang, N. Kolotouros, K. Daniilidis, and M. Badger, “Birds of a feather: Capturing avian shape models from images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 739–14 749.

[174] S. Zuffi, A. Kanazawa, T. Berger-Wolf, and M. J. Black, “Three-d safari: Learning to estimate zebra pose, shape, and texture from images ”in the wild”,” in *Proc. ICCV*, 2019.

[175] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, “Learning category-specific mesh reconstruction from image collections,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 371–386.

[176] S. Goel, A. Kanazawa, and J. Malik, “Shape and viewpoint without keypoints,” in *Proc. ECCV*, 2020, pp. 88–104.