sensors

MDPI

Review

Benchmarking Deep Trackers on Aerial Videos

Abu Md Niamul Taufique, Breton Minnehan † and Andreas Savakis *

Rochester Institute of Technology, Rochester, NY 14623, USA; at7133@rit.edu (A.M.N.T.);
breton.minnehan.1@us.af.mil (B.M.)
* andreas.savakis@rit.edu
† Current address: Air Force Research Laboratory, WPAFB, OH 45433, USA.

Received: 14 December 2019; Accepted: 10 January 2020; Published: 19 January 2020

Abstract: In recent years, deep learning-based visual object trackers have achieved state-of-the-art performance on several visual object tracking benchmarks. However, most tracking benchmarks are focused on ground level videos, whereas aerial tracking presents a new set of challenges. In this paper, we compare ten trackers based on deep learning techniques on four aerial datasets. We choose top performing trackers utilizing different approaches, specifically tracking by detection, discriminative correlation filters, Siamese networks and reinforcement learning. In our experiments, we use a subset of OTB2015 dataset with aerial style videos; the UAV123 dataset without synthetic sequences; the UAV20L dataset, which contains 20 long sequences; and DTB70 dataset as our benchmark datasets. We compare the advantages and disadvantages of different trackers in different tracking situations encountered in aerial data. Our findings indicate that the trackers perform significantly worse in aerial datasets compared to standard ground level videos. We attribute this effect to smaller target size, camera motion, significant camera rotation with respect to the target, out of view movement, and clutter in the form of occlusions or similar looking distractors near tracked object.

Keywords: visual object tracking; correlation filters; siamese networks; deep learning

1. Introduction

Visual object tracking is an important area of computer vision with applications in robotics [1], autonomous driving [2,3], video surveillance [4], pose estimation [5], medicine [6,7], activity recognition [8], and many others. Visual object tracking refers to locating a region of interest, typically a bounding box around the tracked object, in a sequence of frames. Visual tracking is challenging due to variations in appearance, illumination and scaling, changes in zoom, rotation, distortions, occlusion, abrupt motion, similar looking distractors, out of frame movement, etc.

Early visual tracking methods relied on hand crafted features, such as optical flow, and keypoints, and related benchmarking studies analyzed their performance [9–25]. Popular trackers such as Kernelized Correlation Filters (KCF) [20], Structured output tracking with Kernels (STRUCK) [21], Spatially Regularized Discriminative Correlation Filters (SRDCF) [23], and Background-Aware Correlation Filters (BACF) [22] used hand-crafted features. However, traditional methods may fail in challenging situations, such as those encountered in the 2018 Visual Object Tracking (VOT 2018) challenge [26] and the 2015 Object Tracking Benchmark (OTB 2015) challenge [27]. In recent years, deep learning has advanced object detection and other computer vision tasks, including tracking, semantic segmentation, pose estimation, visual question answering, and style transfer. In the VOT 2018 challenge, almost all of the top performing trackers used deep learning features based on convolutional neural networks (CNN) [26]. In our study, we focus on trackers based on such deep learning features.

Recent CNN based trackers can be broadly categorized into four groups, as illustrated in Figure 1: (i) Tracking by detection (TD), (ii) Correlation Filters (CF), (iii) Siamese networks (SN), and (iv)

Reinforcement learning (RL). In CF based trackers, correlation filters are learned to match the target distribution aiming for a response that is Gaussian distributed. The implementation of the CF trackers is usually done in the Fourier domain for computational efficiency and filters are updated during online tracking. However, the filter update reduces the speed of the tracking procedure [23,28,29]. SN-based trackers approach tracking as a similarity learning problem, where matching is done in feature space. SN trackers are trained offline and are not updated online, which is efficient but may reduce tracking performance [30–32]. TD-based trackers treat tracking as a binary classification problem that aims to separate the target from the background. Multiple patches are taken in the target frame (tth frame) near the target location in the previous frame ($(t-1)$th frame) and the patch with the highest score is selected as the target patch [33–35]. RL-based trackers learn an optimal path to the tracked object, either by moving the predicted location to the target location or by learning hyperparameters for tracking [36,37].

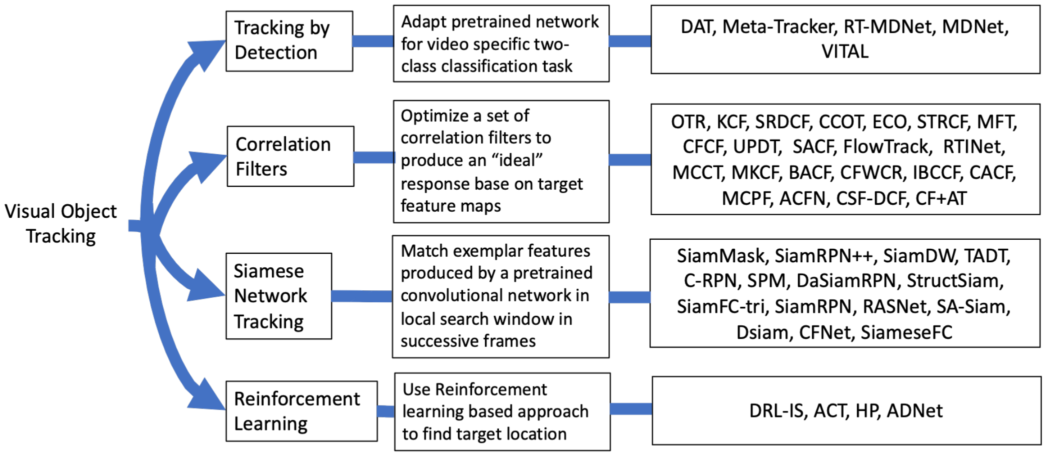

Visual Object Tracking
Tracking by Detection
Adapt pretrained network for video specific two-class classification task
DAT, Meta-Tracker, RT-MDNet, MDNet, VITAL
Correlation Filters
Optimize a set of correlation filters to produce an "ideal" response base on target feature maps
OTR, KCF, SRDCF, CCOT, ECO, STRCF, MFT, CFCF, UPDT, SACF, FlowTrack, RTINet, MCCT, MKCF, BACF, CFWCR, IBCCF, CACF, MCPF, ACFN, CSF-DCF, CF+AT
Siamese Network Tracking
Match exemplar features produced by a pretrained convolutional network in local search window in successive frames
SiamMask, SiamRPN++, SiamDW, TADT, C-RPN, SPM, DaSiamRPN, StructSiam, SiamFC-tri, SiamRPN, RASNet, SA-Siam, Dsiam, CFNet, SiameseFC
Reinforcement Learning
Use Reinforcement learning based approach to find target location
DRL-IS, ACT, HP, ADNet

Figure 1. Grouping of convolutional neural network (CNN)-based visual object tracking algorithms.

Evaluating tracking algorithms requires large and diverse datasets with annotated ground truth of the target location at every frame. Additionally, attribute annotation is important to fully assess tracker performance in different challenging situations, such as occlusion, illumination variation, etc. There are several tracking benchmarks in the literature [10,26,27,38–54]. Two of the most popular tracking benchmarks are the OTB [27,38] and the VOT challenge [26,39–43] datasets. In our study we benchmark various trackers on aerial datasets, another important category of datasets that consist of sequences taken from aerial platforms. We consider a video to be aerial if the camera is on an airplane or unmanned aerial vehicle (UAV). To include a greater variety of datasets, we broaden this definition to include cases where the camera is located anywhere above the ground ranging from a roof to a satellite.

There are some key factors that make aerial tracking different from tracking in standard ground level videos. In the aerial videos, the area covered by the field of view of the camera is usually much larger than that of ground level videos. More importantly, in aerial videos, the tracked object is much smaller in size in terms of pixels. Due to the smaller object size, it is more difficult to generate discriminative features, which adversely affects tracking performance. The object's size and viewpoint can change significantly and quickly in aerial videos. Additionally, the tracked object may be occluded for a long time and even disappear for several frames. Camera motion often causes abrupt changes in the object appearance and may result in out-of-frame movement for the tracked object. These characteristics present a unique set of challenges for aerial tracking compared to standard tracking on ground level videos. Initial work by Minnehan et al. [55] indicated that the performance of deep trackers varies significantly when tracking in aerial videos. Popular trackers, such as tracking by detection MDNet [33], or CF-based CCOT [29], were not as effective when tracking in aerial videos due to occlusion, smaller target size, and camera motion [55]. Our study differs from [55] in various aspects. We consider a larger number of more recent trackers from four general groups. We perform our benchmarking on multiple datasets and conduct detailed evaluation for various attributes of aerial imagery.

Several review papers exist in the literature that overview visual object tracking research [9,10,56,57]. Li et al. [56] studied deep learning trackers based on network architecture, network training, and network function and ran experiments on OTB100, TC-128 [46], and VOT2015 [41] to compare different deep learning based trackers. Fiaz et al. [57] performed an extensive review that compared various trackers based on different feature extraction methods. Trackers based on both deep learning and hand crafted features were evaluated on benchmarks, such as OTB 2015, OTB 2013 [38], TC-128, OTTC [57], and VOT 2017 [39]. VOT challenges compare many different trackers and provide a good overview of the performance of recent trackers.

However, these review papers and tracker benchmark studies deal with datasets that are ground based. In this work, we focus on aerial tracking because the conditions encountered vary significantly from ground level situations. The main contributions of this paper can be described as follows.

- We focus on aerial tracking using videos taken from aerial platforms. To our knowledge, this is the first comprehensive benchmarking study of visual object tracking on aerial videos.
- We benchmark ten recent deep learning based trackers from four tracking groups.
- We consider four different aerial benchmarks and compare the trackers' performance in various challenging situations which provides a better understanding of the state-of-the-art of visual object tracking on aerial videos.

The rest of the paper is organized as follows. In Section 2, we discuss the relevant tracking algorithms we incorporated in our benchmarking. In Section 3, we discuss the datasets used in our experiments and the evaluation metrics. In Section 4, we show the evaluation results on different benchmarks and discuss the comparison among different trackers on different benchmarks as well as specific attributes present within the datasets. In Section 5, we present final remarks based on our evaluation.

2. Tracking Algorithms

In this section, we overview the tracking algorithms based on their approach to tracking and network architectures. We only selected trackers that use deep learning features and considered their performance and source code availability.

A brief description follows for the four types of trackers outlined in Figure 1.

2.1. TD-Based Trackers

Tracking by detection frameworks view tracking as a foreground (target) vs. background classification problem. TD-based methods have been used for some time and continue to be popular with deep trackers in recent years [33–35,58–62]. In these frameworks, the networks generally learn the region of interest by sampling multiple patches from the input image. Positive and negative instances within the training samples are selected based on the intersection-over-union (IoU) score with the ground truth. Online training is usually done in these networks to improve the classification accuracy during tracking.

The TD-based trackers that we included in our benchmarking are MDNet [33], DAT [35], Meta-Tracker [59], and RT-MDNet [34]. MDNet was the winner of the VOT2015 challenge and was used as a baseline tracker for the other TD-based trackers. DAT improved over MDNet using reciprocation learning. The Meta-Tracker improved over MDNet using a meta-learning approach and RT-MDNet improved the real time performance of MDNet. Short descriptions of these trackers are given below.

2.1.1. Multidomain Network Tracker (MDNet)

In visual object tracking, there are some desirable properties for target representation learning such as invariance with respect to illumination, scale, perspective, and motion blur. The goal of using multidomain learning is to learn a discriminative model that learns a shared representation of the target in various domains. To achieve this, MDNet is trained offline with large set of video sequences, where each sequence is considered as a domain.

The MDNet architecture is shown in Figure 2. In the network shown, the domain specific layers are shown as FC6_1 to FC6_n where all the preceding layers are considered as shared layers. During tracking, all branches of the sixth fully connected layer are removed and replaced with a single fully connected layer, where online adaptation is performed by fine-tuning the fully connected layers. For precise localization of the object during tracking, a bounding box regression technique is used on bounding boxes with high scores.

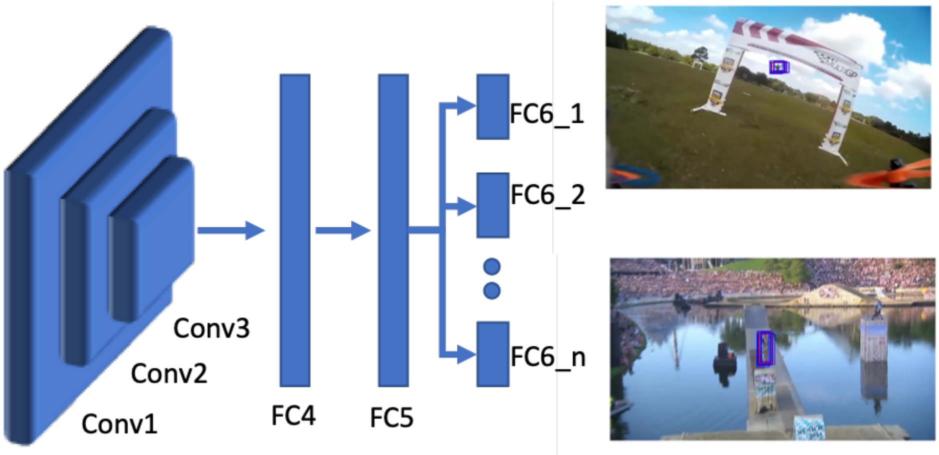


Figure 2. Network architecture of the MDNet tracker

2.1.2. Deep Attentive Tracking (DAT)

DAT learns to attend a specific foreground class from the background by using reciprocation learning. Here, reciprocation learning refers to utilizing the backpropagated gradient to aid the learning procedure. It is achieved by passing the input image through the network and gathering the backpropagated gradients in the input layer which are later used as a regularization term with the classification loss. The MDNet architecture is used for the implementation. However, VOT benchmark training is no longer required for this tracker. Instead, for the first three convolutional blocks, VGG-M weights pretrained on ImageNet [63] are used and never updated. Similarly to MDNet, the fully connected layers are updated during online fine tuning. During this process, multiple patches are sampled at every input frame. The sampled patches are passed through the network and an attention map is obtained by taking the partial derivative of the classification score with respect to the input image at each patch. This attention map regularizes the original loss function through an additional term, with the goal of better classification where the map localizes the target. The objective is to maximize the mean and minimize the standard deviation of the attention map corresponding to the true class and do the inverse for the background. Eventually a hyperparameter is added to combine the two losses. Evaluation on OTB benchmarks showed the effectiveness of this tracking algorithm.

2.1.3. Meta-Tracker (Meta-SDNet)

Meta-SDNet improves over the baseline MDNet tracker using a meta-learning approach for online adaptation taking into consideration the uncertainty during tracking. Meta learning refers to a few shot learning procedure, where an algorithm adapts in a new environment by learning either model parameters, or a metric, or proper optimization techniques. In the meta-learning process, the network parameters are learned such that uncertainties in future frames can be minimized by better modeling the target appearance without overfitting the recent target appearances. Specific video sequences suitable for training were selected from a large scale video detection dataset and used to learn the appropriate parameters for meta-learning. The first three layers of Meta-SDNet are based on the pretrained VGG16 framework and used as feature extractors. The last three fully connected layers were randomly initialized and trained with the Adam optimizer.

2.1.4. Real-Time MDNet (RT-MDNet)

RT-MDNet improves over MDNet tracking framework by incorporating ROI alignment technique from Fast-RCNN [64]. The technique allowed to construct a high resolution feature map. The channel activation was based on large receptive field. An adaptive ROI alignment layer was added after the convolutional layers and before the fully connected layers in the original MDNet architecture. This feature extraction method improves the computational complexity of the overall tracking process. Dilated convolution was used to extract high resolution feature maps, which improved the quality of the representation of the target in feature space. Modified bilinear interpolation was considered for the adaptive ROIAlign layer instead of linear interpolation which changed the tracking performance significantly. Another contribution was combining instance embedding loss with the classification loss to discriminate between similar foreground targets in multiple domains. The network was tested on OTB 2015 and UAV123 datasets [47] and performed comparable to the state-of-the-art trackers in real-time.

2.2. CF-Based Trackers

A popular method for visual object tracking is learning Discriminative Correlation Filters (DCF) to predict the location of the tracked object in a patch [65–79]. A basic correlation filter based tracking framework is shown in Figure 3. Generally, a large patch around the tracked object is cropped at tth frame during tracking. Any feature extraction technique may be used to extract features from the cropped patch. Then the features are utilized to learn a bank of correlation filters that generates a Gaussian response map at the desired target location. Based on the response map, the bounding box of the tracked object is predicted. However, the filter learning procedure is performed at various time instances. Generally, a few frames and the corresponding target locations are saved based on the tracking confidence and utilized during the filter learning procedure.

CF-based trackers aim to find the filters f where the template x is given from a patch with ideal response map y, typically modeled by a Gaussian distribution. For example, the Kernelized Correlation Filter (KCF) [20] utilized a Gaussian kernel to model the target response. The best filter parameters are computed as follows.

$$f^* = \underset{f \in \mathbb{R}^{m \times m}}{\arg\min} \left\| \sum_{c=1}^{D} x^c \star f^c - y \right\|_2^2 \tag{1}$$

However, CFs have limited detection range and may perform poorly when the object undergoes deformation. The patch size and the filter size must be equal, which often makes the tracker learn the background within the patch for irregularly shaped objects. If the patch is small, then in cases of occlusion the object may not be redetected after reappearing. Therefore, it is important to incorporate regularization in the CF-based tracking framework.

Danelljan et al. proposed Spatial Regularization for learning the DCFs (SRDCF [23]). The objective was to weaken the responses due the background information by spatially modifying the filter coefficients. The background is suppressed by assigning higher values of the filter coefficients which are outside of the target bounding box and vice versa. The filter parameters f^* are learned using Equation (2) where α_t describes the impact of each training sample and w is the spatial regularization term.

$$f^* = \mathop{\arg\min}_{f \in \mathbb{R}^{m \times m}} \sum_{t=1}^{T} \alpha_t \left\| \sum_{c=1}^{D} x_t{}^c \star f^c - y_t \right\|_2^2 + \sum_{c=1}^{D} \| w \cdot f^c \|_2^2 \tag{2}$$

An improved version of SRDCF is the Continuous Convolution Operation Tracker (CCOT [29]), where filters are learned for multiple resolutions of target patches in the continuous domain. These filters are then used to produce multiple resolution feature maps. However, CCOT suffers from the large number of filters that are required to be learned to capture the target representation. Another limitation is that the tracker updates at every frame, which causes overfitting to the most recent target appearance.

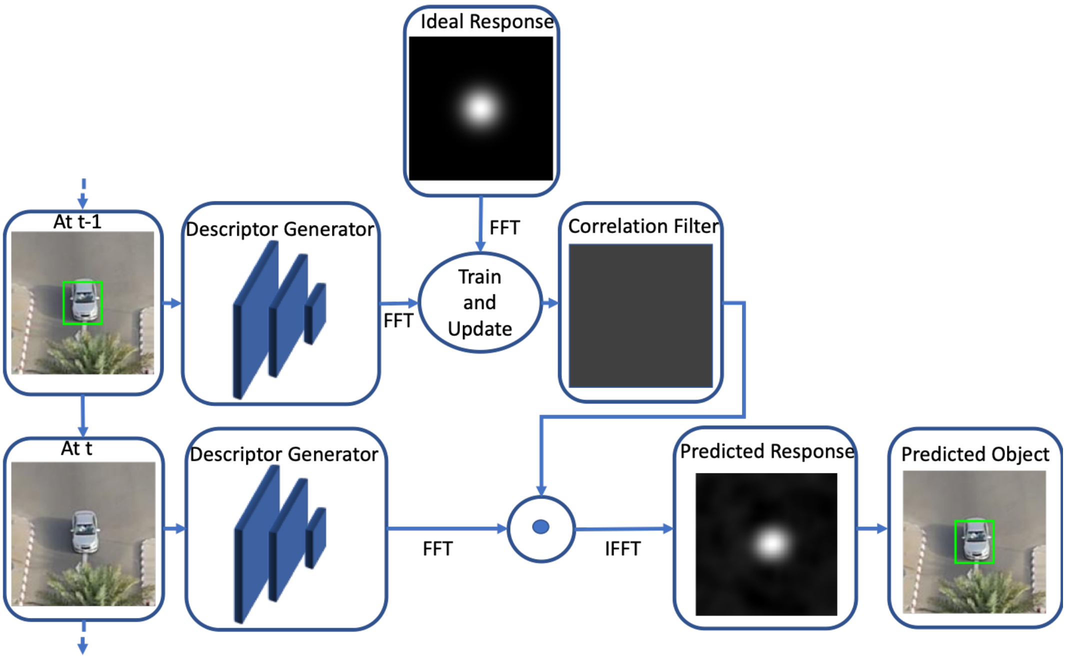


Figure 3. A general deep feature-based correlation filter tracking framework.

There are many variations and improvements to CF trackers since SRDCF and CCOT. We selected the following CF based trackers in our benchmarking. The Efficient Convolution Operators (ECO) [28], the Spatial Temporal Regularized Correlation Filter (STRCF) [80], and the Multi-Fusion Tracker (MFT) [81]. ECO was the winner tracker of VOT2016 challenge. MFT was the winner tracker of VOT2018 challenge. STRCF is also one of the top performing CF-based trackers. A brief overview of these approaches is given below.

2.2.1. Efficient Convolution Operators (ECO)

The ECO formulation is based on discriminative correlation filters with a factorized convolution operator introduced to reduce overfitting, a generative model to estimate the training sample distribution, and an efficient model update strategy. The base framework of CCOT had many filters that contained negligible energy, and these were eliminated in ECO to make the training more efficient. Then, filters are reproduced as the linear combination of the learned filters. The Gauss–Newton method was used to optimize the quadratic loss using conjugate gradient method. The factorized convolution operation reduces the computational and memory complexity of the tracker. To improve overfitting compared with CCOT, a new sample space was introduced based on Gaussian mixtures to obtain a representative sample set. A model update strategy was also introduced to reduce overfitting based on updating the model every N_s frames, where this parameter was identified by heuristics. Small values of N_s may result in overfitting, whereas large values reduce the convergence speed of the optimization. The base network model was VGG, where features from the first and last convolutional layers were used along with HOG and Color features. Finally, the comparison was done on the tracking benchmarks and the model achieved state-of-the-art performance on the VOT2016 challenge.

2.2.2. Multi-Fusion Tracker (MFT)

The Multi-Fusion Tracker (MFT) improves upon the baseline SRDCF tracker by using a motion model and hierarchical feature selection with adaptive fusion. Typically, motion models are ignored in CF and TD trackers. However, MFT utilized a motion model to improve over partial occlusion and bounding box estimation. For the motion model, Kalman filtering was used, which effectively reduces the noise of the bounding box center locations from the DCF predictions. Another important distinction is that the online training was formulated to learn independent correlation filters for multilevel CNN features. It was determined that early layer features are better for adapting the scale changes for small deformation, but deeper features are better for adapting with larger deformation. Additionally, middle-level features are the most representative of the target scale, as the deeper features are prone to drifting towards similar objects. This shortcoming is solved by fusing multilevel features from the network. For these multilevel features, adaptive independent correlation filters were learned using conjugate gradient method. The model was updated at a fixed frame interval instead of every frame following ECO. Then, the outputs of the multiple independent hierarchical filters are fused using an adaptive weighting scheme, where the center location of the target bounding box is extracted from the feature map. Scale change is achieved using a multiscale search strategy of the image patches, after cropping based on the motion estimation model which predicts the center location of the patch. The MFT algorithm outperforms the baseline ECO architecture on the OTB dataset.

2.2.3. Spatial Temporal Regularized Correlation Filter (STRCF)

Spatial Temporal Regularized Correlation Filter (STRCF) tracking incorporates both spatial and temporal regularization on the DCF framework. In a comparative study on different sequences where the target appearance varies significantly, STRCF outperforms SRDCF due to its effective appearance modeling. To solve for the filter parameters, the Alternating Direction Method of Multipliers (ADMM) was used to achieve closed form solutions. The algorithm can operate in real-time when using handcrafted features. With the temporal regularization term, the objective function to update the filters $f^*$is given by

$$f^* = \underset{f \in \mathbb{R}^{\mathrm{m} \times \mathrm{m}}}{\arg\min} \frac{1}{2} \left\| \sum_{c=1}^{D} {x_t}^c \star f^c - y_t \right\|_2^2 + \frac{1}{2} \sum_{c=1}^{D} \| w \cdot f^c \|_2^2 + \frac{\mu}{2} \| f - f_{t-1} \|_2^2 \tag{3}$$

Here, f_{t-1} represents the filters used in the $(t-1)$th frame and μ denotes the hyperparameter for regularization. This expansion to online passive-aggressive algorithm [82] improves over SRDCF in two ways: (i) better model updating with multiple samples and (ii) better occlusion handling by passively updating the correlation filters. Evaluation results on Temple-color, VOT-2016, and OTB 2015 benchmarks showed that this algorithm achieved state-of-the-art accuracy.

2.3. SN-Based Trackers

Siamese networks contain twin branches with shared weights and are widely used in visual object tracking [30,83–93]. The objective of Siamese networks is to learn a shared representation of the input images in a similarity learning fashion. Generally, two input images are fed to the network in each of the branches, and both branches of the network are updated at the same time with shared weights. Bertinetto and Valmadre et al. [30] proposed a fully convolutional Siamese architecture for tracking (SiamFC) that has a template branch and a search window branch. The network was trained with the ImageNet video detection dataset [63], where two different frames in a sequence are cropped and resized such that the area A of the resized patch is

$$A = s(w + 2p) \cdot s(h + 2p) \tag{4}$$

where w and h are the width and height of the corresponding target bounding box, p is the context amount which is set to $p = (w + h)/4$, and s is the scale factor. Finally, cross-correlation is applied in feature space to get the final response map. Multiples scales and aspects are considered to deal with the scale and aspect ratio changes. Training the network is done using positive and negative pairs with logistic loss

$$l(y, v) = log(1 + exp(-yv)) \tag{5}$$

where v is the score for one pair of patches and $y \in \{+1, -1\}$ is the target value. The overall network is trained using SGD with average loss.

One limitation of SiamFC is that it does not always find the tightest bounding box around the target and the resulting localization accuracy is not as good compared to the CF-based trackers. Li et al. [32] incorporated the Faster R-CNN [94] with the SiamFC architecture. The bounding box proposal generation and bounding box regression from Faster R-CNN improved the overall performance of the tracking framework. He et al. [95] proposed a twofold Siamese network architecture where semantic appearance information is encoded to get a better response map. Zhu et al. [31] further improved SiamRPN using distractor-aware training. They also used a local to global search strategy to improve tracking during occlusion.

Among SN-based trackers, we select SiamFC [30] and DaSiamRPN [31]. SiamFC was the winning tracker of the VOT2017 realtime challenge and DaSiamRPN was the winning tracker of the VOT2018 realtime challenge. Furthermore, these trackers are the backbone of many other state-of-the-art SN-based trackers [83–86]. Brief descriptions about these trackers are provided below.

2.3.1. Fully Convolutional Siamese Tracker (SiamFC)

SiamFC casts the tracking problem as a similarity learning problem and uses fully convolutional branches for feature embedding. The SiamFC network architecture is shown in Figure 4. Two input images of different size are fed into the two branches of the network and the same transformation is applied to both images. Keeping the target at the center, the first frame of the sequence is cropped and resized to pass through one of the branches of the Siamese network. The first frame of the sequence is kept fixed in one of the branches in the network. All the other frames of the sequence are cropped, resized, and passed through the second branch of the network. After getting the feature embedding from both branches, a correlation layer is used to find the correlation between two images in the latent space. In the final correlation map, the bounding box is determined in the detection frame based on the similarity score. In the SiamFC architecture, binary cross-entropy loss with SGD is used during training. The network is trained offline on the 2015 version of ImageNet Large Scale Visual Tracking Challenge [63] dataset and no online update is made during tracking. One drawback of this approach is that it uses an expensive multiscale test to adapt for changes in the object's scale, which is not very efficient and does not capture the scale change very well.

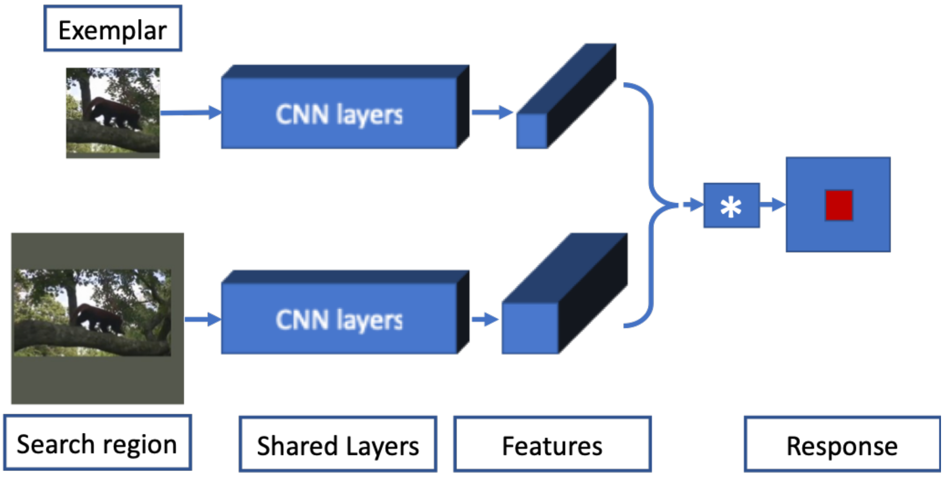


Figure 4. SiamFC network architecture [30].

2.3.2. Distractor-Aware Siamese Region Proposal Networks (DaSiamRPN)

The initial SiamRPN [32] architecture utilizes a key concept in Siamese networks, where the goal is to learn an embedding space maximizes the distance between the inter-class objects and minimizes the distance between intra-class objects. SiamRPN introduces several new concepts including a Region Proposal Network (RPN) and one-shot learning. The region proposal network is utilized on top of the base Siamese network adopted from the original SiamFC architecture for feature extraction. The DaSiamRPN network improves the training procedure by increasing the number of positive examples to reduce the imbalance between positive and negative examples. The original SiamRPN is trained on Youtube-BB dataset [96] consisting of 200,000 video sequences which are annotated every 30 frames. In DaSiamRPN, the ImageNet Detection [63] and COCO Detection [97] datasets were augmented into image pairs, and used to train the network. Semantic negative pairs were included, instead of considering hard negative pairs as in object detection. Additionally, motion blur images were used for training.

Distractors were collected using non maximum suppression to avoid redundant candidates. Among all the predicted bounding boxes, the box with highest score was set as the target and boxes with score greater than a threshold were chosen as distractors. This approach works well for short-term tracking, but it may fail for long-term tracking, as the search region is not large enough to cover the entire range within the image where the object may reappear. DaSiamRPN tried to solve this issue by using local-to-global search strategy when tracking failure occurs and keeping track of the number of frames in which the target is not found. For the training architecture, a modified AlexNet pretrained on ImageNet was used. Evaluation on multiple benchmarks showed that the tracker achieves state-of-the-art performance.

2.4. RL-Based Trackers

In RL-based approaches an agent learns to find an optimal path in an environment from its own experience using feedback. The agent generally receives observations in discrete time steps with rewards and chooses an action from a set of available options. This process continues until convergence. Reinforcement learning algorithms are extensively used in game theory and have been considered for visual object tracking as well [36,37,98–101]. For example, in the HP [99] tracker, Dong et al. introduced a novel hyper parameter selection technique which can learn sequence specific hyperparameters using continuous deep Q-learning. Supancic et al. [100] formulated the tracking problem as an online partially-observable Markov decision making process (POMDP) where, instead of heuristic target initialization, an optimal decision making policy is learned to update the appearance model. Among the RL-based trackers, the Actor-Critic [37] tracker outperformed the other trackers based on the evaluation results provided in the corresponding papers. We have included the Actor–Critic tracker in our benchmarking and provide a short description below.

Actor–Critic Tracking (ACT)

ACT is a reinforcement learning tracker that can operate in real-time. The two main components of the framework are the Actor and the Critic models. The actor network moves the bounding box to the target location and the critic network guides the learning process during offline training. The process is guided by calculating the Q-value using reinforcement learning to train both the Actor and the Critic networks. A modified deep deterministic policy gradient algorithm is used to effectively train the model. During online tracking, the Actor model employs a dynamic search framework to learn the position of the target and the Critic model verifies the position to make the tracker more robust. A pretrained VGG architecture was used to initialize the Actor and Critic networks. In the Critic network, Q learning was done using the Bellman equation in Q-learning, while the Actor network learns using chain rule. During training, the samples were generated from translation and scaling of the bounding box, where the scale was sampled from a Gaussian distribution centered by the object location. The ImageNet Videos were used to train the Actor network so that for each iteration 20 to 40 frames were randomly chosen for training. The tracker achieves 30 fps speed with performance comparable with state-of-the-art trackers on popular tracking benchmarks.

3. Experiments

In our experiments, we evaluated a subset of the OTB 2015 [27] dataset, the DTB70 [48] dataset, the UAV123 [47] dataset and the UAV20L [47] dataset as our benchmark datasets. For the aerial style subset of OTB 2015 dataset, we selected the sequences where the camera is above the ground. The chosen sequences are: Basketball, Bolt, CarScale, Couple, Crossing, Crowds, Human3, Human4-2, Human5, Human6, Jogging-1, RedTeam, Subway, SUV, Walking, Walking2, and Woman.

The DTB70 dataset contains 70 sequences of UAV collected data where the bounding boxes are drawn manually. Some of the sequences are collected from YouTube. Different types of camera motion, including translation and rotation, are incorporated to make the dataset more challenging. Three types of targets appear in the videos: human, animal, and rigid objects.

The UAV123 dataset contains 123 video sequences taken from UAV platforms. Note that we excluded the seven synthetic sequences from the UAV123 dataset for our evaluation. A subset of the UAV123 dataset, UAV20L, is also evaluated for long-term tracking analysis.

The attributes of the aerial datasets are listed in Table 1. These attributes make aerial tracking more challenging and their annotations are available within the corresponding datasets. The comparison of different trackers with these attributes provide better understanding of their performance under different tracking scenarios. Comparisons across UAV123 and DTB70 will indicate the generalizability of different trackers. For long-term tracking, an important consideration is consistent performance in a long temporal span, which tests the tracker's ability to create a robust model and perform efficient model updates. Some trackers may drift a little from one frame to the next, which may not be noticeable in short term sequences, but eventually could result in target loss during long-term tracking.

Regarding the datasets, there are inherent differences between the OTB aerial subset and other aerial datasets in terms of resolution, attributes, and size of the objects to be tracked. In the standard OTB sequences, the objects are much larger and occupy a larger portion of the image frame, while in aerial datasets such as DTB70, UAV123, and UAV20L, the object to track takes a smaller portion of the image because the camera is higher and covers a larger area. Another important distinction of the DTB or UAV sequences is that the camera rotates around the object, whereas none of the videos in OTB sequences have this attribute. Additionally, the OTB benchmark sequences have minimal camera jitter and less clutter in the background.

Table 1. Dataset attributes.

UAV123			DTB70		
Abb	**Full Name**	**Total Sequence**	**Abb**	**Full Name**	**Total Sequence**
ARC	Aspect Ratio Change	65	ARV	Aspect Ratio Variation	25
BC	Background Clutter	21	BC	Background Clutter	13
CM	Camera Motion	64	DEF	Deformation	18
FM	Fast Motion	22	FCM	Fast Camera Motion	41
FOC	Full Occlusion	33	IPR	In-plane-rotation	47
IV	Illumination Variation	25	MB	Motion Blur	27
LR	Low Resolution	48	OPR	Out-of-plane Rotation	06
OV	Out of View	28	OV	Out-of-view	07
POC	Partial Occlusion	68	OCC	Occlusion	17
SOB	Similar Object	39	SOA	Similar Object Around	27
SV	Scale Variation	103	SV	Scale Variation	22
VC	Viewpoint Change	55			

Tracker codes were obtained from their Github repository at the URLs provided in Table 2. None of the algorithms in our implementation were trained from scratch. All of the trackers were implemented in a server workstation with NVIDIA TITAN-V GPU. The CF-based algorithms were implemented using MATLAB and MATConvNet, whereas the other trackers are implemented using Python and PyTorch. Readers are referred to the Github pages in Table 2 for further implementation details.

All the trackers were evaluated using one pass evaluation (OPE) introduced by the OTB benchmark evaluation procedure. The trackers were initialized in the very first frame and never re-initialized after a tracking failure. All the trackers were set to provide the output in the OTB format ($[tclx, tcly, w, h]$), where $tclx$ and $tcly$ are the coordinates of the top left corner of the bounding box, respectively, and w and h are the width and height of the box, respectively.

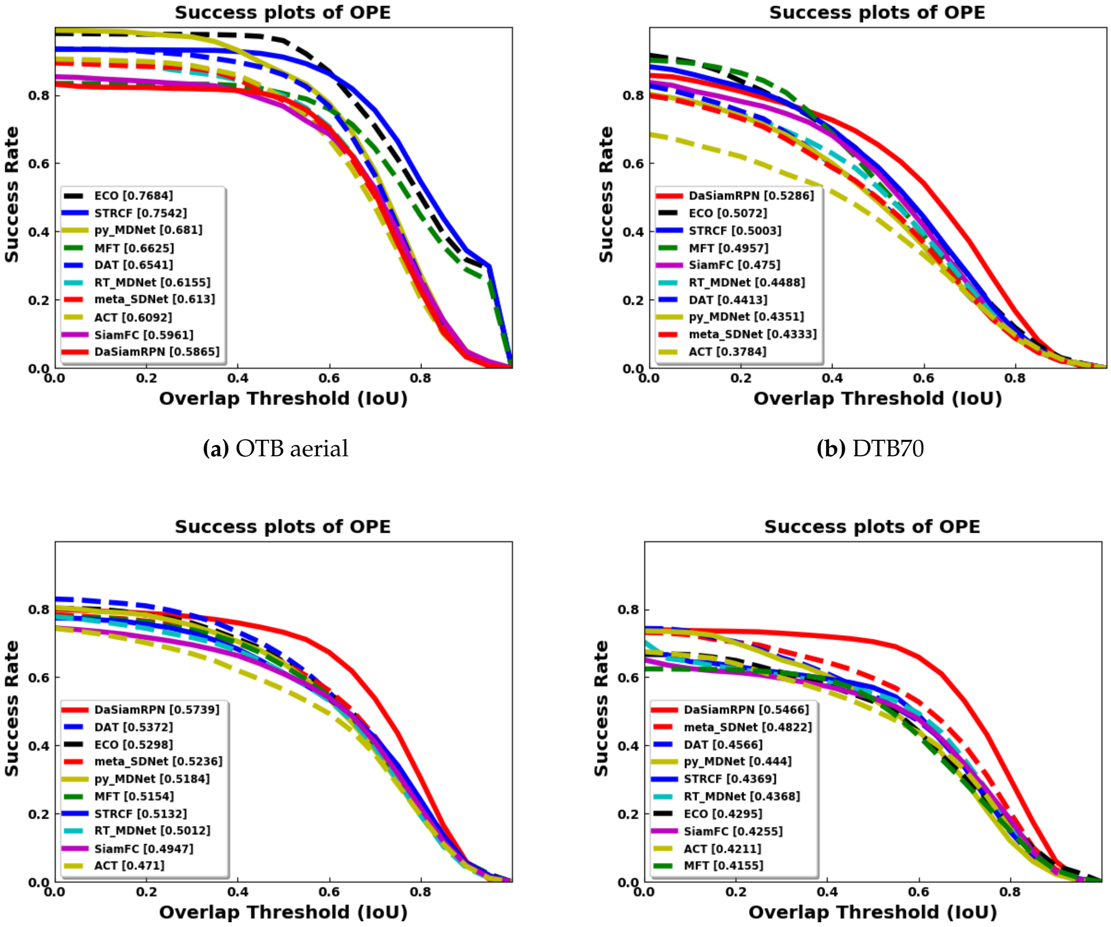
To evaluate tracker performance on the aerial benchmarking datasets, we examined visual examples as well as results based on evaluation metrics. To assess overlap performance, successful tracking is considered if the predicted bounding box and the groundtruth bounding box have an intersection over union (IOU) overlap greater than or equal to some threshold (e.g., 0.5). The tracker is evaluated for different thresholds and the success vs. threshold plot is obtained. The area under the curve (AUC) is computed based on the success vs. threshold plot and the trackers are ranked based on this value.

Table 2. URLs for the codes of the implemented trackers. The code is denoted as P when the implementation is in Python and PyTorch and M when the implementation is in MATLAB and MatConvNet. MDNet is denoted by py-MDNet since it is based on a PyTorch implementation.

Tracker	Base Network	Code	Code Repository
ACT	MDNet	P	https://github.com/bychen515/ACT
DaSiamRPN	SiamFC	P	https://github.com/foolwood/DaSiamRPN
DAT	MDNet	P	https://github.com/shipubupt/NIPS2018
ECO	SRDCF	M	https://github.com/martin-danelljan/ECO
meta-SDNet	MDNet	P	https://github.com/silverbottlep/meta_trackers
MFT	SRDCF	M	https://github.com/ShuaiBai623/MFT
py-MDNet	MDNet	P	https://github.com/HyeonseobNam/py-MDNet
RT-MDNet	MDNet	P	https://github.com/IlchaeJung/RT-MDNet
SiamFC	SiamFC	P	https://github.com/HengLan/SiamFC-PyTorch
STRCF	SRDCF	M	https://github.com/lifeng9472/STRCF

4. Results and Discussion

In this section, we have present our benchmarking results. In Figure 5, the overlap success plot is shown for all benchmark datasets. The AUC, computed for each of the trackers, is indicated in brackets in the legends. The results show that DaSiamRPN outperforms the other trackers in three out of four datasets. In Figure 6, the overlap success plot is shown for various attributes in the UAV123 dataset. The AUC is computed and shown in the legends. This plot shows how well various trackers perform in specific challenges such as occlusion, out of view, fast motion, etc. It can be seen from these results that DaSiamRPN does the best in most of the challenges. In Table 3, tracker performance (in terms of AUC) is compared for the ground based OTB dataset and the aerial datasets. The results show that even though the AUC for all trackers is high for OTB datasets, their performance was significantly reduced in the aerial datasets.



(a) OTB aerial

(b) DTB70

(c) UAV123

(d) UAV20L

Figure 5. Overlap success plots of OPE for the aerial datasets. Results show that DaSiamRPN outperforms all other trackers in DTB70, UAV123, and UAV20L datasets. ECO performed well in the OTB aerial subset. Best viewed zoomed in and in color.

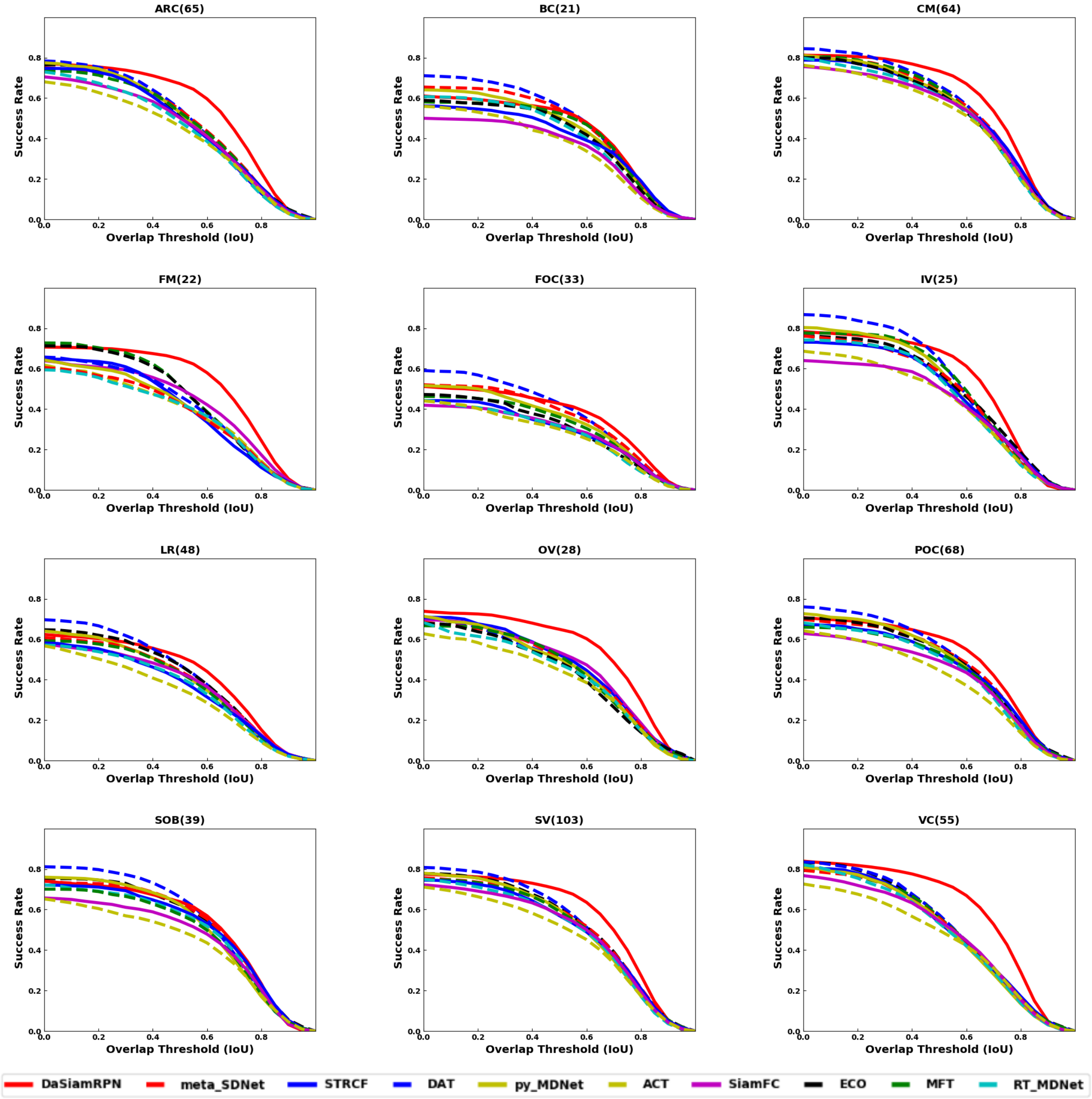
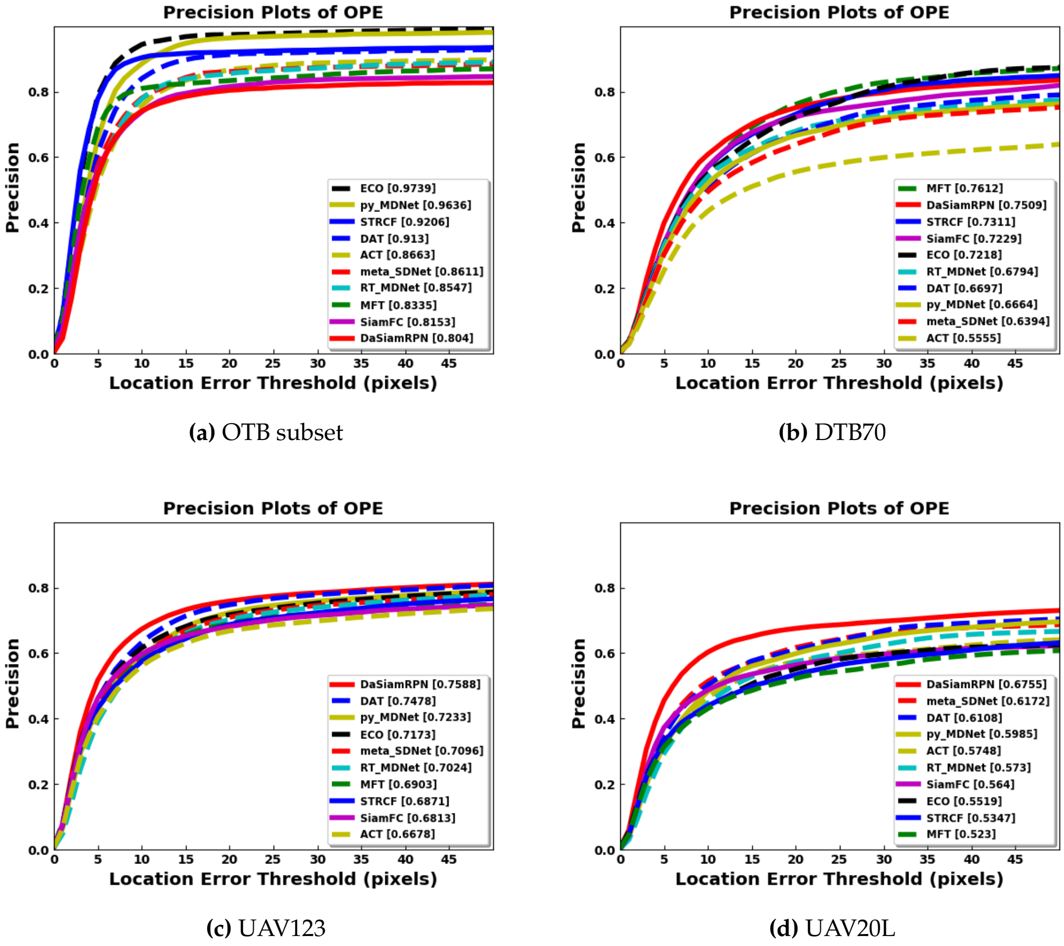


Figure 6. Overlap plots of OPE for UAV123 dataset attributes. DaSiamRPN outperforms other trackers in most of the challenges except BC, FOC, and SOB. Best viewed in color.

In terms of precision, the success rate of each tracker is evaluated based on the center to center distance, between the predicted bounding box and the groundtruth bounding box, compared to some predefined threshold in pixels. The center distance threshold is swept to find the precision vs. threshold plots. The precision values for a threshold of 20 pixels are shown in the brackets of the legends in the precision plots. Figure 7 shows the precision plots for all the benchmark datasets. It is seen that DaSiamRPN outperforms the other trackers in terms of precision as well. In Figure 8, the precision plots for different attributes of UAV123 dataset are depicted. The results show that DaSiamRPN outperforms the other tracker in most of the challenges.



(a) OTB subset

(b) DTB70

(c) UAV123

(d) UAV20L

Figure 7. Precision plots of OPE. DaSiamRPN outperforms other trackers in UAV123 and UAV 20L datasets. ECO shows best performance for the OTB subset and MFT outperform other trackers in the DTB70 datset. Best viewed zoomed in and in color.

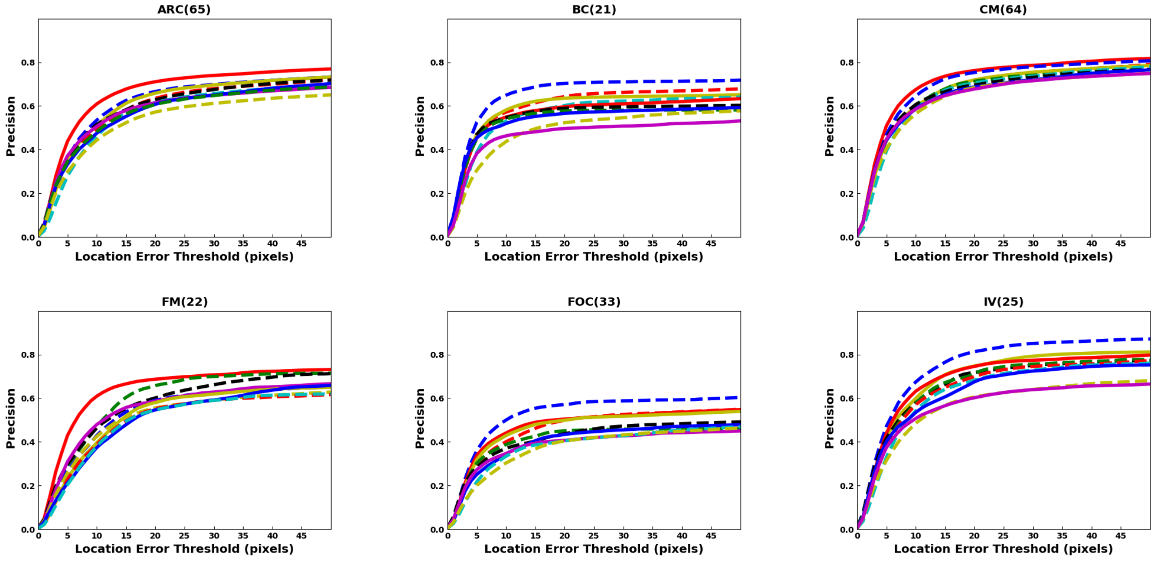


Figure 8. *Cont.*

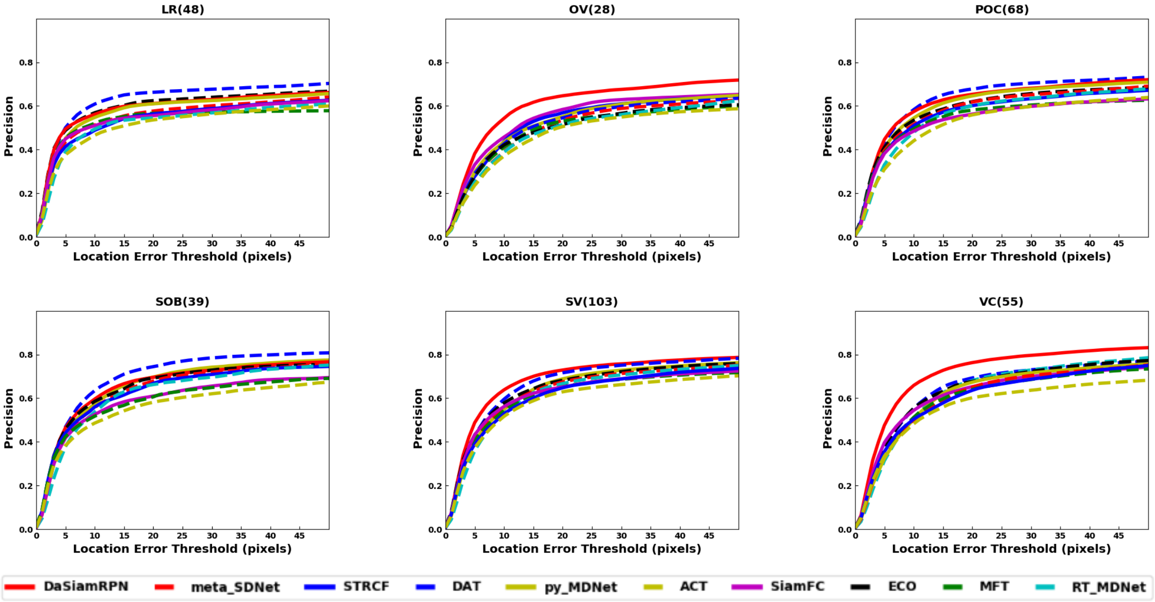


Figure 8. Precision plots of OPE of the UAV123 dataset attributes. DaSiamRPN outperforms the other trackers in most of the challenges where DAT outperforms other trackers in BC, FOC, IV, LR, and SOB challenges. Best viewed in color.

Table 3. Comparison of AUC tracker performance for various datasets. Scores show that trackers perform worse in aerial datasets compared to the ground based datasets. Red indicates top performance and blue indicates runner up. Best viewed in color.

Tracker	OTB 50	OTB 100	OTB Aerial	DTB 70	UAV123	UAV 20L
ACT	0.657	0.625	0.6092	0.3784	0.471	0.4211
DaSiamRPN	N/A	N/A	0.5865	0.5286	0.5739	0.5466
DAT	0.704	0.668	0.6541	0.4413	0.5372	0.4566
ECO	N/A	0.70	0.7684	0.5072	0.5298	0.5466
meta-SDNet	N/A	0.662	0.613	0.433	0.5236	0.4822
MFT	0.726	0.698	0.6625	0.4957	0.5154	0.4155
py-MDNet	0.708	0.678	0.681	0.4351	0.5184	0.444
RT-MDNet	N/A	0.650	0.6155	0.4488	0.5012	0.4368
SiamFC	0.612	N/A	0.5961	0.475	0.4947	0.4255
STRCF	N/A	0.683	0.7542	0.5003	0.5132	0.4369

In Table 4, the AUC of overlap success is shown for various attributes present in the UAV123 dataset. In Table 5, the precision at 20 pixel threshold is shown for the UAV123 dataset attributes. Both DaSiamRPN and DAT trackers perform well in these challenges. In Table 6, the AUC of overlap success is provided for various attributes in the DTB70 dataset. In Table 7, the precision at the 20 pixel threshold is provided for different challenges present in the DTB70 dataset. DaSiamRPN achieved better results in terms of both overlap success and precision in most of the challenges of the DTB70 dataset.

Visual results on the aerial datasets are shown in Figure 9. These illustrate the challenges for trackers due to the small target size, image rotation change in zoom level, and presence of similar looking distractors. The speed comparison is provided in Figure 10 on the UAV123 dataset. The fastest tracker is DaSiamRPN which runs above 200 fps and the slowest tracker is DAT which runs below 1 fps. The ECO and DAT trackers achieve good tracking performance at lower speeds.

Table 4. Overlap results for various UAV123 dataset attributes listed in Table 1. Top performing tracker is shown in red and the second best performer is shown in blue. Best viewed in color.

Tracker	ARC(65)	BC(21)	CM(64)	FM(22)	FOC(33)	IV(25)	LR(48)	OV(28)	POC(68)	SOB(39)	SV(103)	VC(55)
ACT	0.3993	0.3382	0.4865	0.3677	0.259	0.4151	0.3126	0.3829	0.3831	0.4116	0.4404	0.4333
DaSiamRPN	0.527	0.4168	0.5794	0.5024	0.349	0.5298	0.4076	0.5235	0.4851	0.5032	0.5477	0.5836
DAT	0.4627	0.4577	0.544	0.3972	0.3604	0.5229	0.4068	0.4297	0.4741	0.5277	0.5109	0.4929
ECO	0.4594	0.3898	0.5167	0.4321	0.2873	0.484	0.3936	0.4101	0.4565	0.5098	0.5044	0.4953
meta-SDNet	0.4596	0.4346	0.5334	0.3667	0.3379	0.4696	0.3737	0.429	0.4566	0.5043	0.495	0.4784
MFT	0.4539	0.4047	0.5259	0.4372	0.2983	0.4988	0.3646	0.4257	0.4329	0.4644	0.4885	0.4827
py-MDNet	0.4569	0.4139	0.5206	0.3805	0.3206	0.4855	0.376	0.4271	0.4548	0.4928	0.4906	0.4723
RT-MDNet	0.4198	0.3876	0.5042	0.3607	0.2655	0.4558	0.3407	0.4058	0.4243	0.4739	0.4716	0.4685
SiamFC	0.427	0.3336	0.4977	0.4113	0.2713	0.4143	0.3566	0.4457	0.4048	0.439	0.4694	0.4627
STRCF	0.4508	0.3761	0.5231	0.3863	0.2774	0.4632	0.3456	0.446	0.4408	0.4895	0.4819	0.4828

Table 5. Precision results for various UAV123 dataset attributes listed in Table 1. Top performing tracker is shown in red and the second best performer is shown in blue. Best viewed in color.

Tracker	ARC(65)	BC(21)	CM(64)	FM(22)	FOC(33)	IV(25)	LR(48)	OV(28)	POC(68)	SOB(39)	SV(103)	VC(55)
ACT	0.572	0.5233	0.6841	0.5557	0.4046	0.6058	0.5361	0.5056	0.5591	0.5814	0.6289	0.602
DaSiamRPN	0.7108	0.5962	0.7619	0.687	0.5045	0.7454	0.6108	0.6465	0.6534	0.6955	0.7289	0.7627
DAT	0.6661	0.7028	0.7531	0.5904	0.5706	0.8129	0.6625	0.5455	0.6776	0.7444	0.716	0.6928
ECO	0.6287	0.5891	0.6957	0.6012	0.4379	0.7179	0.6242	0.5151	0.6197	0.6921	0.6817	0.6768
meta-SDNet	0.6347	0.6427	0.7187	0.5522	0.499	0.6984	0.5767	0.5352	0.618	0.6783	0.6741	0.6507
MFT	0.6069	0.5767	0.713	0.6577	0.4506	0.7111	0.5664	0.5308	0.5811	0.6061	0.6521	0.6403
py-MDNet	0.6583	0.6353	0.7186	0.58	0.499	0.7476	0.6096	0.5516	0.6457	0.697	0.6884	0.6731
RT-MDNet	0.6158	0.6019	0.7052	0.5459	0.4047	0.6826	0.5451	0.5149	0.6007	0.6623	0.6669	0.6812
SiamFC	0.6143	0.4967	0.6774	0.5861	0.4073	0.602	0.5555	0.5841	0.5627	0.6098	0.6515	0.6527
STRCF	0.6075	0.5657	0.6871	0.5466	0.4341	0.6747	0.5636	0.5686	0.6014	0.6683	0.6476	0.6361

Table 6. Overlap results for various DTB70 dataset attributes listed in Table 1. Top performing tracker is shown in red and the second best performer is shown in blue. Best viewed in color.

Tracker	SV(22)	ARV(25)	OCC(17)	DEF(18)	FCM(41)	IPR(47)	OPR(6)	OV(7)	BC(13)	SOA(27)	MB(27)
ACT	0.3439	0.3507	0.3995	0.3210	0.3534	0.3261	0.2351	0.3661	0.2715	0.3420	0.3089
DaSiamRPN	0.5905	0.5339	0.4203	0.5502	0.5240	0.5013	0.4337	0.5008	0.4402	0.4514	0.4668
DAT	0.4483	0.4280	0.3910	0.4489	0.4361	0.4384	0.3589	0.4665	0.3308	0.3835	0.3652
ECO	0.4753	0.4344	0.5121	0.4477	0.5202	0.4584	0.3257	0.4318	0.4434	0.5255	0.5090
meta-SDNet	0.4508	0.3993	0.4045	0.4630	0.4090	0.3996	0.3162	0.3949	0.2989	0.4029	0.3322
MFT	0.5362	0.4475	0.4324	0.5004	0.5080	0.4897	0.4239	0.4019	0.4614	0.4849	0.5073
py-MDNet	0.4286	0.3900	0.4305	0.4334	0.4314	0.4176	0.3500	0.4643	0.3527	0.3815	0.3655
RT-MDNet	0.4254	0.3777	0.4878	0.3243	0.4948	0.4011	0.3175	0.3999	0.3083	0.4694	0.4035
SiamFC	0.4764	0.4280	0.4257	0.4311	0.4974	0.4615	0.3701	0.4159	0.4663	0.4600	0.4736
STRCF	0.5156	0.4388	0.4714	0.4979	0.5181	0.4718	0.3550	0.4356	0.4107	0.4827	0.4797

Table 7. Precision results on various DTB70 dataset attributes listed in Table 1. Top performing tracker is shown in red and the second-best performing tracker is shown in blue. Best viewed in color.

Tracker	SV(22)	ARV(25)	OCC(17)	DEF(18)	FCM(41)	IPR(47)	OPR(6)	OV(7)	BC(13)	SOA(27)	MB(27)
ACT	0.4434	0.4658	0.6437	0.4245	0.5484	0.4707	0.2138	0.5688	0.4229	0.5321	0.4892
DaSiamRPN	0.7749	0.7363	0.6233	0.7774	0.7465	0.7086	0.5252	0.6950	0.6527	0.6638	0.6811
DAT	0.6783	0.6451	0.5823	0.6978	0.6586	0.6501	0.4935	0.7192	0.5316	0.6043	0.5694
ECO	0.5982	0.6007	0.7449	0.6052	0.7523	0.6376	0.3210	0.5851	0.7196	0.8023	0.7362
meta-SDNet	0.6224	0.5751	0.6272	0.6545	0.5963	0.5771	0.3883	0.5802	0.4336	0.6371	0.4851
MFT	0.7517	0.6463	0.7074	0.7155	0.7922	0.7551	0.5144	0.6315	0.8332	0.7807	0.8237
py-MDNet	0.6299	0.5577	0.6778	0.6628	0.6624	0.6284	0.4605	0.7110	0.5665	0.6093	0.5816
RT-MDNet	0.6017	0.5512	0.7539	0.4828	0.7558	0.5924	0.4378	0.6247	0.5065	0.7135	0.6315
SiamFC	0.6800	0.6407	0.6780	0.6392	0.7626	0.7004	0.5309	0.6713	0.7552	0.7184	0.7348
STRCF	0.7079	0.6371	0.7048	0.7440	0.7700	0.6904	0.4970	0.6572	0.6587	0.7318	0.7230

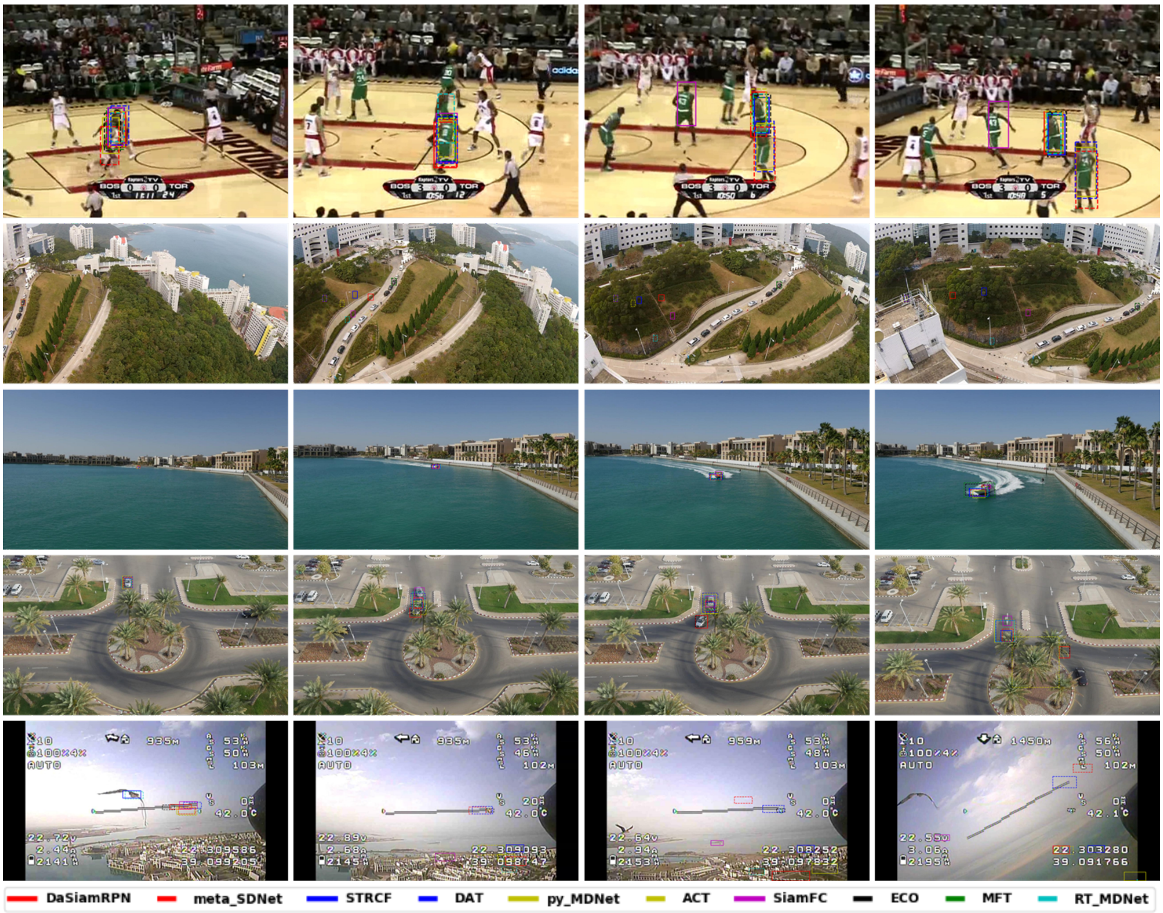


Figure 9. Tracking visualization on different sequences (top to bottom): Basketball (OTB), Car2 (DTB70), Boat8 (UAV123), Car7 (UAV123), and bird1 (UAV20L). Best viewed in color after zooming in.

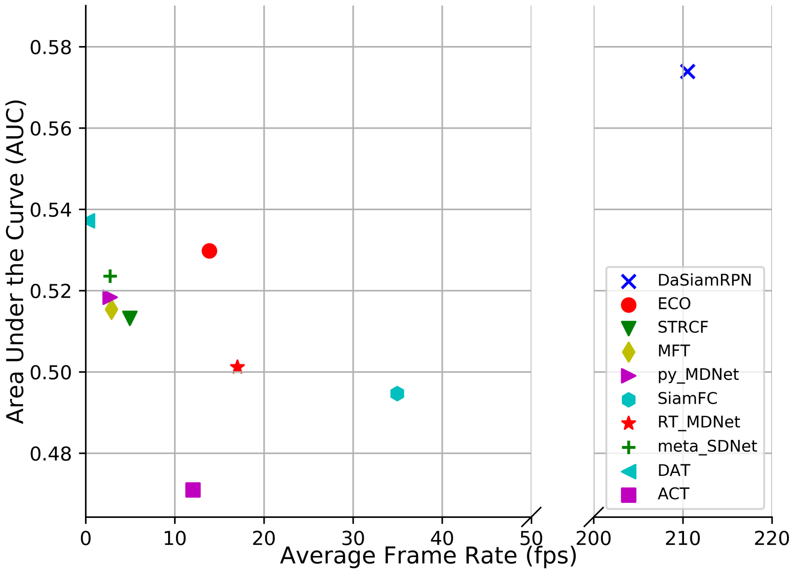


Figure 10. Results showing AUC vs. Frame Rate on UAV123 dataset. Top right corner indicates fastest implementation and best performance. Best viewed in color.

4.1. Overall Comparison

The overall quantitative comparison of the trackers is shown in Table 3 for all benchmark datasets. The results for the OTB aerial subset are slightly worse than the overall OTB results. As the OTB subset does not exhibit all the attributes of the other aerial datasets, tracker performance remains similar with respect to the overall ground level OTB dataset. The evaluation plots in Figures 5 and 7 showed that the ECO tracker performs the best and STRCF is the second best for the OTB subset.

Tracker performance degrades consistently for the aerial datasets compared to ground level tracking. For the DTB70 dataset, the DaSiamRPN and MFT trackers perform the best for overlap and precision respectively. The DaSiamRPN tracker performs the best for the UAV123 dataset and UAV20L datasets. The distractor aware training, region proposal network for better IoU, and local to global search for redetection make the DaSiamRPN the best overall performer for aerial tracking.

4.2. Attribute Comparison

Attributes are annotated in the datasets, as shown in Table 1, and are used to evaluate the tracker performance under various challenging conditions. In different sequences, one or several attributes may be present. The evaluation of trackers based on specific video attributes is shown in Figures 6 and 8 and Tables 4 and 5 for UAV123 and in Tables 6 and 7 for DTB70. These results provide insights in the performance of the trackers under various conditions.

The aspect ratio change (ARC-UAV123) and the aspect ratio variation (ARV-DTB70) attributes specify the target's aspect ratio change over the temporal span of the sequence. The evaluation shows that the DaSiamRPN outperforms all the other trackers on this challenge. The region proposal network helps DaSiamRPN achieve better performance in terms of aspect ratio change. Note that in this specific challenge, in the UAV123 dataset, CF-based trackers and MDNet-based trackers perform similarly, whereas CF-based trackers perform relatively well compared to the MDNet-based trackers in the DTB70 dataset. We conclude that the model update strategy of the CF-based trackers help to achieve relatively better performance compared to he MDNet-based trackers.

The Background Clutter (BC) attribute is present in both datasets. This attribute specifies instances where the target is hardly distinguishable from the background. Based on the evaluation, DAT outperforms the other trackers for UAV123 and SiamFC does best for DTB70, for both overlap and precision. Note that SiamFC performs the worst for UAV123 sequences where BC is present.

Camera motion (CM-UAV123), fast motion (FM-UAV123), and fast camera motion (FCM-DTB70) attributes are evaluated. These attributes specify faster relative speed between the camera and the target. The DaSiamRPN tracker outperforms the other trackers for both datasets. However, MDNet based trackers performed comparatively well in the UAV123 dataset when camera motion is involved. However, when the relative motion is fast, the CF-based trackers outperform the MDNet based trackers in both datasets. We attribute this to the better model update strategy of the correlation filter based trackers.

Full occlusion (FOC-UAV123), partial occlusion (POC-UAV123), and Occlusion (OCC-DTB70) attributes are also evaluated. In POC, the target becomes partially occluded and then reappears without full occlusion, whereas in FOC, the target may be fully occluded for several frames and then reappears. The OCC attribute in DTB70 addresses both full and partial occlusion. It is important for the tracker to have redetection capabilities and stop updating the appearance model during full or partial occlusion to prevent drift from the target. Based on the tracker evaluation on these datasets, we find that for full occlusion DAT performed well, whereas for partial occlusion, DaSiamRPN performed well. For OCC in the DTB70 dataset, ECO performed best.

The Out of View (OV) attribute is present in both datasets. This attribute specifies that the target is no longer within the field of view of the camera, which is particularly challenging for most trackers. To do well in this challenge, the tracker must have redetection capabilities and be discriminative enough to distinguish the target from other similar looking objects. DaSiamRPN outperforms all the other trackers in this challenge, because it incorporates local-to-global search strategy.

The attributes Similar Object (SOB-UAV123) and Similar Object Around (SOA-DTB70) indicate that objects with shape and appearance similar to the target appear near the target. These objects are also called distractors. Tracking is more challenging when the target is partially occluded and distractors are present close to the target. The object may also be fully occluded by the distractor. For this challenge, the results show that CF-based trackers perform well in the DTB70 dataset, whereas MDNet-based trackers perform well in the UAV123 dataset. The performance of the Siamese trackers is comparatively lower because the correlation map generally provides high scores on the distractors, which sometimes causes tracker failure.

The Scale Variation (SV) attribute is present in both datasets. It specifies those sequences where the scale of the target changes over time. The results show that DaSiamRPN is significantly better than other trackers. Again, we attribute this to the Region Proposal Network architecture of the DaSiamRPN tracker. Illumination variation (IV), low resolution (LR), and viewpoint change (VC) are some other attributes that are present in the UAV123 dataset. IV specifies the sequences where illumination change is involved related to the target and LR specifies those sequences where the target has low resolution. VC specifies the changes in the camera viewpoint over the temporal span of a sequence. In terms of overlap, DaSiamRPN outperforms the other trackers in these challenges. However, for the precision, MDNet outperforms the other trackers for LR and IV attributes.

Deformation (DEF), In-plane Rotation (IPR), Out of Plane Rotation (OPR), and Motion Blur (MB) are some other attributes present in the DTB70 dataset. Deformation specifies the shape change of the object, in-plane and out-of-plane rotation specifies whether the target object is rotating inside or outside from the image plane, and motion blur specifies blurred target during tracking. For deformation, out-of-plane rotation, and in-plane-rotation, DaSiamRPN performs best. For MB, ECO performs best among the compared trackers for overlap success. However, for the precision success, MFT performs best. The results show that trackers perform much better in the IPR challenges compared to the OPR challenges. In presence of the MB attribute, CF-based trackers generally performs better compared to the other trackers.

4.3. Visual Comparison

We present visual examples of the results for all trackers in Figure 9. The Basketball sequence is taken from the OTB dataset. Although it is not an aerial image, it contains distractors that are large enough to observe. Partial occlusion and distractor attributes are present in this sequence. SiamFC and some MDNet based trackers lost the target by the end of the sequence.

The Car2 sequence from DTB70 dataset specifies the camera motion where the camera rotates around the target. The results show that eight out of 10 trackers lost track as soon as there is significant rotation of the camera in this sequence.

The Boat8 and Car7 sequences are taken from the UAV123 dataset. Boat8 has significant scale and aspect ratio change where the appearance of the target also changed. These results show that ECO and MFT trackers perform well but eventually drift. Surprisingly, DaSiamRPN got stuck on small part of the object, most likely due to a proposal generated for that region. In Car7, the target was occluded by a tree while another distractor appeared at the same time. Only the ECO and the DaSiamRPN trackers were able to successfully handle the situation, whereas all other trackers started to track the distractor. In cases of full occlusion, all the trackers lost the target.

Finally, the bird1 sequence from UAV123 dataset is evaluated. Here the target is moving fast and went out-of-view multiple times for several frames. It is seen that due to the small target size, fast motion and partial occlusion, only STRCF and RT-MDNet successfully track the object until it goes out-of-view, but no tracker can redetect the target when it reappears.

4.4. Speed Comparison

The speed comparison among all the trackers is shown in Figure 10 where the AUC vs. frame rate is plotted for the UAV123 dataset. SN-based trackers have higher frame rate compared to the other trackers. This is because the network parameters are not updated during online tracking. The DaSiamRPN achieves significantly higher frame rate because of its approach to online tracking as one shot learning. Among the CF-based trackers, ECO has the highest frame rate and outperforms the other CF-based trackers. The factorized convolution operation makes the tracker more efficient, thereby allowing it to achieve a higher frame rate and better performance. Among the MDNet based trackers, RT-MDNet achieves real-time performance with a small accuracy drop from the the original MDNet tracker. It is also seen that DAT performs well, but it has the lowest frame rate due to the update strategy of the tracker where the tracker updates on all the frames with score lower than a certain threshold.

5. Conclusions

In this study, we benchmarked ten state-of-the-art CNN-based visual object trackers from four different classes: Siamese Network-based, Tracking by Detection-based, Correlation Filter-based, and Reinforcement Learning. We considered four datasets: a subset of OTB, DTB70, UAV123, and UAV20L datasets for testing and comparing the tracking algorithms. Visual examples of different trackers are shown and the results of an One Pass Evaluation (OPE) are reported. We compared the results among different datasets as well as specific attribute challenges within the datasets. Trackers performed worse in the aerial datasets than in the typical ground level videos.

In our study, we found that Siamese network based trackers face difficulty when there are distractors present within the sequence. This is because the cross-correlation operation will create strong peaks on the distractors and drift may occur particularly when the main target is occluded. Siamese trackers do not have any online model update, which makes them fast but occasionally cannot handle significant appearance change of the object. However, DaSiamRPN performs well due to distractor aware training and accurate localization based on the RPN. The challenge for the CF-based trackers is to find a proper update strategy such that the tracker does not update the appearance model when the target is absent. Among MDNet based trackers, py-MDNet and DAT are computationally expensive where RT-MDNet and meta-tracker run at higher frame rate with relatively lower accuracy. Finally, the RL-based tracker is yet to achive the desired accuracy.

It is notable that none of the trackers is designed for reidentification, which is needed for target reacquisition when the target goes out of view and reappears.

The overall performance of the implemented trackers indicates that further research is needed to reach the full potential of deep learning tracking on aerial sequences.

Author Contributions: A.M.N.T. contributed in performing the experiments and writing the report. B.M. contributed in conceptualization. A.S. contributed in conceptualization, review and editing, and supervising the project. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Air Force Research Laboratory, Sensors Directorate (AFRL/RYAP) under contract number FA8650-18-C-1739 to Systems and Technology Research.

Conflicts of Interest: The authors declare no conflicts of interest. The funders helped reviewing the manuscript and provided the decision to publish the result.

References

1. Papanikolopoulos, N.P.; Khosla, P.K.; Kanade, T. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Trans. Robot. Autom.* **1993**, *9*, 14–35. [CrossRef]
2. Lee, K.; Hwang, J. On-Road Pedestrian Tracking Across Multiple Driving Recorders. *IEEE Trans. Multimed.* **2015**, *17*, 1429–1438. [CrossRef]
3. Laurense, V.A.; Goh, J.Y.; Gerdes, J.C. Path-tracking for autonomous vehicles at the limit of friction. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5586–5591.
4. Tang, S.; Andriluka, M.; Andres, B.; Schiele, B. Multiple people tracking by lifted multicut and person reidentification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3539–3548.

5. Girdhar, R.; Gkioxari, G.; Torresani, L.; Paluri, M.; Tran, D. Detect-and-track: Efficient pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 350–359.
6. Walker, S.; Sewell, C.; Park, J.; Ravindran, P.; Koolwal, A.; Camarillo, D.; Barbagli, F. Systems and Methods for Localizing, Tracking and/or Controlling Medical Instruments. U.S. Patent App. 15/466,565, 25 April 2017.
7. Speidel, S.; Kuhn, E.; Bodenstedt, S.; Röhl, S.; Kenngott, H.; Müller-Stich, B.; Dillmann, R. Visual tracking of da vinci instruments for laparoscopic surgery. In Proceedings of the SPIE Medical Imaging 2014: Image-Guided Procedures, Robotic Interventions, and Modeling, San Diego, CA, USA, 12 March 2014; Volume 9036.
8. Aggarwal, J.K.; Xia, L. Human activity recognition from 3d data: A review. *Pattern Recognit. Lett.* **2014**, *48*, 70–80. [CrossRef]
9. Yilmaz, A.; Javed, O.; Shah, M. Object tracking: A survey. *ACM Comput. Surv. (CSUR)* **2006**, *38*, 13. [CrossRef]
10. Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1442–1468.
11. Ning, J.; Zhang, L.; Zhang, D.; Wu, C. Robust object tracking using joint color-texture histogram. *Int. J. Pattern Recognit. Artif. Intell.* **2009**, *23*, 1245–1263. [CrossRef]
12. Zhou, H.; Yuan, Y.; Shi, C. Object tracking using SIFT features and mean shift. *Comput. Vis. Image Underst.* **2009**, *113*, 345–352. [CrossRef]
13. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
14. Han, Z.; Jiao, J.; Zhang, B.; Ye, Q.; Liu, J. Visual object tracking via sample-based Adaptive Sparse Representation (AdaSR). *Pattern Recognit.* **2011**, *44*, 2170–2183. [CrossRef]
15. Das, S.; Kale, A.; Vaswani, N. Particle filter with a mode tracker for visual tracking across illumination changes. *IEEE Trans. Image Process.* **2011**, *21*, 2340–2346. [CrossRef] [PubMed]
16. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
17. Jia, X.; Lu, H.; Yang, M.H. Visual tracking via adaptive structural local sparse appearance model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1822–1829.
18. Zhong, W.; Lu, H.; Yang, M.H. Robust object tracking via sparsity-based collaborative model. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1838–1845.
19. Yoon, J.H.; Kim, D.Y.; Yoon, K.J. Visual tracking via adaptive tracker selection with multiple features. In *European Conference on Computer Vision*; Springer: London, UK, 2012; pp. 28–41.
20. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef]
21. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2096–2109. [CrossRef] [PubMed]
22. Kiani Galoogahi, H.; Fagg, A.; Lucey, S. Learning background-aware correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1135–1143.
23. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
24. Dong, W.; Chang, F.; Zhao, Z. Visual tracking with multifeature joint sparse representation. *J. Electron. Imaging* **2015**, *24*, 013006. [CrossRef]
25. Ontiveros-Gallardo, S.E.; Kober, V. Objects tracking with adaptive correlation filters and kalman filtering. In Proceedings of the SPIE Optics and Photonics for Information Processing IX, San Diego, CA, USA, 9 September 2015; Volume 9598.

26. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Zajc, L.Č.; Vojír, T.; Bhat, G.; Lukežič, A.; Eldesokey, A.; et al. The Sixth Visual Object Tracking VOT2018 Challenge Results. In *European Conference in Computer Vision ECCV 2018 Workshops*; Leal-Taixé, L., Roth, S., Eds.; Springer: New York, NY, USA, 2019; pp. 3–53.
27. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
28. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; Volume 1, p. 3.
29. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision*; Springer: New York, NY, USA, 2016; pp. 472–488.
30. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional Siamese networks for object tracking. In *European Conference on Computer Vision*; Springer: New York, NY, USA, 2016; pp. 850–865.
31. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese Networks for Visual Object Tracking. *arXiv* **2018**, arXiv:1808.06048.
32. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking With Siamese Region Proposal Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
33. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
34. Jung, I.; Son, J.; Baek, M.; Han, B. Real-time mdnet. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 83–98.
35. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep Attentive Tracking via Reciprocative Learning. In *Neural Information Processing Systems*; NIPS: Barcelona, Spain, 2018.
36. Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Young Choi, J. Action-decision networks for visual tracking with deep reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2711–2720.
37. Chen, B.; Wang, D.; Li, P.; Wang, S.; Lu, H. Real-time'Actor-Critic'Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 318–334.
38. Wu, Y.; Lim, J.; Yang, M.H. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013.
39. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Hager, G.; Lukezic, A.; Eldesokey, A.; et al. The visual object tracking VOT2017 challenge results. In Proceedings of the ICCV2017 Workshops, Workshop on Visual Object Tracking Challenge, Venice, Italy, 2–29 October 2017; pp. 1949–1972.
40. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Čehovin, L.; Vojír, T.; Häger, G.; Lukežič, A.; Fernández, G.; et al. The Visual Object Tracking VOT2016 Challenge Results. In *European Conference in Computer Vision—ECCV 2016 Workshops*; Hua, G., Jégou, H., Eds.; Springer: New York, NY, USA, 2016; pp. 777–823.
41. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; Pflugfelder, R. The visual object tracking VOT2015 challenge results. In Proceedings of the ICCV2015 Workshops, Workshop on Visual Object Tracking Challenge, Santiago, Chile, 11–18 December 2015; pp. 1–23.
42. Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Cehovin, L.; Nebehay, G.; Gustavo, F.; Vojir, T.; Dimitriev, A.; Petrosino, A.; et al. The visual object tracking VOT2014 challenge results. In Proceedings of the ECCV2014 Workshops, Workshop on Visual Object Tracking Challenge, Zurich, Switzerland, 6–12 September 2014; pp. 191–217.
43. Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Porikli, F.; Cehovin, L.; Nebehay, G.; Vojir, T. The visual object tracking VOT2013 challenge results. In Proceedings of the ICCV2013 Workshops, Workshop on Visual Object Tracking Challenge, Sydney, Australia, 1–8 December 2013; pp. 98–111.
44. Song, S.; Xiao, J. Tracking revisited using RGBD camera: Unified benchmark and baselines. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 233–240.

45. Li, A.; Lin, M.; Wu, Y.; Yang, M.H.; Yan, S. NUS-PRO: A new visual tracking challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 335–349. [CrossRef]
46. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef]
47. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In *European Conference on Computer Vision*; Springer: New York, NY, USA, 2016; pp. 445–461.
48. Li, S.; Yeung, D.Y. *Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models*; AAAI: Menlo Park, CA, USA, 2017; pp. 4140–4146.
49. Kiani Galoogahi, H.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. Need for speed: A benchmark for higher frame rate object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1125–1134.
50. Zajc, L.C.; Lukežic, A.; Leonardis, A.; Kristan, M. Beyond standard benchmarks: Parameterizing performance evaluation in visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3343–3351.
51. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386.
52. Muller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 300–317.
53. Valmadre, J.; Bertinetto, L.; Henriques, J.F.; Tao, R.; Vedaldi, A.; Smeulders, A.W.; Torr, P.H.; Gavves, E. Long-term tracking in the wild: A benchmark. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 670–685.
54. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. *arXiv* **2018**, arXiv:1809.07845.
55. Minnehan, B.; Salmin, A.; Salva, K.; Savakis, A. Benchmarking deep learning trackers on aerial videos. In Proceedings of the SPIE Pattern Recognition and Tracking XXIX, Orlando, FL, USA, 30 April 2018; Volume 10649, p. 1064915.
56. Li, P.; Wang, D.; Wang, L.; Lu, H. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338. [CrossRef]
57. Fiaz, M.; Mahmood, A.; Javed, S.; Jung, S.K. Handcrafted and Deep Trackers: A Review of Recent Object Tracking Approaches. *arXiv* **2018**, arXiv:1812.07368.
58. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.; Yang, M.H. Vital: Visual tracking via adversarial learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8990–8999.
59. Park, E.; Berg, A.C. Meta-Tracker: Fast and Robust Online Adaptation for Visual Object Trackers. *arXiv* **2018**, arXiv:1801.03049.
60. Fan, H.; Ling, H. Sanet: Structure-aware network for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 42–49.
61. Teng, Z.; Xing, J.; Wang, Q.; Lang, C.; Feng, S.; Jin, Y. Robust object tracking based on temporal and spatial deep networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1144–1153.
62. Nam, H.; Baek, M.; Han, B. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv* **2016**, arXiv:1608.07242.
63. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
64. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
65. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.

66. Kart, U.; Lukezic, A.; Kristan, M.; Kamarainen, J.K.; Matas, J. Object Tracking by Reconstruction with View-Specific Discriminative Correlation Filters. *arXiv* **2018**, arXiv:1811.10863.
67. Gundogdu, E.; Alatan, A.A. Good features to correlate for visual tracking. *IEEE Trans. Image Process.* **2018**, *27*, 2526–2540. [CrossRef] [PubMed]
68. Bhat, G.; Johnander, J.; Danelljan, M.; Shahbaz Khan, F.; Felsberg, M. Unveiling the power of deep tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 483–498.
69. Zhang, M.; Wang, Q.; Xing, J.; Gao, J.; Peng, P.; Hu, W.; Maybank, S. Visual tracking via spatially aligned correlation filters network. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 469–485.
70. Zhu, Z.; Wu, W.; Zou, W.; Yan, J. End-to-end flow correlation tracking with spatial-temporal attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 548–557.
71. Yao, Y.; Wu, X.; Zhang, L.; Shan, S.; Zuo, W. Joint representation and truncated inference learning for correlation filter based tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 552–567.
72. Wang, N.; Zhou, W.; Tian, Q.; Hong, R.; Wang, M.; Li, H. Multi-cue correlation filters for robust visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4844–4853.
73. Tang, M.; Yu, B.; Zhang, F.; Wang, J. High-speed tracking with multi-kernel correlation filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4874–4883.
74. He, Z.; Fan, Y.; Zhuang, J.; Dong, Y.; Bai, H. Correlation filters with weighted convolution responses. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1992–2000.
75. Li, F.; Yao, Y.; Li, P.; Zhang, D.; Zuo, W.; Yang, M.H. Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2001–2009.
76. Mueller, M.; Smith, N.; Ghanem, B. Context-aware correlation filter tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1396–1404.
77. Zhang, T.; Xu, C.; Yang, M.H. Multi-task correlation particle filter for robust object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4335–4343.
78. Choi, J.; Jin Chang, H.; Yun, S.; Fischer, T.; Demiris, Y.; Young Choi, J. Attentional correlation filter network for adaptive visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4807–4816.
79. Bibi, A.; Mueller, M.; Ghanem, B. Target response adaptation for correlation filter tracking. In *European Conference on Computer Vision*; Springer: New York, NY, USA, 2016; pp. 419–433.
80. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking. *arXiv* **2018**, arXiv:1803.08679.
81. Bai, S.; He, Z.; Xu, T.B.; Zhu, Z.; Dong, Y.; Bai, H. Multi-hierarchical Independent Correlation Filters for Visual Tracking. *arXiv* **2018**, arXiv:1811.10302.
82. Crammer, K.; Dekel, O.; Keshet, J.; Shalev-Shwartz, S.; Singer, Y. Online passive-aggressive algorithms. *J. Mach. Learn. Res.* **2006**, *7*, 551–585.
83. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2019.
84. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. *arXiv* **2018**, arXiv:1812.11703.
85. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2019

86. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M.H. Target-Aware Deep Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 24–27 June 2019
87. Fan, H.; Ling, H. Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking. *arXiv* **2018**, arXiv:1812.06148.
88. Wang, G.; Luo, C.; Xiong, Z.; Zeng, W. SPM-Tracker: Series-Parallel Matching for Real-Time Visual Object Tracking. *arXiv* **2019**, arXiv:1904.04452.
89. Zhang, Y.; Wang, L.; Qi, J.; Wang, D.; Feng, M.; Lu, H. Structured Siamese network for real-time visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 351–366.
90. Dong, X.; Shen, J. Triplet loss in Siamese network for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 459–474.
91. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning attentions: residual attentional Siamese network for high performance online visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4854–4863.
92. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic Siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
93. Minnehan, B.; Taufique, A.M.N.; Savakis, A. Fully convolutional adaptive tracker with real time performance. In Proceedings of the SPIE Geospatial Informatics IX, Baltimore, MD, USA, 13 May 2019; Volume 10992.
94. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; NIPS: Barcelona, Spain, 2015; pp. 91–99.
95. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold Siamese network for real-time object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4834–4843.
96. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5296–5305.
97. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: New York, NY, USA, 2014, pp. 740–755.
98. Ren, L.; Yuan, X.; Lu, J.; Yang, M.; Zhou, J. Deep reinforcement learning with iterative shift for visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 684–700.
99. Dong, X.; Shen, J.; Wang, W.; Liu, Y.; Shao, L.; Porikli, F. Hyperparameter optimization for tracking with continuous deep q-learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 518–527.
100. Supancic, J., III; Ramanan, D. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 322–331.
101. Huang, C.; Lucey, S.; Ramanan, D. Learning policies for adaptive tracking with deep feature cascades. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 105–114.