

Article

An Anchor-Free Siamese Network with Multi-Template Update for Object Tracking

Tongtong Yuan ^{1,2}, Wenzhu Yang ^{1,2,*}, Qian Li ^{1,2} and Yuxia Wang ^{1,2}

¹ School of Cyber Security and Computer, Hebei University, Baoding 071002, China; hdwjytt@163.com (T.Y.); lee55777@163.com (Q.L.); yuxiawang1996@163.com (Y.W.)

² Hebei Machine Vision Engineering Research Center, Hebei University, Baoding 071002, China

* Correspondence: wenzhuyang@hbu.edu.cn; Tel.: +86-15720127565

Abstract: Siamese trackers are widely used in various fields for their advantages of balancing speed and accuracy. Compared with the anchor-based method, the anchor-free-based approach can reach faster speeds without any drop in precision. Inspired by the Siamese network and anchor-free idea, an anchor-free Siamese network (AFSN) with multi-template updates for object tracking is proposed. To improve tracking performance, a dual-fusion method is adopted in which the multi-layer features and multiple prediction results are combined respectively. The low-level feature maps are concatenated with the high-level feature maps to make full use of both spatial and semantic information. To make the results as stable as possible, the final results are obtained by combining multiple prediction results. Aiming at the template update, a high-confidence multi-template update mechanism is used. The average peak to correlation energy is used to determine whether the template should be updated. We use the anchor-free network to implement object tracking in a per-pixel manner, which computes the object category and bounding boxes directly. Experimental results indicate that the average overlap and success rate of the proposed algorithm increase by about 5% and 10%, respectively, compared to the SiamRPN++ algorithm when running on the dataset of GOT-10k (Generic Object Tracking Benchmark).

Keywords: object tracking; Siamese network; anchor free; dual fusion; multi-template update



Citation: Yuan, T.; Yang, W.; Li, Q.; Wang, Y. An Anchor-Free Siamese Network with Multi-Template Update for Object Tracking. *Electronics* **2021**, *10*, 1067. <https://doi.org/10.3390/electronics10091067>

Academic Editors: Jun Liu and Savvas A. Chatzichristofis

Received: 27 February 2021

Accepted: 27 April 2021

Published: 30 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Visual object tracking is a fundamental research direction in computer vision. It is widely used in diverse fields such like visual surveillance, vehicle tracking, and human-computer interaction [1]. Visual object tracking, which detects and locates a specified target in a changing video sequence, depends on the ground-truth bounding box of the initial frame and then obtains the complete trajectory of the target. Recently, rapid progress has been made in visual object tracking. However, it is still a great challenge in real-world applications, as objects under unconstrained recording conditions often suffer from illumination variation, heavy occlusion, background clutters, and scale deformation, to name a few [1]. Modern trackers can be roughly divided into methods based on correlation filters or deep learning.

Recently, Siamese network-based trackers [2–6] have drawn considerable attention due to their outstanding tracking performance, especially well-balanced accuracy and speed. Siamese trackers formulate the visual object tracking task as a target-matching problem that learns a general similarity map by cross-correlation operation between the feature representations learned for the target template and the search region. Since one single similarity map contains limited spatial and semantic information, a region proposal network (RPN) [7] is attached to the Siamese network. By jointly training both classification and regression branches, and by setting default anchor bounding boxes of numerous fixed sizes and aspect ratios on the similarity maps, these methods discard the time-consuming step of extracting multi-scale feature maps for the target scale invariance. Although these

anchor-based trackers achieve state-of-the-art results on tracking benchmarks, they are still limited by pre-defined anchors that lead to many parameters, such as numbers and aspect ratios of anchor boxes.

In this paper, an anchor-free Siamese network (AFSN) for object tracking is proposed. It is built on a ResNet-50 [8] network and combines multi-level features with rich spatial and semantic information. It combines multiple prediction results to make the final result more stable and smoother. An anchor-free mechanism is adopted to predict the object category and location in a per-pixel manner, which can avert a large number of complicated calculations and complex parameters related to anchor boxes.

Our main contributions are as follows:

1. We propose an anchor-free Siamese network (AFSN) for object tracking, which can perform end-to-end online training and offline tracking. It changes the original strides and receptive field and eventually achieves powerful performance.
2. A dual-fusion method is designed to combine feature maps and prediction results. The high-level features are added to the low-level and middle-level features to make full usage of both spatial and semantic information. Application of the weighted-sum method to multiple prediction results can improve accuracy and boost robustness.
3. A multi-template update mechanism is designed to determine whether the template should be updated. The score of the peak to correlation energy is used to measure the degree of occlusion of the object and ensure the effectiveness of the template.
4. We present a proposal of replacing the RPN module with the anchor-free prediction network, which can decrease the number of hyper-parameters, make the tracker simpler, speed up the tracking process, and enhance performance.
5. The proposed method achieves state-of-the-art performance on GOT-10k [9], LaSOT (Large-Scale Single Object Tracking) [10], UAV123 (A Benchmark and Simulator for UAV Tracking) [11], and OTB100 (Object Tracking Benchmark 100) [12] tracking datasets.

2. Related Works

Visual object tracking has always been a hot research direction. In recent years, tracking researchers have been focusing on improving speed and accuracy from different aspects such as feature extraction, template updating, classifier design, and location regression [6]. With the development of various methods, significant progress has been achieved. Siamese trackers have drawn great attention and achieved outstanding tracking performance. This section mainly introduces two aspects: Siamese trackers and anchor-free detection algorithms.

2.1. Siamese Network-Based Trackers

A Siamese network consists of two subnetworks with the same structures sharing weights. The proposed fully convolutional Siamese networks (SiamFC) [2] first introduces the correlation operation to produce a similarity map by the method of sliding windows. The highest score represents the target position. It is simple in construction but effective in improving accuracy. SiamRPN (Siamese Region Proposal Network) [3] constructs a Siamese network following an RPN module. It decomposes the tracking task as classification and regression for the object. SiamRPN sets a number of pre-defined anchor boxes to evade the time-consuming step of extracting multi-scale feature maps for object scale invariance and achieves robust results. DaSiamRPN (Distractor-aware SiamRPN) [4] introduces a sample training strategy to train the distractor-aware module and a local-to-global search strategy for long-term tracking. Through increasing high-quality positive and hard negative samples, it improves the discrimination of the tracker and achieves robust tracking performance. Siamese C-RPN (Siamese Cascaded Region Proposal Network) [5] constructs a sequence of RPNs cascaded from deep to shallow layers in the backbone network. Such cascade structure can balance training samples by motivating hard negative sampling. The cascaded RPNs are more discriminative in distinguishing difficult and complex distractors.

Up to now, these Siamese trackers cannot make use of features from deep networks. To deal this problem, SiamRPN++ [6] proposes a spatial-aware sampling strategy to avoid putting a strong center bias on objects and then successfully trains the Siamese tracker by using the ResNet as a feature extraction network. It also adopts a method to aggregate multiple output results obtained by multi-layer feature maps. Experimental results show that Siamese trackers driven by deep networks can achieve better tracking performance.

To address tedious and heuristic configurations, the Siamese Box Adaptive Network (SiamBAN) [7] views the tracking task as a parallel classification and regression problem, and directly classifies objects and regresses their bounding boxes in a unified FCN (Fully Convolutional Network). ATOM (Accurate Tracking by Overlap Maximization) [13] introduces a classification component that is trained online to guarantee high discriminative power in the presence of distractors. Xuesong Chen et al. [14] present a new optimization objective function with dual-attention mechanisms to generate adversarial perturbations for ensuring the efficiency of the one-shot attack. SiamAttn [15] introduces a new Siamese attention mechanism that computes deformable self-attention and cross-attention. The self-attention captures rich context information. The cross-attention aggregates contextual inter-dependencies between template and search region. Siam R-CNN (Siamese Re-detection Convolutional Neural Network) [16] takes advantage of re-detections of both the first-frame template and previous-frame predictions to model the target. The tracklet-based dynamic programming algorithm enables to re-detect targets after occlusion.

These Siamese trackers above are totally adopted from a set of pre-defined anchors to evade time-consuming computations, which can significantly improve the tracking performance in terms of both accuracy and speed. However, since these trackers need to define anchor boxes with fixed size and aspect ratios, they still have difficulty in tracking targets with large-scale deformation and pose change.

2.2. Anchor-Free Method of Detection

The current mainstream detectors such as Faster-RCNN [17], SSD (Single Shot Multi-Box Detector) [18], and YOLOv3 (You Only Look Once version 3) [19] rely on a set of pre-defined anchor boxes and achieve state-of-the-art performance. However, these anchor-based detectors have several disadvantages. On the one hand, a set of anchor boxes needs to be pre-defined with large parameters and fixed hyper-parameters, and the detection performance is sensitive to the hyper-parameters related to anchors. On the other hand, to cope well with scale deformation problems, the detectors set a large number of anchor boxes, causing a serious imbalance between positive and negative samples [20].

Therefore, many detectors based on the anchor-free method have been proposed recently. A fully convolutional one-stage object detector (FCOS) [20] eliminates the pre-defined set of anchor boxes and solves object detection in a per-pixel prediction fashion. It completely escapes the large parameters and complex computations related to anchors. CenterNet [21] detects each object using a triplet, including one center keypoint and two corners. These anchor-free approaches can achieve comparable performance to the anchor-based method, but with faster speed.

3. Methodology

In this section, we introduce the proposed anchor-free Siamese network in detail. As shown in Figure 1, the AFSN adopts the Siamese network to extract multi-level deep features and computes similarity maps between the template and the search region, followed by multiple anchor-free prediction networks. The backbone network consists of two subnetworks with the same structure sharing weights. The proposed dual-fusion method is divided into combining feature maps and prediction results. The high-level features are added to the low-level and middle-level features. The final result is the sum of the multiple prediction results. The anchor-free prediction networks discard the pre-defined anchor boxes and consist of classification and regression networks in per-pixel prediction.

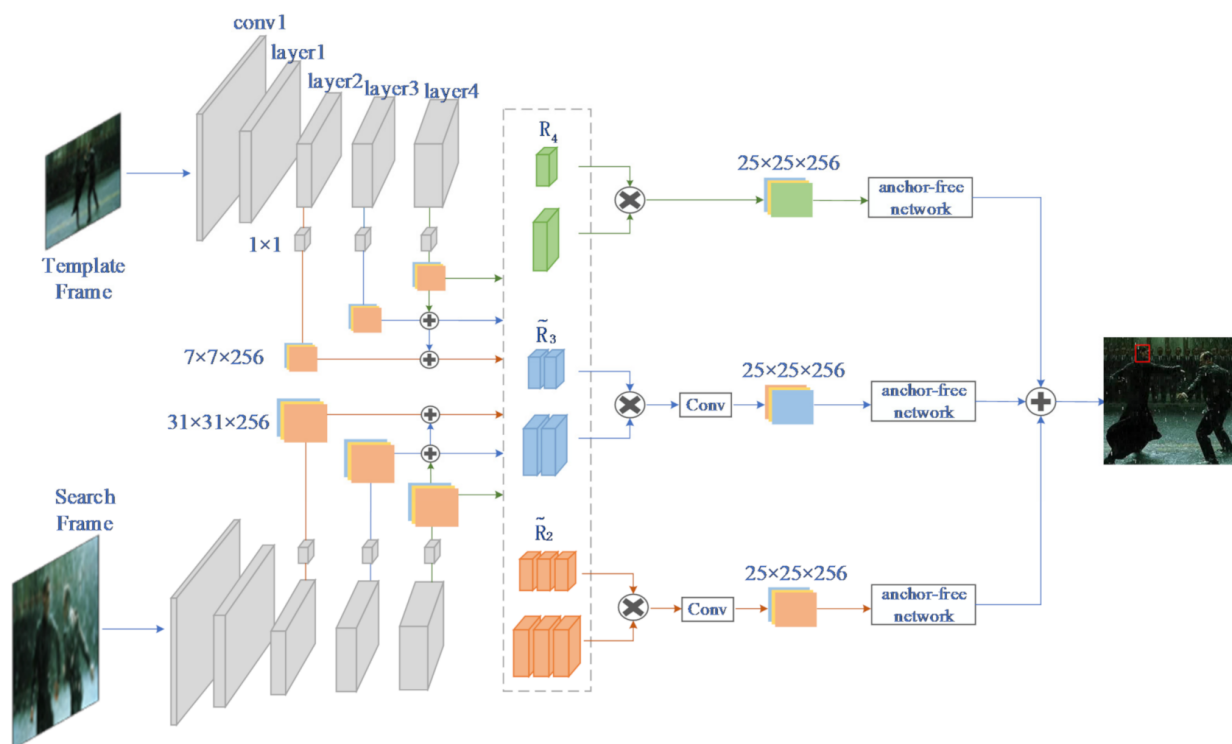


Figure 1. The architecture of our AFSN (Anchor-Free Siamese Network).

3.1. Feature Extraction with a Siamese Network

Previous Siamese networks were designed to be shallow to satisfy strict translation invariance. Recently, the backbone network for object detection and semantic segmentation task has been gradually replaced by deep networks, which lead to decreased performance [1]. However, almost every modern network adds padding structures to make it go deeper, which destroys the strict translation invariance restriction. To this end, SiamRPN++ [6] has conducted detailed analysis experiments on intrinsic factors, obtaining the following quantitative conclusions:

(1) A padding structure leads to a spatial bias because of the violation of the restriction. A spatial-aware sampling strategy with a suitable shift can avoid putting a strong center bias on targets.

(2) The original modern networks generally have a larger stride, which is not suitable for Siamese trackers.

Based on the results of the above analysis, the proposed AFSN adopts the modified ResNet-50 [8] as its backbone network. We reduce the original strides of res3 and res4 blocks from 16 and 32 pixels to 8 pixels and increase the receptive field by dilated convolution operation. Meanwhile, a spatial-aware sampling strategy is adopted to train the whole network. To reduce the number of parameters, we change the channels of multi-level feature maps to 256 by a 1×1 convolution operation. The backbone network consists of two subnetworks: a target subnetwork, which extracts feature maps of the tracking template patch T , and a search subnetwork, which extracts feature maps of the search region S . The two subnetworks share the same convolutional neural network (CNN) architecture with the same parameters.

Compared with shallow networks such as AlexNet, ResNet-50 can aggregate multiple layers to retain richer information. Multi-level features can provide different representations. Low-level and middle-level features primarily focusing on spatial features such as the edge, color, and shape of the target are of great significance to estimate the target location more accurately. High-level features have better representations on semantic information, which are of vital importance to distinguish similar distractors and can be beneficial during

some challenging scenarios such as huge deformation and fast motion. Compounding these representations can improve the inference of classification and localization.

In our network, we use feature maps extracted from the last three residual blocks of ResNet-50, represented as $\{R_2, R_3, R_4\}$. The process of fusion is shown in Figure 1, and these blocks are concatenated as a unity:

$$\tilde{R}_3 = \text{Cat}(R_4, R_3) \quad (1)$$

where R_4 and R_3 include 256 channels. Hence \tilde{R}_3 has 2×256 channels.

Then \tilde{R}_3 and R_2 are concatenated as a unity:

$$\tilde{R}_2 = \text{Cat}(\tilde{R}_3, R_2) \quad (2)$$

where \tilde{R}_3 and R_2 include 2×256 and 256 channels, respectively. Hence \tilde{R}_2 has 3×256 channels.

Finally, the obtained feature maps $\{\tilde{R}_2, \tilde{R}_3, R_4\}$ will be used in the following tracking network.

To embed the information of feature maps from target patch T and search region S , the response map \mathbb{R} can be obtained by cross-correlation operation. We set the extracted feature map of target patch T and search region S as R^T and R^S , respectively. Since the following prediction networks take the response map \mathbb{R} as input to generate the location and classification information of the target, the abundant information of the response map is essential for object prediction. We use the depth-wise cross-correlation operation to calculate the response maps on R^S with R^T as a kernel to retain massive information. We compute the response maps by using

$$\begin{aligned} \mathbb{R}_2 &= \tilde{R}_2^S * \tilde{R}_2^T \\ \mathbb{R}_3 &= \tilde{R}_3^S * \tilde{R}_3^T \\ \mathbb{R}_4 &= \tilde{R}_4^S * \tilde{R}_4^T \end{aligned} \quad (3)$$

where $*$ denotes the channel-by-channel depth-wise cross-correlation operation. The response maps have the same number of channels as R^T ($3 \times 256, 2 \times 256, 256$). The response maps are then convoluted with a 1×1 kernel to change its channels to 256; as a result, the subsequent prediction computation can be significantly sped up. As shown in Figure 1, the output of $\mathbb{R}_2, \mathbb{R}_3$ and \mathbb{R}_4 is fed into three prediction subnetworks individually.

3.2. Anchor-Free Prediction Network

For each point (i, j) in the response map \mathbb{R} , it can be mapped back onto the search region S as (x, y) by computing the corresponding strides.

$$\begin{aligned} x &= \lfloor \frac{s}{2} \rfloor + i * s \\ y &= \lfloor \frac{s}{2} \rfloor + j * s \end{aligned} \quad (4)$$

where s denotes the effective strides.

The anchor-free prediction operation completely eliminates the complicated computations and tricky parameter tuning related to anchor bounding boxes. If the point (x, y) falls into the ground-truth bounding box, it can be considered a positive training sample, and otherwise, a negative training sample. The anchor-free prediction subnetwork is shown in Figure 2. The total prediction subnetwork can be divided into two branches: a classification branch to classify the category for each point and a regression branch to regress the target bounding box at this point. For each response map $\mathbb{R}_{w \times h \times m}$, the classification network outputs a classification feature map $A_{w \times h \times 2}^{cls}$ and its two dimensions represent the foreground and background scores of the corresponding point, respectively. The regression branch outputs a regression feature map $A_{w \times h \times 4}^{reg}$ and it encodes the location of a predicted bounding box at this point. We set regression output $A^{reg}(i, j, :) = (l, t, r, b)$ correspondent to the point $(i, j, :)$, which represents the distance from this point to the four sides

of the bounding box. To suppress numerous low-quality bounding boxes generated by locations far away from the center of the object, following [20], we add a center-ness branch in parallel with the classification branch to predict the distance between the location and the center of the object. The center-ness branch outputs a center-ness feature map $A_{w \times h \times 1}^{cen}$.

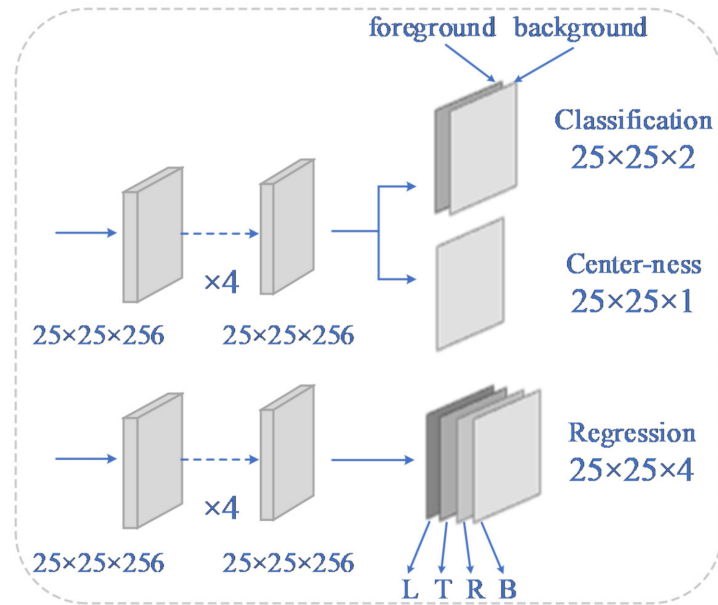


Figure 2. The architecture of the anchor-free network. L, T, R, B represents the distance from this point to the four sides of the bounding box respectively.

Since the output feature maps of the three anchor-free prediction subnetworks have the same spatial resolution, the weighted sum is adopted directly on the output. A weighted-fusion method combines all the outputs.

$$C_{all} = \sum_{l=2}^4 \alpha_l * C_l, R_{all} = \sum_{l=2}^4 \beta_l * R_l, Cen_{all} = \sum_{l=2}^4 \lambda_l * Cen_l. \tag{5}$$

where C , R and Cen denote the outputs of classification, regression, and center-ness branches, respectively. The combination weights (α , β , γ) can be end-to-end trained offline together with the whole network.

3.3. Multi-Template Update Mechanism

Traditional Siamese trackers adopt the features extracted from the first frame as the template. The target often suffers from serious occlusions, large-scale variation, similar background clutters, etc. During the whole tracking process, it is difficult to solve the problems using only the one template extracted from the first frame. Therefore, we adopt a high-confidence multi-template update mechanism to update templates extracted from the last three residual blocks of the backbone network.

To prevent the features of distractors and background to be added into templates, we use the score of the peak to correlation energy to ensure the effectiveness of templates. The APCE (Average Peak-to Correlation Energy) can reflect the degree of occlusion and can be calculated by

$$\begin{aligned} F_{max} &= \max(\mathbb{R}) \\ F_{min} &= \min(\mathbb{R}) \\ APCE &= \frac{|F_{max} - F_{min}|^2}{\text{mean} \left(\sum_{w,h} (F_{w,h} - F_{min})^2 \right)} \end{aligned} \tag{6}$$

where F_{max} , F_{min} , and $F_{w,h}$ denote the maximum, minimum, and corresponding values at the point (w, h) in the response map, respectively. The numerator reflects the peak value, and the denominator represents the fluctuation of the response map. The peak value and the fluctuation can indicate the confidence degree about the tracking results. When the target is not occluded, the *APCE* becomes larger and the response map shows only one sharp peak. Otherwise, the *APCE* dramatically decreases if the target is occluded or missing.

We compute the *APCE* as the sum of multiple response maps ($\mathbb{R}_2, \mathbb{R}_3, \mathbb{R}_4$) and determine whether the threshold is exceeded. An *APCE* greater than the threshold reveals that the result is reliable. Then, we can upgrade templates by using

$$R^T = R^T * \eta + (1 - \eta) * R^X \quad (7)$$

where η denotes the update ratio, R^T denotes template features, and R^X denotes the features extracted from high-confidence results.

3.4. Loss Function

Let $(x_0, y_0), (x_1, y_1)$ denote the left-top and right-bottom corner point, respectively, of the ground-truth bounding box. The output of the regression branch at point $(i, j, :)$ can be set as $A^{reg}(i, j, :) = (l, t, r, b)$ and calculated by

$$\begin{aligned} l &= x - x_0, t = y - y_0 \\ r &= x_1 - x, b = y_1 - y \end{aligned} \quad (8)$$

Then, the IOU between the predicted results and the ground truth can be computed. Here, we only compute the IOU with positive samples, otherwise, it is set to 0. Then we calculate the regression loss by using

$$\begin{aligned} L_{reg} &= \frac{\sum_{i,j} f(A^{reg}(i, j, :)) L_{IOU}(A^{reg}(i, j, :), g(x, y))}{\sum f(A^{reg}(i, j, :))} \\ f(A^{reg}(i, j, :)) &= \begin{cases} 1 & l > 0, t > 0, r > 0, b > 0 \\ 0 & otherwise \end{cases} \end{aligned} \quad (9)$$

where $g(x, y)$ denotes the ground-truth bounding box.

The score $Cen(i, j)$ in center-ness output is defined by

$$Cen(i, j) = \sqrt{\frac{\min(l, r)}{\max(l, r)} \times \frac{\min(t, b)}{\max(t, b)}} \quad (10)$$

If the point (i, j) is a negative sample, it should be regressed here. The center-ness loss can be computed by using

$$\begin{aligned} L_{cen} &= \frac{-\sum_{i,j} f(A^{reg}(i, j, :)) * Cen(i, j) * \log A^{cen}(i, j)}{\sum f(A^{reg}(i, j, :))} \\ &+ (1 - f(A^{reg}(i, j, :)) * Cen(i, j)) * \log(1 - A^{cen}(i, j)) \end{aligned} \quad (11)$$

The loss function of the AFSN consists of the classification loss function, the center-ness loss function, and the regression loss function, which can be calculated by using

$$L = L_{cls} + \lambda_1 L_{cen} + \lambda_2 L_{reg} \quad (12)$$

where L_{cls} represents the cross-entropy loss for the classification branch.

4. Results and Discussion

Our experiments were implemented in Python with Pytorch on one Nvidia Titan 1080Ti GPU (Graphics Processing Unit). The backbone network of our architecture is pre-trained on ImageNet [22] using the parameters as initialization to retrain our model. We trained the whole network on the training sets of COCO (Microsoft Common Objects in Context) [23], ImageNet DET [24], ImageNet VID [24], and YouTube-Bounding Boxes [25] for experiments on GOT-10K [9], LaSOT [10], UAV123 [11], and OTB100 [12]. Specifically, we set a 127×127 region centered on the ground-truth bounding box as a template patch so as to set a 255×255 region as the search region.

During the training process, the proposed AFSN can be trained end to end. There are in total 50 epochs performed, 6000 sample pairs per epoch, by using stochastic gradient descent (SGD) with a learning rate exponentially decayed from 0.01 to 0.0001. A weight decay of 0.0005 and a momentum of 0.9 are used. To substantially boost performance, the parameters of the backbone network are frozen while training the anchor-free prediction subnetwork in the first 10 epochs. In the last 40 epochs, the last three blocks of ResNet-50 are unfrozen to be trained together by setting the learning rate to be 10 times smaller than the anchor-free prediction parts. Especially, we use a warm-up learning rate of 0.001 in the first five epochs to train the anchor-free prediction network.

4.1. Results on GOT-10k

GOT-10k is a recently released large, high-diversity benchmark for generic object tracking in the wild [14]. It collects over 10,000 video segments and manually annotates more than 1.5 million high-precision bounding boxes. The class between training and testing sets are zero-overlapped. It provides class-balanced metrics *mAO* and *mSR* for the evaluation of generic object trackers and builds an official website that offers an online evaluation server [14]. Authors need to test their models on the given testing dataset and upload the tracking results from the website. The provided evaluation indicators include success plots, the average overlap (*AO*), and the success rate (*SR*). The *AO* denotes the average of overlaps between all ground-truth boxes and estimated bounding boxes. The *SR* denotes the percentage of successfully tracked frames where the overlaps exceed a threshold. $SR_{0.5}$ and $SR_{0.75}$ represent the rate of successfully tracked frames whose overlap exceeds 0.5 and 0.75, respectively.

In this experiment, we compared our method with several trackers, including SiamRPN++ [6], CCOT [26], SPM [27], and Staple [28]. All the results are provided by the official website of GOT-10k. Table 1 lists the comparison details of different indicators. It shows that the proposed AFSN is able to rank first in *AO*, $SR_{0.5}$, and $SR_{0.75}$. Results of lines 3 and 4 of the table suggest that compared with SiamRPN++, the proposed AFSN improves the score by 5.8% on $SR_{0.5}$ and 10% on $SR_{0.75}$. As shown in Figure 3, the proposed AFSN can outperform other trackers and obtains a 5% gain from SiamRPN++ in the overlap success rate. Results validate that the AFSN tracker can estimate more precise bounding boxes and have good generalization for a visual object.

When the dual-fusion method and multi-template update mechanism are removed, the proposed algorithm can run at 42 fps. Under the same conditions, the SiamRPN++ algorithm runs at 38.71 fps. The running speed of the proposed algorithm is higher than that of the anchor-based algorithm. This indicates that the anchor-free prediction network can decrease the number of hyper-parameters, make the tracker simpler, and speed up the tracking process.

In this part, we qualitatively compare the proposed AFSN with four different trackers in Figure 4. As our AFSN concatenates deeper semantic information and low-level spatial information, the tracking results shown in the figure can distinguish targets and background accurately. Furthermore, the AFSN can locate the target more accurately with minimum error.

Table 1. Comparisons on GOT-10k (Generic Object Tracking Benchmark). KCF: Kernelized Correlation Filter; SRDCF: Spatially Regularized Correlation Filters; SAMF: Scale Adaptive Multiple Feature; DSST: Discriminative Scale Space Tracker; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; ECO: Efficient Convolution Operators; CFnet: Correlation Filter Network; MDnet: Multi-Domain convolutional neural Networks; CCOT: Continuous Convolution Operators Tracking; SiamFC: Fully-Convolutional Siamese Networks; SiamRPN_R18: Siamese Region Proposal Network with ResNet18; SPM: Series-Parallel Matching; SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks; AFSN: Anchor-Free Siamese Network. AO: Average overlap; SR: Success rate; FPS: Frames Per Second.

Trackers	AO	SR _{0.5}	SR _{0.75}	FPS
KCF [29]	0.203	0.177	0.065	94.66
SRDCF [30]	0.236	0.227	0.094	5.58
Staple [28]	0.246	0.239	0.089	28.87
SAMF [31]	0.246	0.241	0.084	7.43
DSST [32]	0.247	0.223	0.081	18.25
MEEM [33]	0.253	0.235	0.068	20.59
ECO-HC [34]	0.286	0.276	0.096	44.55
CFnet [35]	0.293	0.265	0.087	35.62
MDnet [36]	0.299	0.303	0.099	1.52
ECO [34]	0.316	0.309	0.111	2.62
CCOT [26]	0.325	0.328	0.107	0.68
SiamFC [2]	0.374	0.404	0.144	25.81
SiamRPN_R18 [3]	0.483	0.581	0.270	97.55
SPM [26]	0.513	0.593	0.359	72.30
SiamRPN++ [6]	0.508	0.601	0.313	38.71
AFSN	0.558	0.659	0.413	28.09

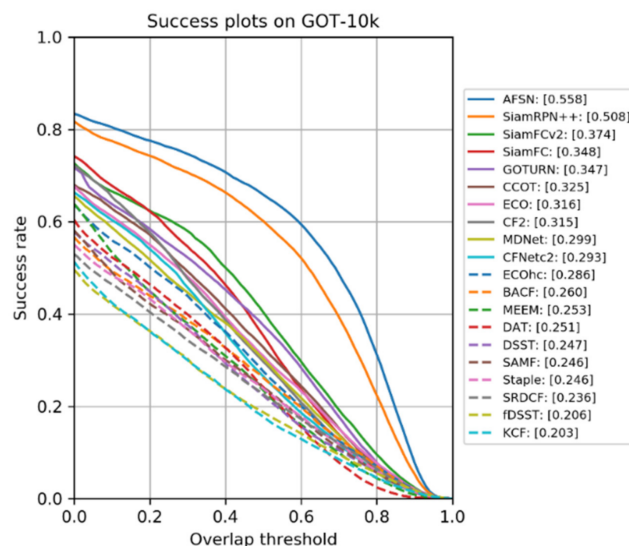


Figure 3. Comparisons on GOT-10k (Generic Object Tracking Benchmark). Our AFSN significantly outperforms other methods. AFSN: Anchor-Free Siamese Network; SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks; SiamFCv2: Fully-Convolutional Siamese Networks version 2; SiamFC: Fully-Convolutional Siamese Networks; GOTURN: Generic Object Tracking Using Regression Networks; CCOT: Continuous Convolution Operators Tracking; ECO: Efficient Convolution Operators; MDNet: Multi-Domain convolutional neural Networks; BACF: Background-Aware Correlation Filters; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; DAT: Deep Attentive Tracking; DSST: Discriminative Scale Space Tracker; SAMF: SAMF: Scale Adaptive Multiple Feature; SRDCF: Spatially Regularized Correlation Filters; KCF: Kernelized Correlation Filter.

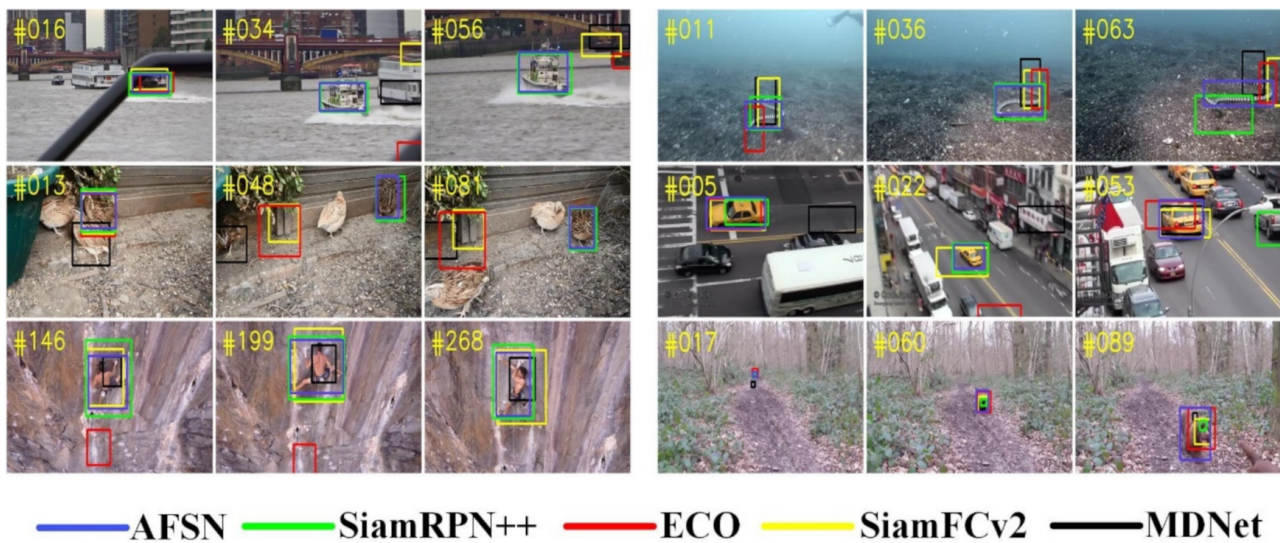


Figure 4. Qualitative result comparison of the proposed method with other trackers. AFSN: Anchor-Free Siamese Network; SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks; ECO: Efficient Convolution Operators; SiamFCv2: Fully-Convolutional Siamese Networks version 2; MDNet: Multi-Domain convolutional neural Networks.

4.2. Results on LaSOT

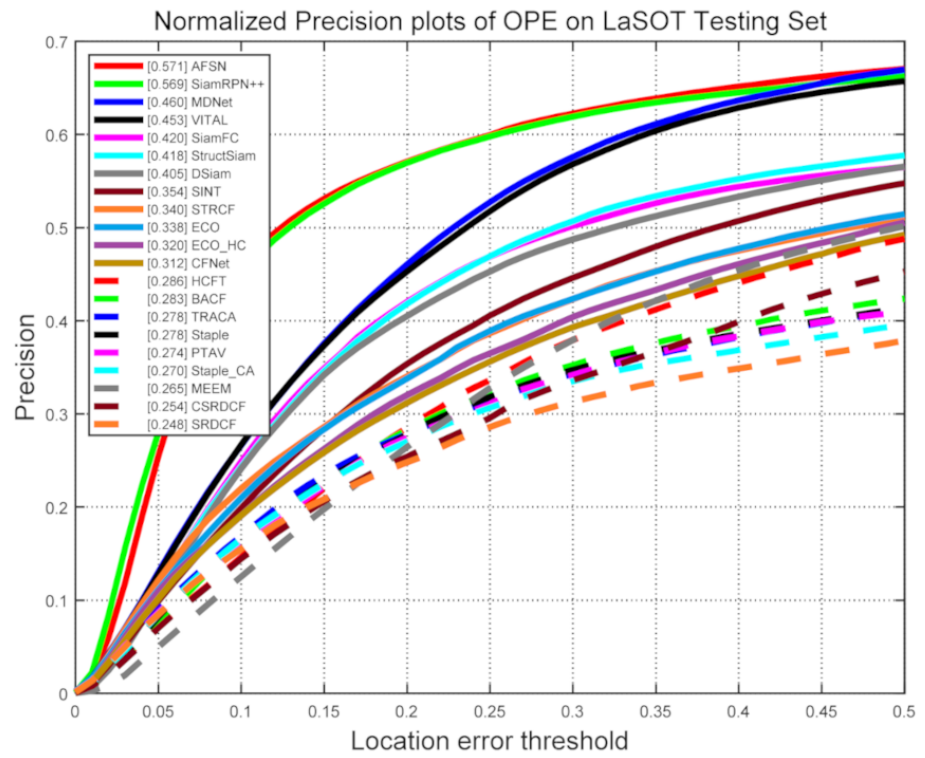
To further validate the proposed AFSN on a more challenging dataset, we conducted experiments on LaSOT. The dataset contains more than 3.52 million manually annotated frames and 1400 videos [10]. It contains 70 classes and each class includes 20 tracking sequences. The official website of LaSOT provides 280 videos with high-quality dense annotations in the testing set and 35 algorithms as baselines. The provided evaluation indicators include normalized precision plots, precision plots, and success plots in one-pass evaluation (OPE).

We compared our tracker with the top 20 trackers, including SiamRPN++ [6], MDNet [36], MEEM [33], ECO [34], SiamFC [2], DSiam [37], and other baselines. Figure 4 reports the overall performance of our AFSN on the testing set. From Figure 5, we observe that the proposed AFSN outperforms the best tracker by 0.2%, 0.4%, and 0.7%, respectively, for the three indicators. Notably, compared with the provided baselines, our AFSN heightens the scores by more than 11%, 12%, and 10%, respectively, for the three indicators over MDNet, which is the best tracker reported on the official website of the LaSOT dataset.

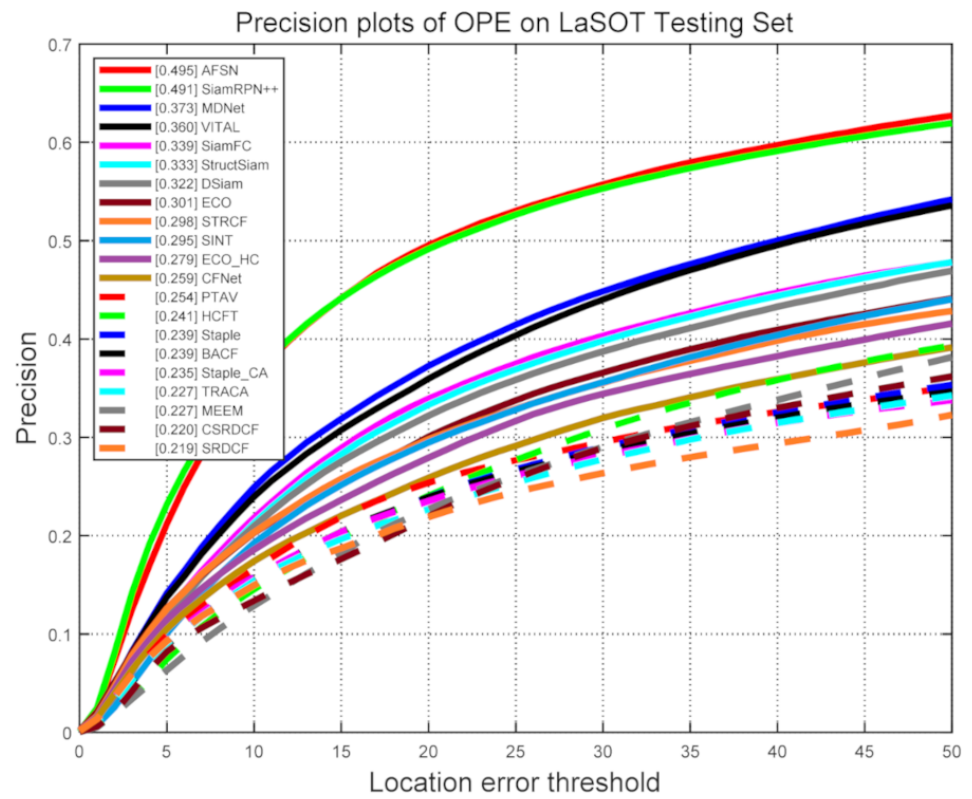
4.3. Results on UAV123

The UAV123 dataset includes in total 123 video sequences comprising more than 110,000 frames [11]. The objects in the dataset mainly suffer from an aspect ratio change, background clutter, fast motion, and illumination variation, which make tracking challenging.

We tested the proposed tracker on the UAV123 dataset in comparison with several representative trackers, including ATOM [13], RLS-RTMDNet [38], DaSiamRPN [4], SiamRPN [3], ECO [34], SRDCF [30], MEEM [33], and KCF [29]. We show the results of the comparison with the UAV123 dataset in Table 2 and Figure 6. As shown in Table 2, our AFSN obtains the best results, with an AUC of 61.3%, $OP_{0.50}$ of 75.8%, and $OP_{0.75}$ of 55.4%. Our AFSN improves the scores by 9.7%, 12.5%, and 23.4%, respectively, for the three indicators over RLS-RTMDNet. Compared with the ATOM algorithm with Siamese-based architecture, our AFSN improves the scores by 0.7% on $OP_{0.50}$ and 7.8% on $OP_{0.75}$.

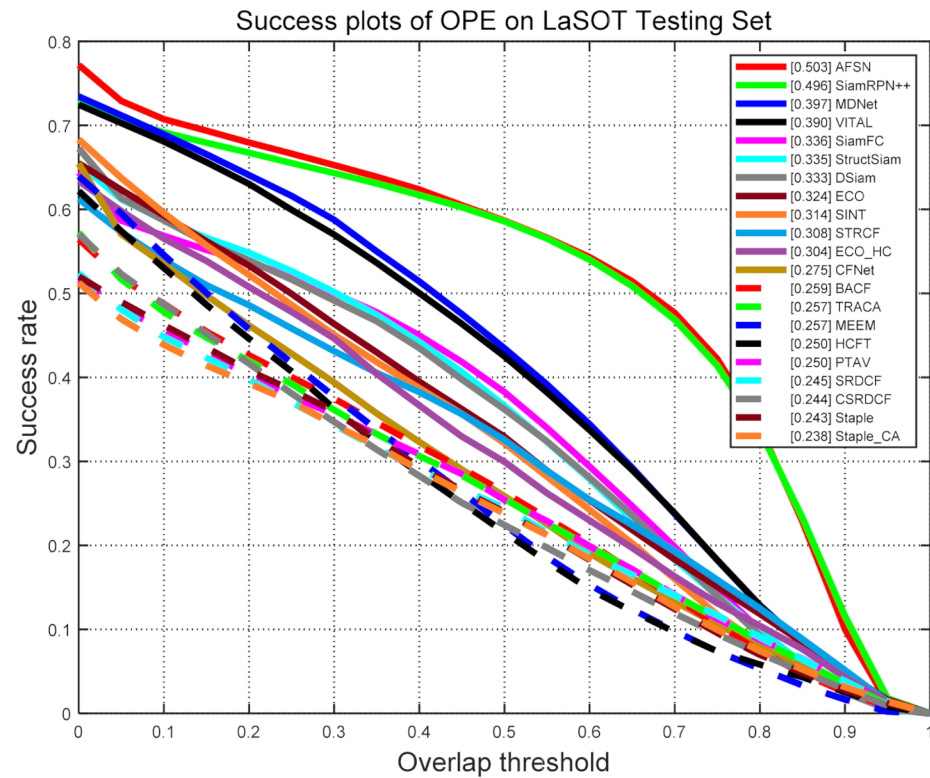


(a)



(b)

Figure 5. Cont.



(c)

Figure 5. Comparisons among top 20 trackers on LaSOT. (a) Normalized precision plots, (b) precision plots, and (c) success plots. Our AFSN significantly outperforms other methods in precision and success rate. AFSN: Anchor-Free Siamese Network; SiamRPN++: Evolution of Siamese VisualTracking with Very Deep Networks; MDNet: Multi-Domain convolutional neural Networks; VITAL: Visual Tracking via Adversarial Learning; SiamFC: Fully-Convolutional Siamese Networks; StructSiam: Structured Siamese Network; DSiam: Dynamic Siamese Network; ECO: Efficient Convolution Operators; SINT: Siamese Instance Search for Tracking; STRCF: Spatial-Temporal Regularized Correlation Filters; CFNet: Correlation Filter Network; BACF: Background-Aware Correlation Filters; TRACA: Context-aware Deep Feature Compression; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; HCFT: Hierarchical Convolutional Features Tracking; PTAV: Parallel Tracking and Verifying; SRDCF: Spatially Regularized Correlation Filters.

Table 2. Comparisons on the UAV123 dataset in terms of AUC and mean OP , given the thresholds 0.50 ($OP_{0.50}$) and 0.75 ($OP_{0.75}$). KCF: Kernelized Correlation Filter; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; SRDCF: Spatially Regularized Correlation Filters; ECO: Efficient Convolution Operators; SiamRPN: Siamese Region Proposal Network; DaSiamRPN: Distractor-aware SiamRPN; ATOM: Accurate Tracking by Overlap Maximization; RLS-RTMDNet: Recursive Least-Squares Estimator-Aided Online Learning. AFSN: Anchor-Free Siamese Network.

UAV123	KCF	MEEM	SRDCF	ECO	SiamRPN	DaSiamRPN	ATOM	RLS-RTMDNet	AFSN
$OP_{0.50}$	0.368	0.440	0.551	0.631	0.711	0.725	0.751	0.633	0.758
$OP_{0.75}$	0.144	0.150	0.263	0.324	0.398	0.405	0.476	0.320	0.554
AUC	0.331	0.392	0.464	0.525	0.557	0.569	0.617	0.516	0.613

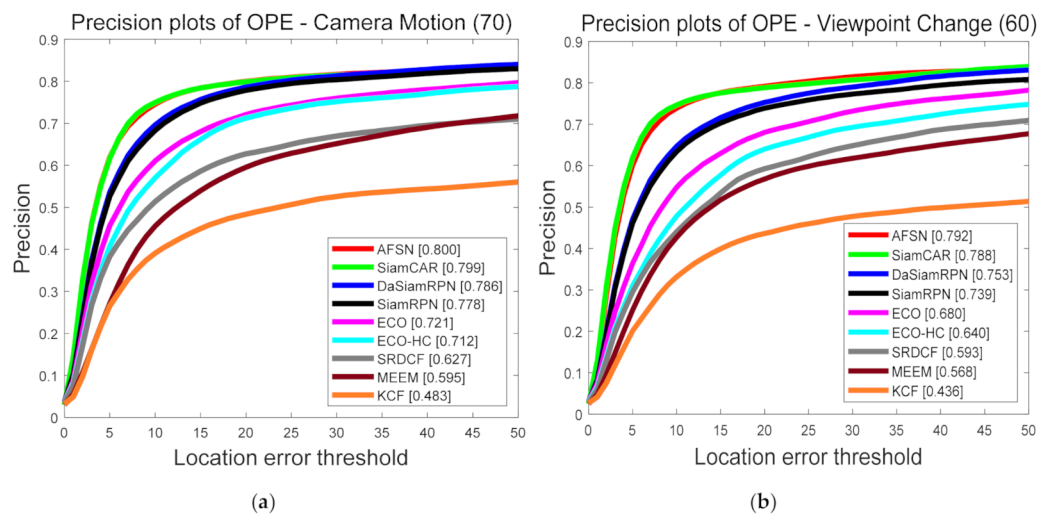


Figure 6. Precision plots for UAV123 in terms of the following attributes: (a) camera motion and (b) viewpoint change. In both cases, our approach improves the state of the art by a certain margin. AFSN: Anchor-Free Siamese Network; SiamCAR: Siamese Fully Convolutional Classification and Regression; DaSiamRPN: Distractor-aware SiamRPN; SiamRPN: Siamese Region Proposal Network; ECO: Efficient Convolution Operators for Tracking; ECO-HC: ECO with HOG and CN features; SRDCF: Spatially Regularized Correlation Filters; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; KCF: Kernelized Correlation Filter.

From Figure 6, we can observe that the proposed AFSN tracker achieves the best performance over the sequences with camera motion and viewpoint change attributes. Compared with SiamCAR [39], our AFSN improves the scores by 0.1% and 0.4%, respectively, for the two aspects. Compared with DaSiamRPN, our AFSN improves the scores by 1.4% and 3.9%, respectively, for the two aspects. The results demonstrate that our proposed network has good representation for a visual object, with deeper and richer feature representation.

4.4. Results on OTB100

OTB100 contains 100 fully annotated sequences tagged with 11 attributes to represent the challenging aspects, including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out of view, background clutters, and low resolution [12]. Each attribute represents a specific challenging factor in object tracking. The evaluation is based on two metrics: precision plot and success plot. The precision plot metric represents the percentage of frames in which the estimated location is within a given threshold distance of the ground-truth position. The success plot metric represents the ratios of successful frames whose overlap is larger than a given threshold ranging from 0 to 1. The area under the curve (AUC) of each success plot is used to rank trackers. In this experiment, we compared our network with eight representative approaches, including SiamRPN [3], Staple [23], and SiamFC [2]. Figure 7 illustrates the precision and success plots of the compared trackers. Compared with SiamRPN, the proposed AFSN improves 0.7% on the success rate and 0.6% on precision. As shown in Table 3, the proposed AFSN significantly improves the tracking success rate for the aspects of background clutter, fast motion, motion blur, out of view, and scale variation. The results demonstrate that the AFSN can better deal with similar distractors and large pose variation, which evidences the importance of richer features in enhancing the robustness and effectiveness of our anchor-free prediction network.

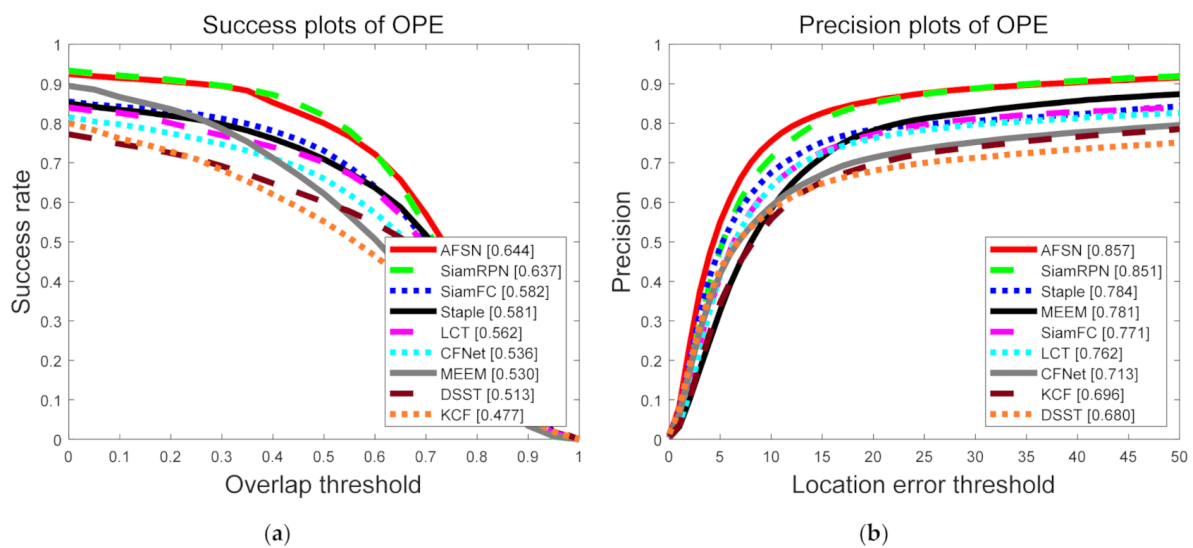


Figure 7. Comparisons on OTB100. (a) Success plots and (b) precision plots. Our AFSN achieves the best results. AFSN: Anchor-Free Siamese Network; SiamRPN: Siamese Region Proposal Network; SiamFC: Fully-Convolutional Siamese Networks; LCT: Long-term Correlation Tracking; CFNet: Correlation Filter Network; MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization; DSST: Discriminative Scale Space Tracker; KCF: Kernelized Correlation Filter.

Table 3. Comparisons on OTB100 with challenging aspects. SiamFC: Fully-Convolutional Siamese Networks; SiamRPN: Siamese Region Proposal Network; AFSN: Anchor-Free Siamese Network.

Challenging Aspects	Trackers			
	SiamFC	Staple	SiamRPN	AFSN
Scale variation	0.556	0.522	0.628	0.641
Deformation	0.510	0.552	0.628	0.599
Illumination variation	0.574	0.596	0.663	0.655
Background clutter	0.523	0.574	0.601	0.628
Motion blur	0.550	0.546	0.627	0.636
Fast motion	0.568	0.537	0.606	0.653
In-plane rotation	0.557	0.552	0.636	0.611
Out-of-plane rotation	0.558	0.534	0.631	0.617
Out of view	0.506	0.481	0.550	0.617
Occlusion	0.547	0.545	0.597	0.589
Low resolution	0.592	0.418	0.597	0.608

4.5. Ablation Study

To study different levers of feature map effectiveness, we performed ablation experiments on OTB100. As shown in Table 4, multi-layer feature maps can improve tracking performance effectively. A model using one single feature map achieves a performance of 0.603 and 0.827 in the AUC and precision, respectively, on OTB100. A model using two feature maps obtains performance gains of 0.022 (0.625 vs. 0.603) in the AUC and 0.018 (0.845 vs. 0.827) in precision on OTB100. A model using three feature maps obtains performance gains of 0.017 (0.642 vs. 0.625) in the AUC and 0.007 (0.852 vs. 0.845) in precision on OTB100. The proposed model with a multi-template update mechanism obtains performance gains of 0.002 (0.644 vs. 0.642) in the AUC and 0.005 (0.857 vs. 0.852) in precision. In addition, adding more feature maps has an obvious contribution to further improvements. The multi-template mechanism is effective for the improvement of precision.

Table 4. Ablation study of the proposed tracker on OTB100. ResNet-50: Residual Network with 50 layers; OTB100: Object Tracking Benchmark 100; AUC: Area Under the Curve.

R_4	ResNet-50		Multi-Template Update	OTB100		
	\tilde{R}_3	R_4		\tilde{R}_2	\tilde{R}_3	R_4
✓					AUC	Precision
	✓				0.603	0.827
		✓			0.625	0.845
			✓		0.642	0.852
			✓	✓	0.644	0.857

5. Conclusions

In this paper, an anchor-free Siamese tracking framework (AFSN) for visual object tracking is proposed. The proposed AFSN adopts a deep network ResNet-50 as its feature extraction network. A dual-fusion method is used to take advantage of multi-layer information, which creates more distinguishable feature representations for dealing with complex distractors. Aiming at the template update, a high-confidence multi-template update mechanism is used to determine whether the template should be updated. In addition, the pre-defined set of anchor boxes is omitted and the object-tracking task is implemented in a per-pixel fashion. The experimental results on GOT-10K [9], LaSOT [10], UAV123 [11], and OTB100 [12] datasets show that the proposed AFSN can achieve state-of-the-art results and run in real time, which proves the generalizability and robustness of our AFSN. We hope that our AFSN can be improved to get better performance after replacing some specific modules in future works.

Author Contributions: Conceptualization, T.Y. and W.Y.; methodology, T.Y. and W.Y.; software, T.Y. and Q.L.; validation, T.Y., Y.W. and Q.L.; formal analysis, W.Y. and T.Y.; investigation, Q.L. and Y.W.; resources, T.Y. and Y.W.; data curation, T.Y. and Q.L.; writing—original draft preparation, T.Y. and W.Y.; writing—review and editing, W.Y.; visualization, T.Y. and Q.L.; supervision, W.Y. and Y.W.; project administration, T.Y. and W.Y.; funding acquisition, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Foundation of Hebei Province under Grant F2021201012 and the Natural Science Foundation of China under Grant 6217070810.

Acknowledgments: We thank the machine vision lab in Hebei University for the equipment and other help it offered.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Qin, X.F.; Zhang, Y.P.; Chang, H.; Lu, H.; Zhang, X.D. ACSiamRPN: Adaptive Context Sampling for Visual Object Tracking. *Electronics* **2020**, *9*, 1528. [\[CrossRef\]](#)
2. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
3. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8971–8980.
4. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 103–119.
5. Fan, H.; Ling, H. Siamese Cascaded Region Proposal Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7952–7961.
6. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2019; pp. 4282–4291.
7. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6668–6677.

8. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
9. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [[CrossRef](#)] [[PubMed](#)]
10. Fan, H.; Lin, L.T.; Yang, F.; Chu, P.; Deng, G.; Yu, S.J.; Bai, H.X.; Xu, Y.; Liao, C.Y.; Ling, H.B. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2019; pp. 5–6.
11. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In *European Conference on Computer Vision*; Springer: Amsterdam, The Netherlands, 2016; pp. 445–461.
12. Wu, Y.; Lim, J.; Yang, M.-H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [[CrossRef](#)] [[PubMed](#)]
13. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
14. Chen, X.; Yan, X.; Zheng, F.; Jiang, Y.; Xia, S.-T.; Zhao, Y.; Ji, R. One-shot adversarial attacks on visual tracking with dual attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10176–10185.
15. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable Siamese attention networks for visual object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6728–6737.
16. Voigtlaender, P.; Luiten, J.; Torr, P.H.S.; Leibe, B. Siam r-cnn: Visual tracking by re-detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6578–6588.
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 17 September 2016; pp. 21–37.
19. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Tian, Z.; Shen, C.; Chen, H.; He, Y. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9627–9636.
21. Zhou, X.Y.; Wang, D.Q.; Krhenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
23. 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 Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
24. 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](#)]
25. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. YouTubeBoundingBoxes. A large high-precision human-annotated dataset for object detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5296–5305.
26. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 472–488.
27. Wang, G.T.; Luo, C.; Xiong, Z.W.; Zeng, W.J. Spm-tracker: Series-parallel matching for real-time visual object tracking. *arXiv* **2019**, arXiv:1904.04452.
28. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
29. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [[CrossRef](#)] [[PubMed](#)]
30. 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.
31. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
32. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [[PubMed](#)]
33. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 188–203.
34. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; 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; pp. 6638–6646.

35. 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.
36. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
37. 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.
38. Gao, J.; Hu, W.; Lu, Y. Recursive Least-Squares Estimator-Aided Online Learning for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 7386–7395.
39. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6269–6277.