information

MDPI

Article

SiamSMN: Siamese Cross-Modality Fusion Network for Object Tracking

Shuo Han [1], Lisha Gao [1], Yue Wu [1], Tian Wei [1], Manyu Wang [2] and Xu Cheng [2,*]

[1] Nanjing Power Supply Branch, State Grid Jiangsu Electric Power Co., Ltd., Nanjing 210024, China; h_shuo0108@163.com (S.H.); sun_gls@163.com (L.G.); moonwu95@163.com (Y.W.); eioway@163.com (T.W.)

[2] School of Computer and Cyberspace Security, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202312200020@nuist.edu.cn

* Correspondence: xcheng@nuist.edu.cn

Abstract: The existing Siamese trackers have achieved increasingly successful results in visual object tracking. However, the interactive fusion among multi-layer similarity maps after cross-correlation has not been fully studied in previous Siamese network-based methods. To address this issue, we propose a novel Siamese network for visual object tracking, named SiamSMN, which consists of a feature extraction network, a multi-scale fusion module, and a prediction head. First, the feature extraction network is used to extract the features of the template image and the search image, which is calculated by a depth-wise cross-correlation operation to produce multiple similarity feature maps. Second, we propose an effective multi-scale fusion module that can extract global context information for object search and learn the interdependencies between multi-level similarity maps. In addition, to further improve tracking accuracy, we design a learnable prediction head module to generate a boundary point for each side based on the coarse bounding box, which can solve the problem of inconsistent classification and regression during the tracking. Extensive experiments on four public benchmarks demonstrate that the proposed tracker has a competitive performance among other state-of-the-art trackers.

Keywords: object tracking; Siamese network; weighted sum; concatenation operation

check for updates

Citation: Han, S.; Gao, L.; Wu, Y.; Wei, T.; Wang, M.; Cheng, X. SiamSMN: Siamese Cross-Modality Fusion Network for Object Tracking. *Information* **2024**, *15*, 418. https://doi.org/10.3390/info15070418

Academic Editor: Francesco Fontanella

Received: 11 June 2024
Revised: 5 July 2024
Accepted: 16 July 2024
Published: 19 July 2024

1. Introduction

Visual object tracking is one of the fundamental tasks in computer vision. It aims to track a given object in each frame over a video sequence. Object detection, which focuses on identifying and locating objects within individual frames, complements object tracking by providing initial object localization. Together, detection and tracking form a robust framework for many real-world applications. For instance, detection can identify and locate objects in the initial frame, and tracking can ensure continuous observation of these objects across subsequent frames [1]. Object tracking is widely used in many fields, such as visual surveillance [2], human–computer interaction [3], augmented reality [4], etc. Despite recent advances, it is still widely acknowledged as being an extremely difficult assignment because of background clutter, scale variations, significant variations in illumination, etc.

The currently used object tracking methods can be divided into two categories: correlation filter-based [5–12] and deep learning-based trackers [13–17]. In correlation filter-based tracking, a correlation filter is trained online on the region of interest by minimizing a least-squares loss. The object is detected in consecutive frames by convolving the trained filter via the Fast Fourier Transform (FFT) [18]. In order to estimate the object location in the next frame, the learned filter is applied to the region of interest in which the location of the maximum response is the target location. Early correlation filter-based trackers such as MOSSE [5] and CSK [6] exploited intensity features for object tracking. To achieve a more discriminating image representation, ValMadre et al. [7] proposed a correlation filter-based network (CFNET) in an offline manner that follows an end-to-end approach. Despite significant advancements, correlation filter-based trackers are less resistant to objects in fast-moving or low-frame-rate films and less flexible with respect to scale changes. In addition, further research suggests several targeted improvement techniques from a variety of angles, including scale improvement (e.g., DSST [8], CSR-DCF [9], etc.), elimination of boundary effects (SRDCF [10], C-COT [11], ECO [12], etc.), etc. These trackers have a clear advantage in real time, but still need to be optimized in situations such as complex background interference and similarity occlusion.

Deep learning technologies have significantly advanced the task of visual tracking by providing a powerful feature representation capacity. A variety of tracking methods based on deep learning have been presented, such as FCNT [13], MDNet [14], STCT [15], ADNet [16], SiamFC [17]. Among them, Siamese-based trackers have the potential advantages of significantly improving the tracking performance. Bertinetto et al. [17] first introduced a Siamese network for visual tracking. Since then, object trackers built on Siamese networks and object detection frameworks have achieved state-of-the-art performance, such as SiamRPN [19], SiamRPN++ [20], and SiamMask [21]. The Siamese-based trackers formulate the object tracking task as a similarity matching problem by computing cross-correlation similarities between a template image and a search image, which converts the tracking into finding the target object from an image region by computing the highest visual similarity. Therefore, it casts the tracking problem into a Region Proposal Network (RPN)-based detection framework by leveraging Siamese networks, which is the key to boosting the performance of deep trackers.

For most of the popular trackers (such as SiamFC [17], SiamRPN [19], and SiamBAN [22]), multi-level similarity maps can provide different representations. Similarity maps from shallow layers focus on low-level information, such as color and shape, which are essential for localization but lack semantic information; similarity maps from deeper layers have rich semantic information that is useful in some challenging scenarios, such as motion blur and huge deformation. Thus, the fusion of different similarity maps plays a critical role in accurate target tracking. Weighted sum and concatenation operation are common for aggregating multi-layer similarity maps. However, these methods can only combine different levels of similarity maps through a fixed linear approach, failing to fully utilize the complementary information from high-level and low-level similarity maps. This limitation restricts the tracker from achieving an interactive fusion of spatial information and semantic cues. Inspired by the transformer architecture [23], we design a novel multi-scale similarity-map fusion module that models the relationship between spatial information from high-resolution layers and semantic cues from low-resolution layers. The fusion module contains only one layer of feature encoder and feature decoder. The feature encoder aims to learn interdependencies between different similarity maps, while the decoder aggregates the low-level and high-level semantic information. The main problem in the tracking process is the inconsistency between classification and regression. Specifically, the classification probability is high but the positioning is inaccurate. In experiments, we observed that points near the object boundary were more likely to predict accurate locations. Motivated by this observation, we devised a learnable prediction module to refine the bounding boxes based on the predicted offset map. The proposed SiamSMN efficiently achieved robust and precise performance under complex scenarios, while maintaining good real-time processing capabilities. This is crucial for practical applications that require both speed and accuracy. The main contributions of this work are as follows:

- We design a transformer-based similarity-map fusion module that fully explores the interdependencies among multiple similarity maps associated with different semantic meanings, which helps the tracker accurately locate objects in complex scenarios.
- We propose a learnable prediction module to generate a boundary point for each side based on the rough bounding box, which can solve the problem of inconsistent classification and regression.
- Our methods achieve competitive performance with the state-of-the-art trackers on four different benchmarks, while maintaining real-time processing capabilities.

2. Related Work

2.1. Siamese Network-Based Object Tracking

Recently, a Siamese network-based tracking framework has attracted great attention in the vision tracking community due to its end-to-end training capacity and high efficiency. The Siamese tracker consists of two branches: a template branch and a search branch. The template branch receives the target image patch from the previous frame as input, while the search branch receives the target image patch in the current frame as input. Both of these branches share CNN parameters so that the two image patches encode the same transformation, which is suitable for tracking.

As one of the pioneering works, SiamFC [17] adopted a fully convolutional Siamese network as a feature extractor and introduces correlation layers to combine feature maps. Inspired by the success of SiamFC, more and more researchers began to pay attention to the Siamese Network tracking method. Zhu et al. [24] proposed a distractor-aware Siamese network (DaSiamRPN) that utilized the local-to-global search strategy to deal with the challenges of full occlusion and out-of-view. Wang et al. [25] put forward a residual attentional Siamese network (RASNet), which embedded an attention mechanism into Siamese trackers to promote the discriminating ability of the tracking model. Other methods include SiamDW [26], SiamMASK [21], SiamFC++ [27], etc. Though the above methods utilize a multi-scale strategy to cope with scale variation, they cannot handle aspect ratio changes due to target appearance variations. In order to make more accurate predictions for target locations, SiamRPN [19] combines a Region Proposal Network (RPN) in the object detection with a Siamese network. By jointly training a classification branch and a regression branch for the region proposal, SiamRPN [19] avoids the time-consuming step of extracting multi-scale feature maps for the object scale invariance and achieves very efficient results. However, it has difficulty dealing with distractions with a similar appearance to the object. Based on SiamRPN [19], DaSiamRPN [24] increases the hard-negative training data during the training phase. Through data enhancement, it improves the discrimination of the tracker and obtains a much more robust result. SiamRPN++ [20] optimizes the network architecture by using ResNet [28] as a backbone. At the same time, it randomly shifts the training object location in the search region during model training to eliminate the center bias. Despite these advancements, existing methods still face challenges in effectively fusing multi-layer similarity maps to fully exploit spatial and semantic information. Our proposed method, SiamSMN, aims to address these shortcomings by introducing a novel multi-scale fusion module and a learnable prediction head, thereby enhancing tracking performance.

2.2. Transformer in Object Tracking

The Vision Transformer (ViT) [29] first presented a pure vision transformer architecture, obtaining an impressive performance on image classification. Briefly, a transformer is an architecture for transforming one sequence into another with the help of attention-based encoders and decoders. The attention mechanism observes an input sequence and decides at each step which other parts of the sequence are important, facilitating the capture of global information from the input sequence. In recent years, some studies attempted to introduce the transformer to object tracking and achieved promising performance. Yu et al. [30] proposed a deformable Siamese attention network, referred to as SiamAttn, by introducing a new Siamese attention mechanism that computed deformable self-attention to improve the discriminating ability of target features before applying depth-wise cross-correlation. CGACD [31] learns attention from the correlation result between the template and search region and then adopts the learned attention to enhance the search region features for further classification and regression. TransT [32] is a transformer-based fusion network for target-search information incorporation. Although these works have improved tracking accuracy with the attention mechanism, they still heavily rely on the correlation operation in fusing the template and search region feature. In this work, we exploit a transformer to directly fuse multi-layer similarity maps without using any weighted sum or concatenation operations.

3. Proposed Method

In this section, we present a detailed description of the proposed SiamSMN framework. As shown in Figure 1, our SiamSMN consists of three components: a feature extraction network, a multi-scale fusion module, and a prediction head. First, the feature extraction network separately extracts the features of the template image and the search image. Second, these features are calculated by a depth-wise cross-correlation to produce multiple similarity maps. Then, these different scales of similarity maps are aggregated by the proposed feature fusion network. Finally, the fused feature maps are input into the prediction head, which is responsible for classifying the enhanced features and regressing the bounding boxes to generate the final tracking results.

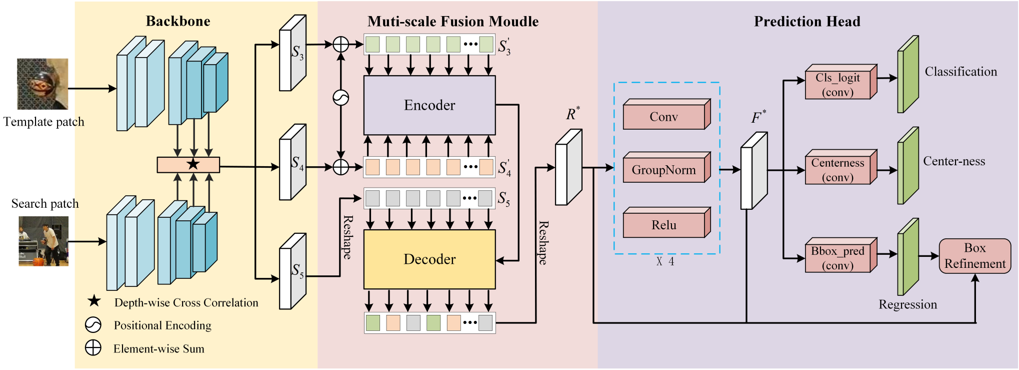

Figure 1. Framework of SiamSMN, which contains a feature extraction network, a multi-scale fusion module, and a prediction head.

3.1. Feature Extraction Network

Like Siamese-based trackers, the proposed SiamSMN method takes a pair of image patches as the inputs of the backbone network. The Siamese backbone network consists of two identical branches. One is called the template branch, which receives the template patch as input (denoted as $\boldsymbol{Z}$). The other is the search branch, which receives the search patch as input (denoted as $\boldsymbol{X}$). The two branches share parameters to embed the inputs $\boldsymbol{Z}$ and $\boldsymbol{X}$ into a common feature space for cross-correlation. The cross-correlation between template and search regions is implemented in the common feature embedding space as follows:

$$S = \phi(\boldsymbol{X}) \star \phi(\boldsymbol{Z}), \tag{1}$$

where $\star$ denotes the channel-by-channel correlation operation. The generated similarity map S has the same number of channels as $\phi(\boldsymbol{X})$, and it contains massive information for classification and regression.

Object tracking requires rich representations that span levels from low to high, scales from small to large, and resolutions from fine to coarse. Many methods take advantage of fusing both low-level and high-level features to improve tracking accuracy. In our network, multi-layer features are extracted to collaboratively infer the target location. We utilize the ResNet-50 as our backbone network and use blocks 3, 4, and 5 of the ResNet-50 to extract features from the target template and the search region. Features from different blocks of the backbone focus on different hierarchical information about objects. We use a depth-wise cross-correlation to aggregate the features extracted from the last three residual blocks of the backbone, which helps the trackers produce multiple semantic similarity maps. The cross-correlation between the template feature and search feature is implemented as follows:

$$S_{\mathrm{i}} = \phi_i(\boldsymbol{X}) \star \phi_i(\boldsymbol{Z}), i = 3, 4, 5, \tag{2}$$

where $\star$ represents the cross-correlation operator, and $\phi(\cdot)$ is the embedding function for feature extraction. As shown in Figure 1, $S_3^{'}$, $S_4^{'}$, and S_5 are fed into the feature fusion module individually to aggregation multi-layer similarity maps.

3.2. Multi-Scale Fusion Module

Inspired by the transformer [23], we fuse different levels of similarity maps by designing a novel transformer fusion network. Unlike the original transformer [23], our transformer fusion model only contains one layer for both feature encoder and decoder. A feature encoder aims to learn the interdependencies among different similarity maps, while the feature decoder aggregates the low-level and high-level semantic information.

Feature encoder: As shown in Figure 2, first, a learnable position encoding is used to encode the similarity maps from the 3th and 4th layers, denoted as $S_3^{'}$ and $S_4^{'}$. Then, we perform an addition and normalization operations on $S_3^{'}$ and $S_4^{'}$, and the result is used as the $\boldsymbol{K}$ and $\boldsymbol{Q}$ inputs of the multi-head attention module. $S_3^{'}$ serves as its $\boldsymbol{V}$ input. The multi-head attention output of this feature encoder can be obtained by:

$$\begin{aligned} \boldsymbol{M}_E^1 = MultiHead\Big(&\text{Norm}\Big(S_3^{'} + S_4^{'}\Big), \\ &Norm\Big(S_3^{'} + S_4^{'}\Big), S_3^{'}\Big). \end{aligned} \tag{3}$$

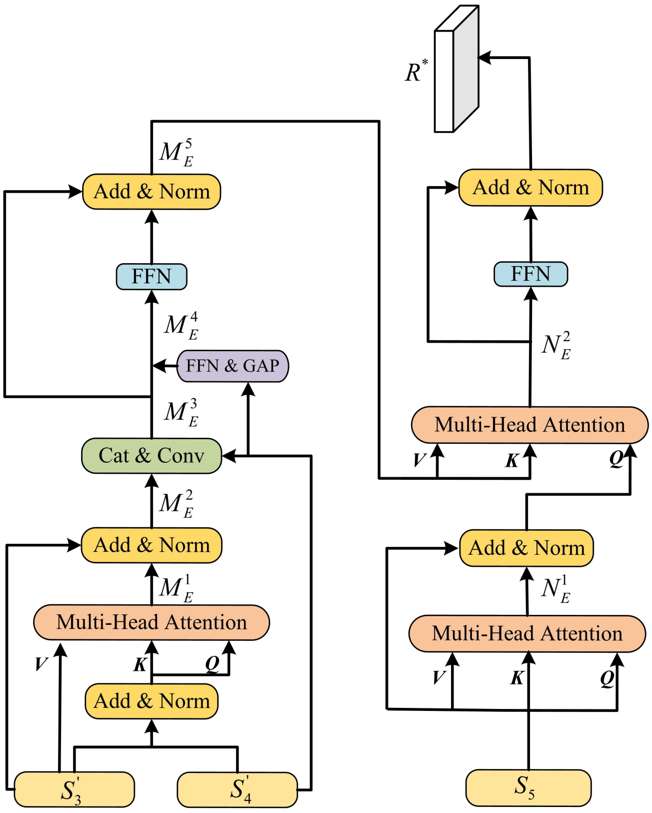

Figure 2. Detailed workflow of the multi-scale fusion module. The left sub-window illustrates the feature encoder. The right one shows the structure of the decoder.

Eventually, the encoded information can be calculated through FFT and normalization. The output of the encoder can be used by the decoder as its input for the multi-head attention module.

Feature decoder: The feature decoder follows the same structure as the encoder. Differently, we built the effectively feature decoder without positional encoding and a global average pooling. In addition, the feature decoder has two heads of attention.

Specifically, the output of the first multi-head attention can be expressed as:

$$\boldsymbol{N}_E^1 = \text{Multi}Head(S_5, S_5, S_5). \tag{4}$$

In order to further increase the tracking accuracy, the second multi-head attention aggregates the semantic information from the low-layer similarity map. We can obtain a fusion result from the following equation:

$$\boldsymbol{N}_E^2 = MultiHead\left(\boldsymbol{N}\text{orm}\left(\boldsymbol{N}_E^1 + \boldsymbol{S}_5\right), \boldsymbol{M}_E^5, \boldsymbol{M}_E^5\right). \tag{5}$$

The final response map (R^*) can be calculated through FFT and normalization.

3.3. Prediction Head

As shown in Figure 1, the classification branch, regression branch, and centerness branch are applied to localize objects and estimate their shapes. When testing, the final score (used for ranking the detected bounding boxes) is computed by multiplying the predicted centerness with the corresponding classification score, which helps suppress the low-quality detected bounding boxes and improves the overall performance by a large margin. For a response map (R^*) obtained using a multi-scale similarity-map fusion network, the classification branch outputs a classification feature map $A^{cls}_{w\times h\times 2}$, the regression branch outputs a predicted offset map $A^{reg}_{w\times h\times 4}$, the centerness branch outputs a centerness feature map $A^{cen}_{w\times h\times 1}$, where each point value gives the "centerness score" of the corresponding location. Each point (i, j) in $A^{cls}_{w\times h\times 2}$ contains a 2D vector, which represents the foreground and background scores of the corresponding location in the search region. Similarly, each point (i, j) in $A^{reg}_{w\times h\times 4}$ contains a 4D vector $t(i, j) = (l, t, r, b)$, which represents the distance from the corresponding location to the four sides of the bounding box in the input search region.

In the experiments, we observed that points near the object boundary were more likely to predict accurate locations. Inspired by this, we propose a box refinement module to refine the bounding boxes based on the regression branch; as shown in Figure 3, the feature map (F^*), regression (Reg), and response map (R^*) are the three inputs for this box refinement. First, the feature map obtains a set of offsets after a set of convolution operations, denoted as $\boldsymbol{T}_\text{t}$. Next, we perform a reshape function on regression, and the result is recorded as $\boldsymbol{Z}_0$. $\boldsymbol{Z}_0$ represents the distance from the given point to the 4 boundaries. Then, we obtain $\boldsymbol{T}_0$ by Equation (8):

$$\boldsymbol{T}_0 = \theta_c(R^*), \tag{6}$$

where $\theta_\text{c}(\cdot)$ maps the points on the response map R^* back to the search patch and obtains the generated points on each layer. After that, we utilize $\boldsymbol{Z}_0$ and $\boldsymbol{T}_0$ as the input of $\phi_\text{c}(\cdot)$ to obtain a coarse bounding box. This $\boldsymbol{B}_\text{b}$ is defined as follows:

$$\boldsymbol{B}_b = \phi_\text{c}(\boldsymbol{Z}_0, \boldsymbol{T}_0), \tag{7}$$

where $\phi_\text{c}(\cdot)$ decodes the distance prediction into a bounding box. Finally, the coarse bounding box generates a boundary point for each side based on a set of offsets generated by the feature map. A finer bounding box is generated by aggregating the prediction results of the four boundary points.

Loss function: The training loss function in this paper is defined as follows:

$$L = L_{cls} + \lambda_1 L_{cen} + \lambda_2 L_{reg}, \tag{8}$$

where L_{cls} represents the focal loss for classification, L_cen refers to the IoU loss, and L_reg is the binary cross-entropy loss. λ_1 and λ_2 are the weight parameters of L_cen and L_reg, respectively. During model training, we empirically set λ_1 = 1 and λ_2 = 3.

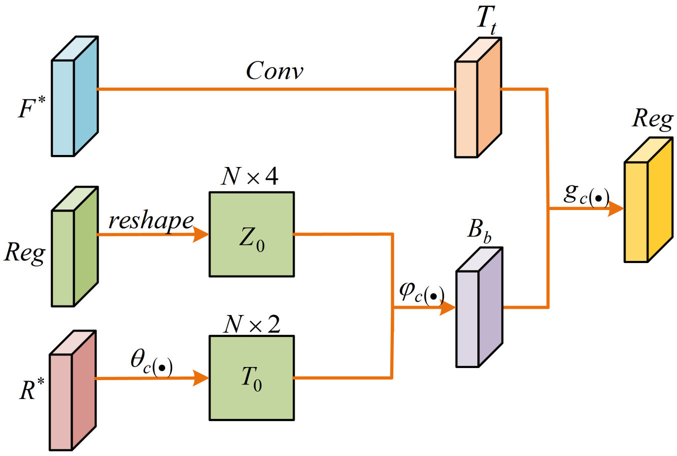

Figure 3. Illustration of the box refinement.

4. Experiments

4.1. Implementation Details

Our entire network was trained end-to-end on large-scale datasets. The training sets included COCO [33], ImageNet VID [28], ImageNet DET [28], and LaSOT [34].

For the video datasets, we directly sampled the image pairs from one video sequence to collect training samples. For the COCO detection datasets, we applied some transformations to the original images to generate pairs. Common data augmentation techniques were applied to enlarge the training set. For easy comparison, the input sizes of the search patch and template regions were 255 × 255 and 127 × 127, respectively. The backbone parameter was initialized on ImageNet and then we used the parameter as initialization to retrain our model.

Training details: In total, there were 20 epochs; for the first 10 ones, the parameters of the Siamese sub-network were frozen while training the classification and regression sub-networks. For the last 10 epochs, the last three blocks of ResNet-50 were unfrozen to be trained together. In addition, the stochastic gradient descent (SGD) was adopted, and batch size, momentum, and weight decay were set to 32, 0.9, and 0.0001, respectively. Our tracker was trained in Python using PyTorch on a PC with a RTX 2080 Ti. Our approach was trained with only the specified training set provided by the official website for fair comparison.

Testing details: During the testing process, we used an offline tracking strategy. Only the object in the initial frame of a sequence was adopted as the template. Consequently, the target branch of the Siamese sub-network could be pre-computed and fixed during the whole tracking period. The search region in the current frame was adopted as the input of the search branch.

4.2. Comparison with State-of-the-Art Trackers

We compared our approach with the state-of-the-art trackers on four tracking datasets.

4.2.1. OTB100

The OTB100 [35] dataset is a public tracking benchmark that contains 100 sequences from different scenes. All frames in the sequence are divided into seven categories: camera motion, illumination change, occlusion, size change, motion change, unassigned, and overall. The shortest sequence “Deer” in OTB100 has 71 frames, and the longest sequence “Doll” is 3872 frames. The average length of each sequence in this benchmark is about 590 frames. We followed the one pass evaluation (OPE) protocol and report the AUC scores of the success plot.

As shown in Figure 4, we compared our tracker with some recent top-performing trackers, including SiamAttn [30], SiamBAN [22], SiamRPN++ [20], ECO [12], Transforming Tracking [32], SiamFC++ [27], Ocean [36], SiamDW [26], and DaSiamRPN [24]. Our tracker achieved the best performance. Compared with the recent SiamBAN [22], our SiamSMN improved by 1.6% in success and 1.5% in precision.

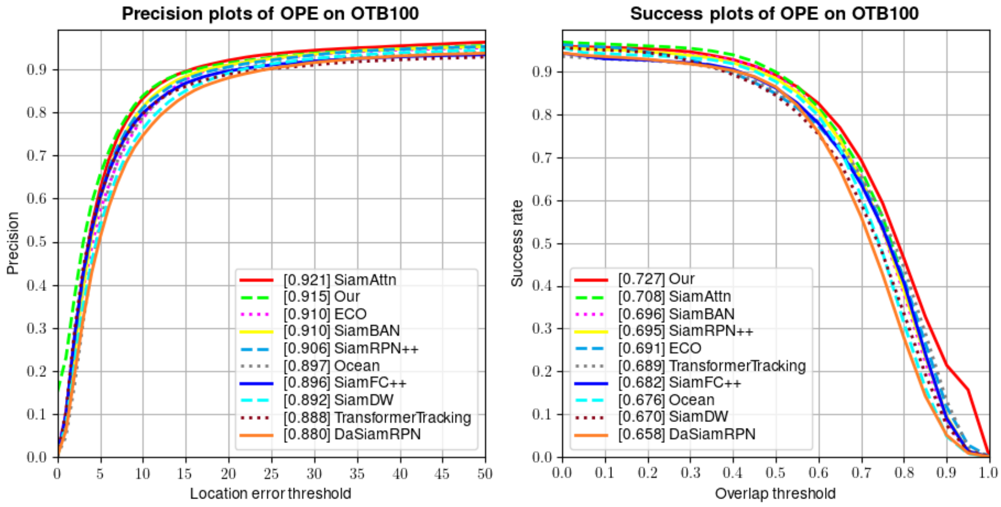

Figure 4. Comparison among the top-10 trackers on OTB100.

4.2.2. UAV123

The UAV123 [37] dataset contains a total of 123 video sequences, including more than 110K frames. All sequences are fully annotated with upright bounding boxes. The objects in the dataset mainly suffer from fast motion, large scale variation, large illumination variation, and occlusions, which make tracking challenging.

We compared our trackers with other nine state-of-the-art real-time trackers, including SiamAttn [30], SiamGAT [38], Ocean [36], CGACD [31], SiamCAR [39], SiamRPN++ [20], SiamBAN [22], SiamRPN [19], and SiamDW [26]. Figure 5 shows the success and precision plots. Our tracker outperformed all other trackers. Compared with the state-of-the-art SiamAttn [30], SiamSMN obtained competitive results with a much simpler network and without heuristic tuning parameters.

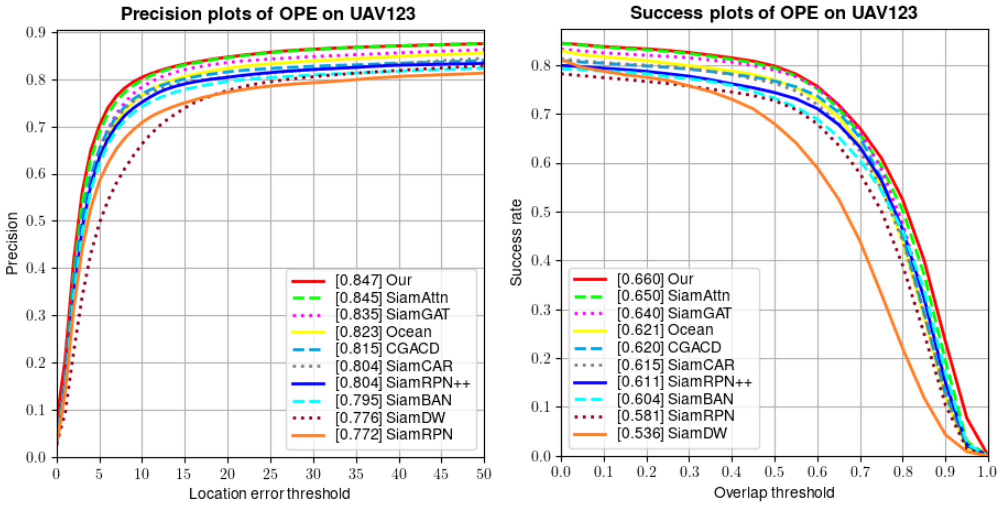

Figure 5. Comparisons on UAV123. Our SiamSMN achieves the best results.

4.2.3. LaSOT

To further validate the proposed framework on a larger and more challenging dataset, we conducted experiments on LaSOT [34]. The LaSOT dataset provides large-scale, high-quality dense annotations with 1400 videos in total and 280 videos in the testing set. Such a large test dataset brings a great challenge to the tracking algorithms. The official website of

LaSOT provides 35 algorithms as baselines. Normalized precision plots, precision plots, and success plots in one-pass evaluation (OPE) were considered as the indicators.

We compared our SiamSMN with the top-nine trackers including SiamBAN [22], ATOM [40], SiamRPN++ [21], SiamMask [3], and so on. The results of SiamBAN [22] are provided on the website of its authors, while other results are provided by the official website of LaSOT. Figure 6 reports the overall performances of our SiamSMN tracker on the LaSOT testing set. SiamSMN increased the AUC and the normalized distance precision relatively by 1.6% and 1.4% over SiamBAN [22], which is the best tracker reported in the original paper.

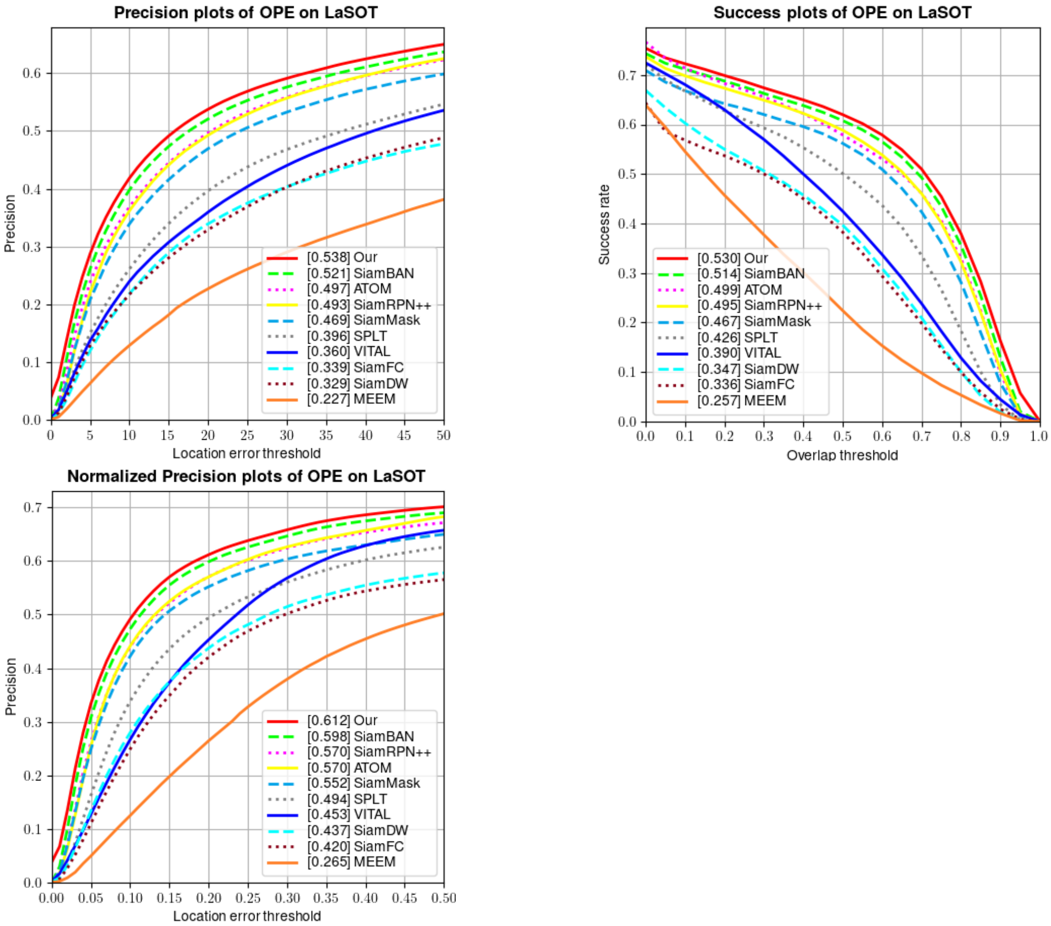

Figure 6. Comparisons among the top-10 trackers on LaSOT. Our SiamSMN significantly outperforms the state-of-the-art methods.

4.2.4. GOT-10K

GOT-10K [41] is a recent large-scale dataset that contains 10K sequences for training and 180 for testing. After uploading the tracking results, the analysis is performed automatically by the official website.

The provided evaluation indicators include success plots, average overlap (AO), and success rate (SR). All the results are provided by the official website of the GOT-10K. Table 1 shows that SiamSMN can outperform all the trackers on the GOT-10K. As shown in Table 1, our tracker ranked first in terms of all the indicators. Compared with Ocean [36], our SiamSMN improved the scores by 2.1%, 3.7% and 5.0%, relatively, for AO, $SR_{0.5}$, and $SR_{0.75}$.

Table 1. Comparison results on the GOT-10K test set. The best two results are highlighted in red and blue fonts, respectively.

Method	**SiamFC** [17]	**ECO** [12]	**ATOM** [40]	**SiamRPN++** [20]	**SiamFC++** [27]	**PrDiMP** [42]	**SiamCAR** [39]	**D3s** [43]	**DCFST** [44]	**Ocean** [36]	**SiamSMN** ours
AO	34.8	31.6	55.6	51.7	59.5	63.4	56.9	59.7	**63.8**	61.1	**63.2**
$SR_{0.5}$	35.3	30.9	63.4	61.6	69.5	73.8	67.0	67.6	**75.3**	72.1	**75.8**
$SR_{0.75}$	9.8	11.1	40.2	32.5	47.9	**54.3**	41.5	46.3	49.8	47.3	**52.3**

4.3. Ablation Study

To analyze and verify the effectiveness of each proposed module, an ablation experiment was performed on the UAV123 [37] dataset.

4.3.1. Box Refinement

To verify the effectiveness of the box refinement (BR), an ablation experiment was performed, and the results are shown in Table 2. Without box refinement, our method reached 63.6% and 82.5%. When we added the box refinement, the success and precision improved by 2.4% and 2.2%, respectively. The outcome in Table 2 demonstrates that the box refinement can consistently improve tracking performance.

Table 2. The ablation study results of the box refinement (BR). The best results are highlighted in red.

MFM	**Weighted Sum**	**Concatenation**	**BR**	**UAV123**	
				Suc	**Pre**
✓			✓	**0.660**	**0.847**
✓			✗	0.636	0.825
	✓		✓	0.626	0.802
	✓		✗	0.604	0.795
		✓	✓	0.647	0.827
		✓	✗	0.615	0.804

4.3.2. Multi-Scale Fusion Module

To analyze the effectiveness of the multi-scale fusion module (MFM), we designed three variants: weighted sum, concatenation operation, and MFM. As shown in Table 3, it was found that the use of the MFM yielded better results than the other two methods. When we used the multi-scale fusion module to fuse multi-layer features, it was obvious that our method showed a great improvement in tracking performance compared to the traditional fusion methods (weighted sum and tandem).

Table 3. The ablation study results of the multi-scale feature fusion. The best results are highlighted in red.

MFM	**Weighted Sum**	**Concatenation**	**UAV123**	
			Suc	**Pre**
✓			**0.636**	**0.825**
	✓		0.604	0.795
		✓	0.615	0.804

4.4. Speed Analysis

In Table 4, we show the evaluation of OTB100 with respect to frames per second (FPS). The reported speed was evaluated on a machine with one RTX 2080 Ti, and those of other methods are provided by the OTB100 official results. As shown in the table, although TransT [32] was faster than our method, the accuracy was 3.8% lower than that of our method. In addition, our network was much simpler than others, and no specially designed parameters were needed for training.

Table 4. The results in terms of success and speed for different methods on OTB100. The best results are highlighted in red.

	SiamBAN [22]	SiamRPN [20]	TransT [32]	SiamATL [45]	Ours
Success	0.696	0.640	0.689	0.655	**0.727**
Speed (FPS)	40.00	34.17	**50.00**	21.30	42.00

5. Conclusions

In our paper, we exploited the expressive power of the transformer and proposed a simple yet effective visual-tracking framework named SiamSMN that fully explores the interdependencies among multi-level similarity maps. SiamSMN directly classifies objects and regresses bounding boxes in a unified network and does not require pre-defined candidate boxes. Experimentation results demonstrated that the proposed SiamSMN method could achieve competitive performance and real-time speed on four popular tracking benchmark datasets, confirming its effectiveness and efficiency.

Author Contributions: Conceptualization, S.H., L.G., Y.W. and X.C.; Methodology, S.H., L.G., Y.W., T.W. and X.C.; Software, S.H., L.G., Y.W., T.W. and X.C.; Validation, S.H., Y.W. and X.C.; Formal analysis, Y.W., T.W. and X.C.; Investigation, Y.W., T.W. and X.C.; Resources, S.H. and L.G.; Data curation, Y.W., M.W. and X.C.; Writing—original draft, S.H., L.G., Y.W., T.W. and X.C.; Writing—review & editing, S.H., L.G., T.W. and M.W.; Visualization, X.C.; Supervision, S.H., L.G. and Y.W.; Project administration, S.H. and X.C.; Funding acquisition, S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Research on Cloud Edge Collaborative Intelligent Image Analysis Technology Based on Artificial Intelligence Platform (No. J2023003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in the following repositories: OTB100: The data are available in the OTB100 repository, reference number [35]. UAV123: The data are available in the UAV123 repository, reference number [37]. LaSOT: The data are available in the LaSOT repository, reference number [33]. GOT-10K: The data are available in the GOT-10K repository, reference number [41].

Conflicts of Interest: This research is supported by the company Nanjing Power Supply Branch, State Grid Jiangsu Electric Power Co., Ltd. and may lead to the development of products that may have been licensed by Nanjing University of Information Science and Technology in which I have the company Nanjing Power Supply Branch, State Grid Jiangsu Electric Power Co., Ltd. business and/or financial interest. I have fully disclosed these interests and have developed an approved plan to manage any potential conflicts that may arise from such an arrangement.

References

1. Reddy, K.R.; Priya, K.H.; Neelima, N. Object Detection and Tracking—A Survey. In Proceedings of the 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 12–14 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 418–421.
2. Xing, J.; Ai, H.; Lao, S. Multiple human tracking based on multi-view upper-body detection and discriminative learning. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1698–1701.
3. Liu, L.; Xing, J.; Ai, H.; Ruan, X. Hand posture recognition using finger geometric feature. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 565–568.
4. Zhang, G.; Vela, P.A. Good features to track for visual slam. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1373–1382.
5. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 2544–2550.

6. Henriques João, F.; Rui, C.; Pedro, M.; Jorge, B. Exploiting the circulant structure of tracking-by-detection with kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 702–715.
7. 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.
8. 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; Bmva Press: Durham, UK, 2014.
9. LuNežič, A.; Vojíř, T.; Zajc, L.Č.; Matas, J.; Kristan, M. Discriminative correlation filter tracner with channel and spatial reliability. *Int. J. Comput. Vis.* **2018**, *126*, 671–688. [CrossRef]
10. 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, Boston, MA, USA, 7–12 June 2015; pp. 4310–4318.
11. Danelljan, M.; Robinson, A.; Shahbaz Khan, F.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part V 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 472–488.
12. 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.
13. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3119–3127.
14. 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, 26 June–1 July 2016; pp. 4293–4302.
15. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Stct: Sequentially training convolutional networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1373–1381.
16. Tian, C.; Xu, Y.; Li, Z.; Zuo, W.; Fei, L.; Liu, H. Attention-guided CNN for image denoising. *Neural Netw.* **2020**, *124*, 117–129. [CrossRef] [PubMed]
17. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; Proceedings, Part II 14; Springer, 2016; pp. 850–865.
18. Brigham, E.O.; Morrow, R. The fast Fourier transform. *IEEE Spectr.* **1967**, *4*, 63–70. [CrossRef]
19. 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.
20. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J.S. Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 16–20.
21. 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/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338.
22. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6668–6677.
23. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS 30), Long Beach, CA, USA, 4–9 December 2017.
24. 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. 101–117.
25. 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.
26. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
27. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12549–12556.
28. 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]
29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
30. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6728–6737.
31. Du, F.; Liu, P.; Zhao, W.; Tang, X. Correlation-guided attention for corner detection based visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6836–6845.

32. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 8126–8135.
33. 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 Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
34. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5374–5383.
35. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–27 June 2013; pp. 2411–2418.
36. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XXI 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 771–787.
37. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
38. Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; Shen, C. Graph attention tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 9543–9552.
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/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6269–6277.
40. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Atom: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4660–4669.
41. 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]
42. Danelljan, M.; Gool, L.V.; Timofte, R. Probabilistic regression for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7183–7192.
43. Lukezic, A.; Matas, J.; Kristan, M. D3s-a discriminative single shot segmentation tracker. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 7133–7142.
44. Zheng, L.; Tang, M.; Chen, Y.; Wang, J.; Lu, H. Learning feature embeddings for discriminant model based tracking. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XV 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 759–775.
45. Huang, B.; Xu, T.; Shen, Z.; Jiang, S.; Zhao, B.; Bian, Z. SiamATL: Online update of siamese tracking network via attentional transfer learning. *IEEE Trans. Cybern.* **2021**, *52*, 7527–7540. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.