electronics

MDPI

Article

Siamese High-Level Feature Refine Network for Visual Object Tracking

Md. Maklachur Rahman [1], Md Rishad Ahmed [2,3], Lamyanba Laishram [1], Seock Ho Kim [1] and Soon Ki Jung [1,*]

[1] School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; maklachur@knu.ac.kr (M.M.R.); yanbalaishram@knu.ac.kr (L.L.); tjrgh8357@knu.ac.kr (S.H.K.)
[2] Brainnetome Center & National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China; 2157971471@mails.ucas.ac.cn
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: skjung@knu.ac.kr

Received: 30 September 2020; Accepted: 11 November 2020; Published: 14 November 2020

Abstract: Siamese network-based trackers are broadly applied to solve visual tracking problems due to its balanced performance in terms of speed and accuracy. Tracking desired objects in challenging scenarios is still one of the fundamental concerns during visual tracking. This research paper proposes a feature refined end-to-end tracking framework with real-time tracking speed and considerable performance. The feature refine network has been incorporated to enhance the target feature representation power, utilizing high-level semantic information. Besides, it allows the network to capture the salient information to locate the target and learns to represent the target feature in a more generalized way advancing the overall tracking performance, particularly in the challenging sequences. But, only the feature refine module is unable to handle such challenges because of its less discriminative ability. To overcome this difficulty, we employ an attention module inside the feature refine network that strengths the tracker discrimination ability between the target and background. Furthermore, we conduct extensive experiments to ensure the proposed tracker's effectiveness using several popular tracking benchmarks, demonstrating that our proposed model achieves state-of-the-art performance over other trackers.

Keywords: siamese network; visual object tracking; feature refine network; attention mechanism

1. Introduction

Visual object tracking (VOT) is widely used to estimate the trajectory of objects in a video sequence and one of the fundamental research areas in computer vision having applications in intelligent surveillance [1], video understanding to analyze human motion [2], robotics [3], and autonomous vehicle driving [4]. The applications of VOT usually depend on the reliable prediction of the target location. The tracker locates objects by exploiting an initial target from a video frame to estimate the new target positions in the subsequent frames. However, VOT is particularly a challenging problem because of the limitation of desired object information supplied in the initial frames from video sequences. Therefore, high-performance VOT approaches with satisfactory efficiency and accuracy are necessary to overcome real-world challenges such as deformation and illumination changes, scale variations, occlusion, background clutter, pose changes, fast motion, and motion blur. Recently, numerous deep learning methods have been employed to solve the VOT challenges.

Tracking by detection and template matching are two major tracking techniques in deep learning-based approaches [5]. The tracking by detection considers tracking as a classification problem. Building a competent classifier is one of the key concerns to build tracking by detection based trackers.

The classifier network is used to learn to discern the desired object from the background, and it is usually updated by utilizing the previous target information to infer the new location in the current frame. With the advent of the correlation filters (CF), some of the detection based trackers [6,7] includes CF with the convolution neural network feature map, and others use domain-based target classifier to locate the object such as MDNet [8]. However, these kinds of trackers use an online updating strategy during tracking, which is enormously time-consuming, and it is prone to be over-fitted on the historical frames.

On the other hand, template matching considers tracking as a similarity learning problem. The template patch is extracted from a video frame as a target object, and it uses to match the most analogous patch in the current frame. Similarity learning is exploited to train the Siamese network, which is first introduced for signature verification [9]. Later, it is gained popularity to the tracking community after introducing the SiamFC [10] tracker. Siamese network is two parallel CNN based Y-shaped deep learning framework where one branch utilizes for template image and another responsible for subsequent frames of the video. It uses cross-correlation to match the template patch with the current frame of the video, which utilizes fewer operations and increases the network computational power. Due to its efficient computational ability, several trackers [11–15] are developed recently to solve the tracking problems using the Siamese network. However, they suffer a lack of robustness to handle challenging sequences, particularly background clutter, occlusion, and appearance changes because of offline training on the large dataset without considering the most discriminative features that reduce the performances. To improve the Siamese network tracking performance, some of the trackers integrated online target updating strategies inside the Siamese architecture during tracking. However, the updating technique enables a pitfall of compromising tracking speed that creates instability of trackers performing in real-time applications.

To overcome the challenges without compromising the real-time tracking facility, we have introduced a feature refine Siamese network (SiamFRN). In SiamFRN, the feature refine mechanism has been incorporated as an extension of the baseline Siamese architecture to improve the underlying backbone feature representation power. It enables the overall network feature gains for learning better discriminative ability. In our feature refined mechanism, we have adopted high-leveled fully convolutional features to capture the salient information of the target object, which shows less discernibility with background and other similar objects. Therefore, we employ an attention module inside the feature refine mechanism to enhance the learning network to be more discriminative. To preserve high tracking speed with accuracy, we only exploit feature refine network in the stationary branch during testing. In summary, the main contributions of the proposed work are the following:

- We introduce an effective feature refine network with an end-to-end learning facility to enhance the target feature representation ability that enables us to capture the salient information location of the target.
- We employ an attention module within a residual feature refine block using identity mapping to augment the overall network discriminative power.
- Extensive experiments has been performed that demonstrates excellent performance over state-of-the-art trackers on several popular benchmarks including (OTB100 [16,17], OTB50 [16,17], UAV123 [18], TC128 [19], VOT2017 [20], and VOT2018 [21]) with 60 frames per second (fps) tracking speed. The codes and results will be available at https://github.com/maklachur/SiamFRN.

2. Related Work

In this section, we have summarized the most relevant research literature on visual object tracking (VOT) covering tracking with convolution neural network, tracking with the siamese network, and tracking with visual attention. The detailed overview of all trackers is beyond the scope of our work. However, we included these survey studies [22,23] for the interested readers to gather more information about tracking frameworks overview.

2.1. Tracking with Convolution Neural Network (CNN)

Deep learning algorithm such as CNN has achieved great success in the field of computer vision, for instance, image classification [24,25], object detection [26], and semantic segmentation [27]. Additionally, work [28] proposed a traffic light detector using a hybrid strategy designed by a support vector machine, and a convolutional neural network (CNN) with the AlexNet structure that acts as the self-learned detector. CNN has also been performed well in visual object tracking, but not enough research is done so far. The main critical challenge behind this belongs to the unavailability of data sources or data size, which is considered as the power of CNN. Besides, updating the CNN model is also another challenge comparing with the conventional tracking approaches [29,30]. The traditional tracking methodologies generally used the handcrafted features preprocessed by the human experience of the tracking object, which has the generalization limitation over CNN features [31]. As in [32,33], former work adopted the pixel values in correlation filter-based tracking, while the later work applied principal component analysis (PCA) for frames processing respectively for VOT.

In [34], Ma et al. considered the convolutional features by replacing the handcrafted features for correlation filter-based tracking. During offline training of the CNN model, most of the research work analyzes the first frame of a sequence and some of them also considered later frames in tracking however in both cases still there is the data and quality problem of tracking performance. Doulamis et al. performed network adaptation to update the algorithm with minimum degradation, which is the aggregation of tracking problems in neural networks [35]. In [36], Wang et al. analyzed the semantic features of the tracking object by fusing the convolutional layers. However, these trackers utilized good feature representation, but it is challenging to train them offline on the large datasets. Thus, the online strategy of these trackers dramatically reduces the performance, especially, the tracking frame rates.

2.2. Tracking with Siamese Network

The Siamese network popularity increases recently in visual object tracking due to the power of similarity matching such as patch comparing [37] and stereo matching [38]. Because of its satisfactory efficiency in similarity matching, works [10,39,40] were introduced Siamese based network into the object tracking field to calculate the similarity score to generate more effective candidate–template sets. Guo et al. [11] proposed DSiam to extract the background information and implemented online training to reduce the target appearance discrepancies. Tao et al. in work [40], to match with template feature, they carefully chosen the candidates from the nearby regions of interest of the previously located region.

However, Bertinetto et al. utilized a sliding window search to learn the similarity between target and search images to design SiamFC tracker [10], which is one of the popular and pioneering works. But it shows less discriminative ability and easily troubled on the challenging sequences. To improve the SiamFC network, numerous follow-up works have been developed. Using a closed-form equation, CFnet [13] employed a correlation filter in the template branch to enhance the target feature map. SiamTri [12] introduced triplet loss to progress SiamFC tracker performance. MemTrack [14] utilized a memory mechanism to handle challenging sequences. To augment the generalization ability of the tracker, IRCA-Siam [41] added several noises [42,43] to the input images during training. In contrast, we have employed a feature refines mechanism in the semantic branch of the baseline siamese network to boost the tracker performance by enhancing target feature representation power.

2.3. Tracking with Visual Attention

Visual attention to the objective image can be helpful to emphasize the vital parts by decreasing the effect of the background, which is a beneficial strategy with changes in ambient behaviors, objects nature, and illumination. The popularity of the visual attention mechanism is increasing in many computer vision applications including image captioning and classification [44,45], semantic segmentation [46], and activity recognition [47].

In [48], Cui at el. introduced a multi-directional recurrent neural network to engender the saliency maps gaining target attention. In [49,50], ACFN, and SCT attention methods have been improved respectively by incorporating the correlation filters for object tracking. He et al. [51] included a channel attention module in calculating the channel-wise weights, but they considered only the max-pooled feature. MemTrack [14] used a long short term memory (LSTM) attention controller for updating features into the memory. IMG-Siam [52] utilized channel attention to integrating the foreground target mask for enhancing target structural information. In this work, we have incorporated the attention module to improve the discriminative ability of the feature refine mechanism to boost the overall tracking performance on the challenging sequences where we have utilized both max-pooled and average-pooled features together. The max-pooled operation considered the distinctive target features, whereas average-pooled used to get the general idea about the feature map.

3. Proposed Method

The overview of the proposed tracking pipeline is shown in Figure 1. We have introduced a fully convolutional siamese network to extract the feature map from the input, and a feature refine network to improve the feature representation power that helps to learn better discrimination ability between the foreground object and the background information. The following sections describe the key components of the SiamFRN network in details.

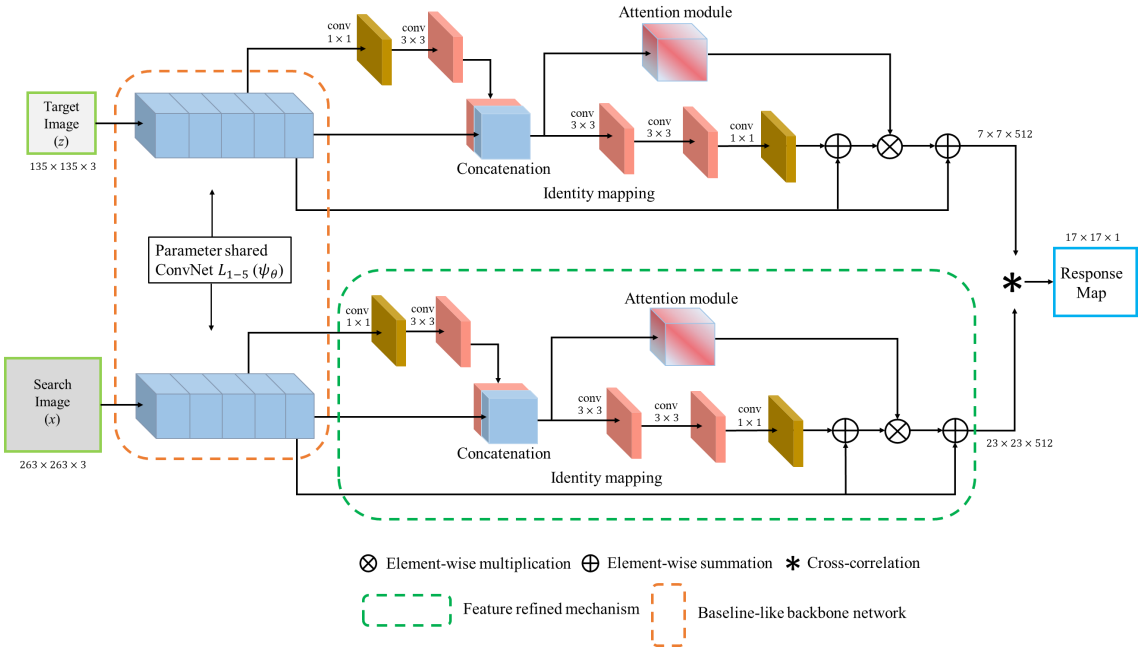

Figure 1. The pipeline of the proposed tracker which consists of two parallel branches: one for target image and another for search image. We have calculated the response map using cross-correlation between target and search feature map at the end of the Y-shaped network. The target branch is computed once during tracking and remained static for the rest of the image frames. This maximum point from the response map is used to predict the target location in the subsequent search frames of a video sequence.

3.1. Siamese Baseline Network

Siamese network was first introduced to solve the tracking problem by Bertinetto et al. [10]. The template matching strategy during tracking is the basic principle of the siamese network. It usually comprised of two parallel feature extractors that learn through parameter sharing. The fully convolutional neural networks (FCNs) are embedded as a feature extractor backbone for two input patches where one is responsible for the exemplar image (z), and another for the search image (x).

Image z is considered as an object of interest that is tracked in the subsequent frames of the video utilizing image x. The objective of the siamese network is to produce a response map to learn the similarity score between exemplar and search images. The maximum similarity score position in the response map is considered as the corresponding target location. To compute the similarity score, a cross-correlation operation is performed at the end of the feature embedding layers as,

$$r_\theta(z, x) = \psi_\theta(z) * \psi_\theta(x) + b \cdot \mathbb{1}, \tag{1}$$

where $\psi_\theta(\cdot)$ represents the fully convolutional embedding function that processed the input images by learning parameters θ, $b \cdot \mathbb{1}$ used as bias value $b \in \mathbb{R}$ for each location, and $*$ represents for the cross-correlation operation between exemplar and search embedded feature map.

Generally, siamese baseline tracker exploits a single image feature map and remains fixed while tracking the whole sequences. Due to this steady nature of siamese-based trackers, it encounters difficulties to handle the challenging scenarios including occlusion, background clutter, fast motion, and motion blur. Hence, learning discrimination ability from the target object to background and also able to learn more generalization ability for the target template are the foremost concerns when designing a tracker framework. However, the baseline network simultaneously learns the target feature representation by solving Equation (1), which may lead to less generalization and discrimination ability.

To overcome these shortcomings, we extend baseline siamese architecture by introducing a feature refined network in the semantic layer that enables the network to be more generalized and discriminative through learning fine target feature representation. The proposed tracker pipeline can be summarized as,

$$r_\theta(z, x) = \xi(\psi_\theta(z) * \psi_\theta(x)) + b \cdot \mathbb{1}, \tag{2}$$

where ξ represents the feature refined network for both target and search branches of siamese pipeline.

To train the proposed framework, we randomly select paired images (z, x) as target, and search regions to compute the response maps $r_\theta(z, x)$. Similar to SiamFC [10], we adopt binary cross-entropy loss function to calculate the overall training loss of the framework as,

$$L(r_\theta(z, x), y) = \frac{1}{|N|} \sum_{n \in N} \log(1 + \exp(r_\theta(z, x)[n].y[n])), \tag{3}$$

where N represents all of the possible response map locations $n \in N$, and ground truth response map is calculated as $y[n] \in \{+1, 0\}$ for respective location n.

We obtain stochastic gradient descent as an optimizer to all over the training pairs T to learn overall proposed network parameter θ using the following equation:

$$argmin_\theta \frac{1}{T} \sum_{i=1}^{T} L(r_\theta(z_i, x_i), y_i). \tag{4}$$

3.2. Feature Refine Network

To handle the limitations of the underlying siamese architecture, we introduce a novel Feature Refined Network (FRN) inside the baseline framework. The feature map from the deep layers is capable of capturing the semantic feature information of salient target regions. In contrast, features from shallow layers hinder the generalized target feature representation by apprehending non-salient details [53]. Hence, FRN learns to focus on the salient region of the target by utilizing a high-level semantic feature map from the backbone network. The FRN mechanism is illustrated in Figure 1.

Since the spatial dimension of the feature maps from different convolutional layers are not equivalent, we apply a 3×3 and 1×1 convolution operation to down-sample the feature map L_4 and produce the invariable feature dimension like L_5. To incorporate high-level features from different convolution layers, we empirically choose features from the last two convolution layers that able to capture more semantic information. We integrate these two feature maps by applying concatenation

in the channel dimension represented as F_{cat}. After concatenating the high-level semantic features, a residual refine module F_{cat}^{γ} is employed to learn the finer target information. F_{cat}^{γ} consists of two 3×3 and a single 1×1 convolutional operations with learn-able ReLU activation functions called PReLU [54]. PReLU works as different compared to the ReLU activation function, and it introduces a learnable parameter for negative values during training so that it can help to learn better nonlinear adaptability with other network parameters such as weights and biases. Hence, we compute the refined feature by incorporating identity mapping L_5 using element-wise summation. In general, the FRN can be summarized as

$$F_{cat} = Concat(f_{\varphi}^{3\times3}(f_{\varphi}^{1\times1}(L_4)), L_5), \tag{5}$$

$$F_{cat}^{\gamma} = f_{\varphi}^{1\times1}(PReLU(f_{\varphi}^{3\times3}(PReLU(f_{\varphi}^{3\times3}(F_{cat}))))), \tag{6}$$

$$\xi(\psi_{\theta}(\cdot)) = ((F_{cat}^{\gamma} \oplus L_5) \otimes F_{cat}^{\alpha}) \oplus L_5, \tag{7}$$

where $\xi(\psi_{\theta}(\cdot))$ represents the ultimate refined feature over corresponding branch, φ is responsible for convolution operation, F_{cat}^{α} denotes the attentional feature map, and Parametric ReLU (PReLU) [54] denoted as $PReLU(x_i) = max(0, x_i) + w_i * min(0, x_i)$.

The alternative way we can define the PReLU equation as

$$PReLU(x_i) = \begin{cases} x_i, & x_i \geq 0 \\ w_i x_i, & \text{otherwise} \end{cases}, \tag{8}$$

where i represents any input from the i^{th} channel and w_i is a learnable parameter.

During backpropagation, the gradient of the PReLU activation function is computed by:

$$\frac{\partial PReLU(x_i)}{\partial w_i} = \begin{cases} 0, & x_i \geq 0 \\ x_i, & \text{otherwise} \end{cases}. \tag{9}$$

Furthermore, an attention module is integrated to increase the network's discriminative ability between target and background information. We present a channel attention module to augment our network's discriminative learning. The channel attention reduces the weights of less important channels by adding lower weights while strengthening the target focused channels with higher weights. The details attention module has been discussed in the later section. The normalized features from the attention module are propagated to fuse with the refined feature using element-wise multiplication. Therefore, the ultimate refined features $\xi(\psi_{\theta}(\cdot))$ from FRN has been computed by employing L_5 as a residual skip connection based identity mapping operation.

3.3. Feature Attention Network

To perceive any objects from a scene, the human visual system focuses on the important parts of the object rather than focusing on the whole scene. Similarly, we integrate an attention module to focus more on the salient target feature than the background. To strengthen the feature fusion network module, we introduce a feature attention module inside the FRN mechanism. Figure 2 depicts the attention module. It is capable of intensifying target information details by exploiting target-specific weights while suppressing the background information. Each channel from a convolutional feature map is considered as a feature detector, and the contribution of each channel is not the same during deep convolution operations. Hence, to obtain the attention feature map, we incorporate a channel attention module to learn the salient information from the image feature map.

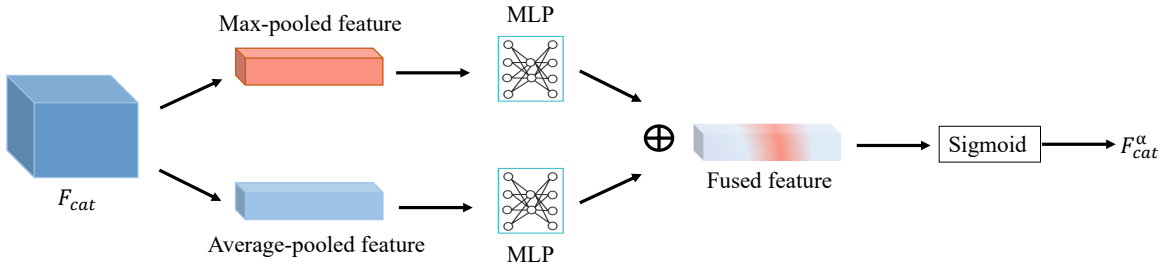

Figure 2. The channel attention module over the concatenated feature map.

The channel attention module learns through exploiting the inter-channel relationship of the feature map that is useful to improve the discrimination ability of the network from background to foreground. It helps to extend the target adaptation ability of the proposed tracker by strengthening more weighted channels while minimizing the impact of less weighted channels. Usually, average pooling utilizes to aggregate the spatial information for channel attention. SAsiam [51], RASNet [55], and AWMF-CFNet [56] trackers used only average pooled feature to improve the target feature representation power. Since our model proposed to track a single object, we also include the max-pooled feature F_{max} beside the average-pooled feature F_{avg} to learn our network finer clue of the distinctive target features rather than considering only the overall idea of the feature map.

For computing our attention module, concatenated feature map F_{cat} has taken as input from the high-level convolution features. We squeeze the spatial dimension by performing both average pooling and max pooling operations to produce two pooled featured map F_{avg} and F_{max}. F_{avg} and F_{max} are considered as generated different special feature descriptors for the feature map. Both feature descriptors are propagated to the multi-layer perceptron (MLP) that comprises of two fully-connected layers (fc_1 with 256 nodes and fc_2 with 512 nodes) and a non-linear operation after performing fc_1 layer, respectively. To facilitate the heterogeneity of the feature descriptors, we perform MLP for both descriptors individually and produce two feature vectors F^v_{avg} and F^v_{max}. Then, we employ element-wise summation operation to fuse both F^v_{avg} and F^v_{max} together. The output of fused feature vectors are normalized through a sigmoid function. Therefore, we broadcast the overall attentional effect F^{α}_{cat} to the residual refined feature map by using element-wise multiplication. In short, the overall attention pipeline are computed as,

$$F^v_{max} = fc_2^{\times 512}(ReLU(fc_1^{\times 256}(MaxPool(F_{cat})))), \tag{10}$$

$$F^v_{avg} = fc_2^{\times 512}(ReLU(fc_1^{\times 256}(AvgPool(F_{cat})))), \tag{11}$$

$$F^{\alpha}_{cat} = sigmoid(F^v_{max} \oplus F^v_{avg}), \tag{12}$$

where the usual sigmoid function denoted as $sigmoid(t) = \frac{1}{1+e^{-t}}$.

4. Experiments

This section presents a detailed explanation of the practical consideration of implementation, including backbone network architecture and parameter setting for training and testing. We also include the evaluation and comparison of our proposed model performance with other's state-of-the-art trackers on various tracking benchmarks such as OTB100 [16,17], OTB50 [16,17], UAV123 [18], TC128 [19], VOT2017 [20], and VOT2018 [21].

4.1. Implementation Details

Our tracker has been implemented using PyTorch deep learning framework with Python. To perform all of the experiments, a desktop with an Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz and Nvidia Geforce RTX 2080 Super GPU has been used.

4.1.1. Network Architecture

Siamese network extracts the feature map from the target and search images by exploiting two parallel networks as a backbone. We introduce two fully convolutional branches in our tracking architecture instead of considering fully connected layers to reduce the overall parameter overhead. Moreover, our network learns through parameter sharing and makes it more generalized during testing. We utilize a different backbone network rather than underlying SiamFC [10] that is better adaptable for the proposed feature refined network. The shared backbone that we have utilized to learn, it is developed by modifying underlying Alexnet architecture [57]. The proposed network includes five convolutional layers with different input-sized images and feature maps, keeping the overall network's kernel size and stride are identical to the SiamFC tracker.

Table 1 shows the overall tracker backbone for the feature extraction network. The network output for target and search images are used to calculate the similarity score (response map) between two extracted feature maps using a cross-correlation operation. Like the SiamFC network, we also exploit the same sized ($17 \times 17 \times 1$) response map to keep the overall tracking framework simple.

Table 1. The backbone architecture of the proposed framework for the feature extraction.

Layers	Target Image	Search Image	Kernel Size	No. of Stride	Feature Map Size
Input image	135×135	263×263	-	-	$\times 3$
convol1	63×63	127×127	11×11	2	$\times 192$
pool1	31×31	63×63	3×3	2	$\times 192$
convol2	27×27	59×59	5×5	1	$\times 512$
pool2	13×13	29×29	3×3	2	$\times 512$
convol3	11×11	27×27	3×3	1	$\times 768$
convol4	9×9	25×25	3×3	1	$\times 768$
convol5	7×7	23×23	3×3	1	$\times 512$

4.1.2. Training

The data curation steps of our method have been integrated with the training process. To prepare the training dataset, we consider the ImageNet Large Scale Visual Recognition Challenge (ILSVRC15) called ImageNet-2015 [58] and GOT-10k [59] dataset together. We adopt the target and search images from these datasets. We utilize $135 \times 135 \times 3$ and $263 \times 263 \times 3$ image-sized for target (z) and search (x) input frames respectively. We preprocess and crop our target and search images where the desired object is located in the center of the frame. If any of the cropped images exceed the original image boundary, we fill it up by the RGB mean of the original image to meet our target and search image size. We perform all of the data curation settings during the training time; therefore, we do not process the training data separately. From ImageNet-2015 and GOT-10k dataset sequences, we randomly choose only ten training pairs (z, x) per sequence within the nearest hundred frames to learn the target's appearance changes.

To train our proposed model, we utilize a similar parameter setting, like the SiamFC [10], to keep our overall training procedure simple. We apply end-to-end offline training for our network and save the trained model for testing purposes. We train our model from scratch and use Stochastic Gradient Descent (SGD) with the learning rate varies from 10^{-2} to 10^{-5} exponentially. We set 0.9 for momentum and weight decay to 5^{-4}, and choose a mini-batch size of 32 empirically for training.

4.1.3. Testing

During testing, we consider the pre-trained model that we trained offline and the first frame of the desired video sequence as a target image (z). In our tracking framework, we utilize only the first frame to track the whole sequence of the video. After extracting the feature map weights of the pre-trained model for the target image, the target branch remains stationary, whereas the search branch will be changing for every upcoming frame. Since we compute the target image once, it ensures to facilitate performing high tracking speed. Moreover, we use three scaled search images to overcome the target scaled changes [10] that make the model more robust against scale variation of the objects and increase accuracy.

Before determining the target location in the subsequent frames, we perform a cross-correlation between z and three scaled $1.0375^{\{-1,0,+1\}}$ search images for x to get three correlation feature maps or response maps ($17 \times 17 \times 1$) for finding the highest response map. By exploiting the similarity score of the correlation feature maps, we select a single response map representing the highest score among them. Therefore, we use that score for determining the ultimate target location on that corresponding position in the search image. Similarly, our model infers the target location for the whole sequence.

To further extend the proposed tracker's discriminative ability, we introduce our training and testing procedures separately. During training, both of the backbone network branches for target and search images were identical, enhancing similarity learning and producing better response maps useful to predict effective target location in the tracking time. But the objective of the proposed tracker is to be more generalized and capable of tracking the objects at high-speed. The identical backbone network is prone to generate a less discriminative score map for the object from the background. Hence, it fails to handle challenging sequences such as background clutter. Another benefit of keeping simple the search branch is to enhance the tracker speed; otherwise, it exploits the feature map for every next frame of the image sequence. Therefore, during testing, our proposed model competent to preserve high tracking accuracy beyond real-time speed.

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

To evaluate the tracking model, performance computing success and precision metrics are the most common practice. The precision score stands for Center Location Error (CLE), which measures the center pixel distance between the predicted and ground-truth bounding boxes. On the other hand, the success score represents the overlap score or Intersection over Union (IoU) between the predicted and ground-truth bounding boxes.

For the comparison of the proposed trackers with other state-of-the-art trackers, we consider the popular single object tracking benchmarks including OTB100 [16,17], OTB50 [16,17], UAV123 [18], TC128 [19], VOT2017 [20], and VOT2018 [21]. We utilize the official toolkit of OTB and VOT to produce the tracker results and comparisons.

4.2.1. Experiments on OTB100 Benchmark

OTB100 or OTB2015 benchmark is used frequently to measure and compare the tracking performance among all of the available tracking benchmarks. This benchmark draws attention to the tracking community, which contains a hundred different challenging video sequences. The sequences are divided into 11 sub-categories to express the challenging sequences individually. Illumination variation (IV), scale variation (SV), occlusion (OC), deformation (DF), motion blur (MB), fast motion (FM), in-plane rotation (IR), out-of-plane rotation (OR), out-of-view (OV), background clutter (BC), and low resolution (LR) are the divided challenging sub-categories. Each category includes several video sequences to represent a single challenging attribute. However, we compare our proposed tracker with other popular trackers including MemTrack [14], DSiamM [11], SiamFC [10], SiamTri [12], TRACA [7], MLT [60], SRDCF [61], UDT [62], and CFnet [13] for performance justification.

Figure 3 displays the overall precision and success plots over the OTB100 benchmark. The proposed tracker SiamFRN perform 63.6% for the success score and 84.0% for the precision score. Our tracker shows an improvement of 8.95% in precision and 9.28% in success scores than the baseline SiamFC tracker. It also achieve 2.44%, 3.07%, 5.40%, 7.55%, and 10.53% increment in precision score and 1.60%, 5.12%, 5.82%, 7.80%, and 8.35% increment in success score compared to MemTrack, DSiamM, MLT, SiamTri, and UDT trackers, respectively.

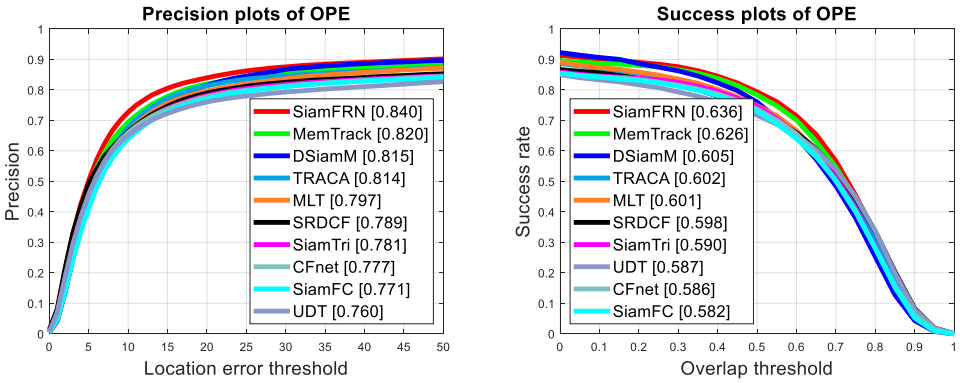

Figure 3. Compared overall precision and success plots over OTB100 benchmark.

To exploit the proposed tracker's effectiveness, we further do extensive experiments on individual challenging attributes on the same benchmark. Figures 4 and 5 illustrates the performance plots for success and precision scores, respectively. Each plot from these figures expresses the challenging image sequences' performance regarding success and precision scores. The proposed tracker achieves a dominant performance on most of the challenges (SV, DF, MB, FM, OC, BC, OV, IR, and OR) on both performance plots over other state-of-the-art trackers. It has shown steady performance even in challenging scenarios that is one of the significant criteria for a successful visual object tracker. SiamFRN also can operate at 60 fps during testing that is beyond real-time.

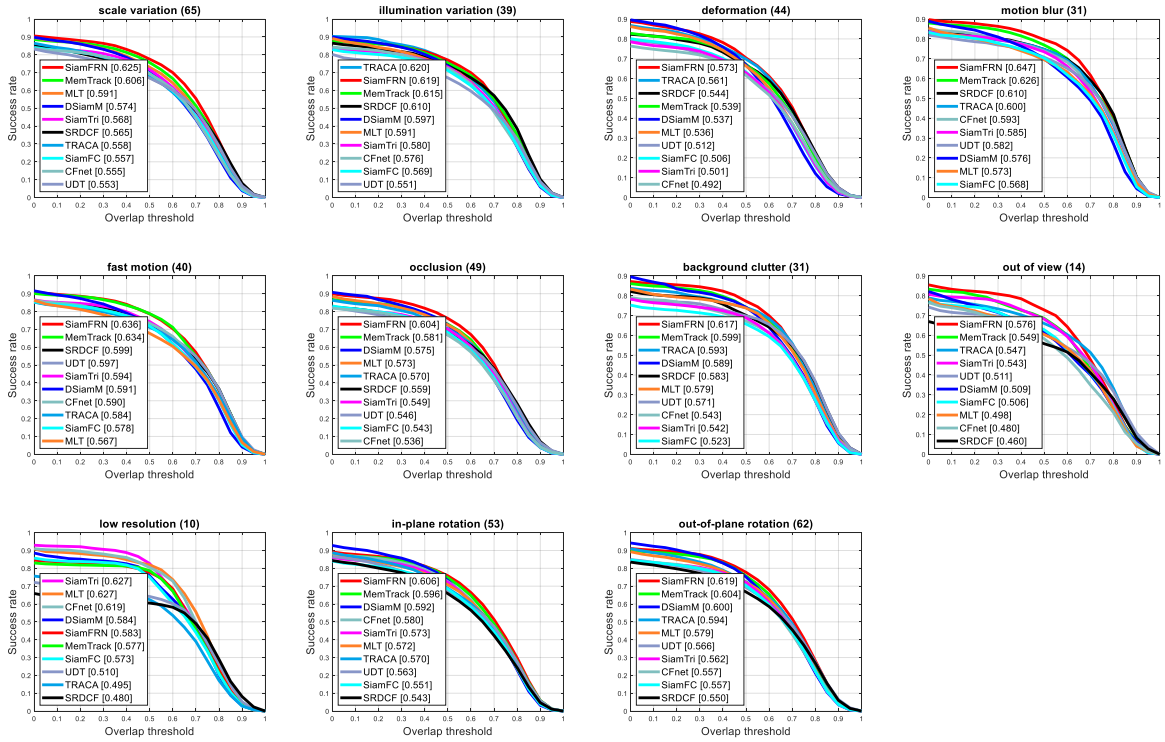

Figure 4. Compared success plots for individual challenging attributes including SV, IV, DF, MB, FM, OC, BC, OV, LR, IR, and OR over OTB100 benchmark.

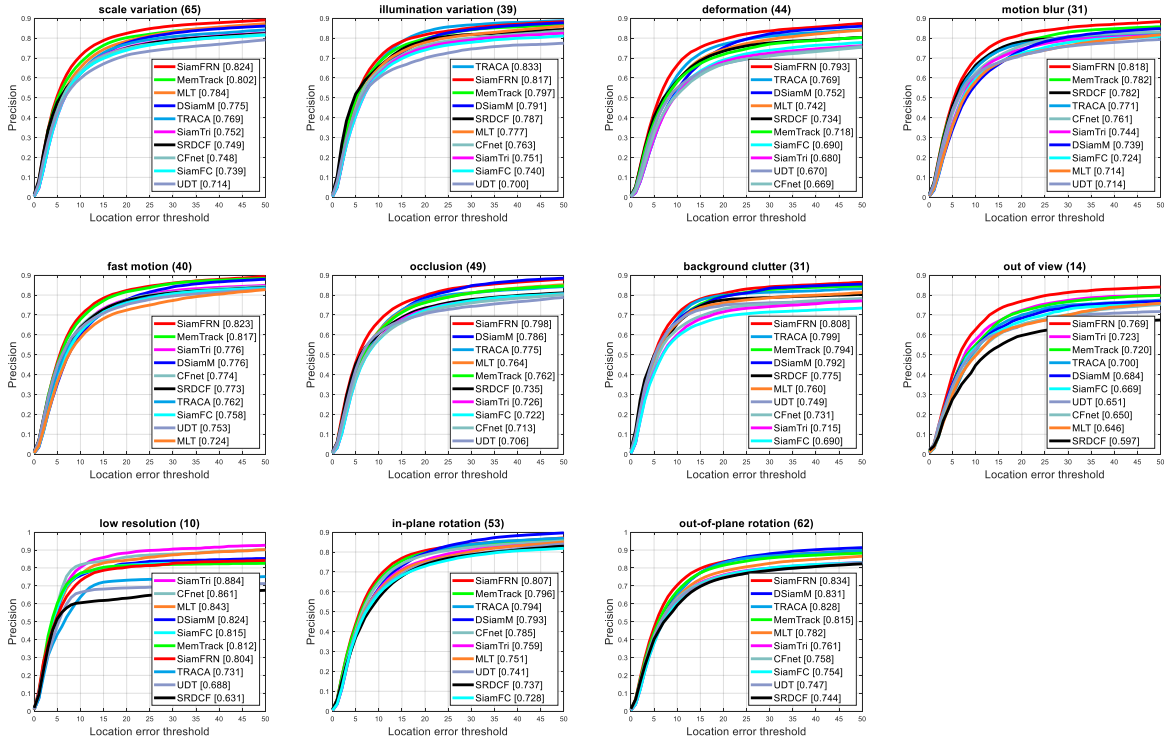

Figure 5. Compared precision plots for individual challenging attributes including SV, IV, DF, MB, FM, OC, BC, OV, LR, IR, and OR over OTB100 benchmark.

Furthermore, we perform a qualitative comparison among recent trackers to evaluate the frame by frame performance analysis for visual understanding. Figure 6 delineates the qualitative study, showing the dominative performance of the proposed tracker over state-of-the-art trackers. Therefore, our tracker preserves a balance between tracking high speed and accuracy with significant robustness against challenges.

Figure 6. The qualitative analysis on some challenging sequences including bird1, bolt2, dragon baby, motor rolling, and rubik from OTB100 benchmark for visual understanding.

4.2.2. Experiments on OTB50 Benchmark

The most challenging fifty sequences from the OTB100 benchmark have been combined to make a separate dataset named OTB50 benchmark [16,17]. It is possible to verify the tracker's ability to track the desired object in such difficult scenes. Therefore, to check out how our tracker performs on these challenging sequences, we use the trackers same as evaluated on the OTB100 benchmark for the performance comparison.

The overall tracking comparison is shown in Figure 7 for OTB50 benchmark. The proposed tracker achieves overall 77.8% and 58.1% precision and success scores respectively. SiamFRN outperforms the underlying SiamFC by 12.43% on precision score and 12.60% on success score. Our tracker shows 3.87%, 6.28%, and 9.12% improvement on precision and 3.57%, 7.79%, and 9.42% success plots over memory attention based tracker MemTrack [14], correlation filter based CFnet [13] tracker, and triplet loss with siamese network SiamTri [12], respectively. It also achieves better results over other recent trackers including DSiamM [11], TRACA [7], MLT [60], SRDCF [61], and UDT [62].

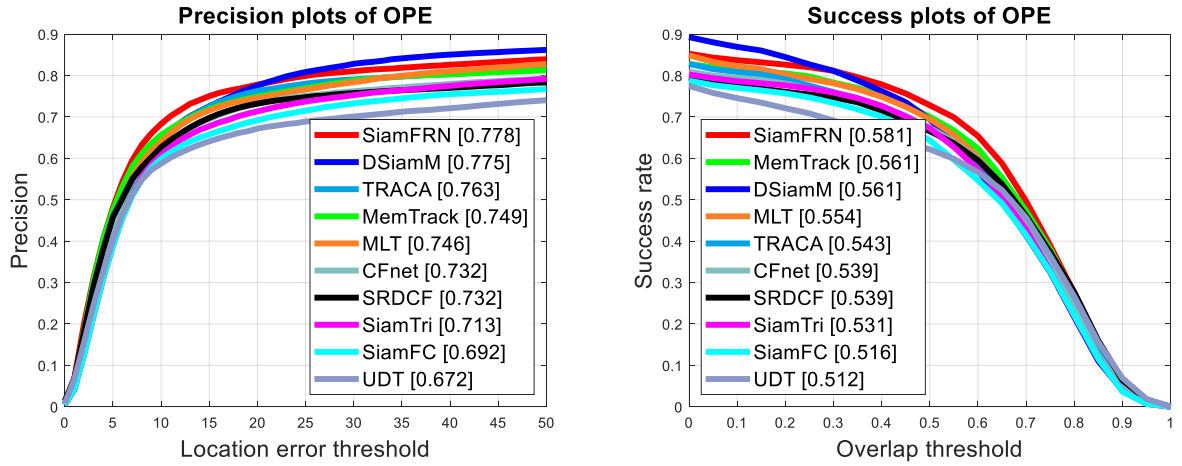

Figure 7. Compared overall precision and success plots over OTB50 benchmark.

The overall comparison shows that the proposed tracker effectively handles more complicated sequences and performs better in both terms of success and precision measures against the compared trackers.

4.2.3. Experiments on UAV123 Benchmark

The Unmanned Aerial Vehicle (UAV) includes 123 video sequences to comprise the UAV123 benchmark [18], which is one of the largest tracking benchmarks that captured using UAV to cover low altitude aerial videos. This benchmark recently got more attention to the tracking community due to its real-life applications, including wildlife monitoring, local surveillance, and navigation system. ECO [6], SRDCF [61], MEEM [63], SAMF [64], MUSTER [65], DSST [66], Struck [67], and KCF [68] trackers are considered for the comparison.

Figure 8 presents overall tracking results comparison on the UAV123 benchmark. The proposed SiamFRN tracker performs 74.3% and 52.1% for precision and success plots. The graph shows that the proposed tracker outperforms all of the compared trackers in both performance metrics except ECO for success score, where it has shown a bit less score. However, ECO does not operate in real-time speed, whereas our proposed tracker can operate even more than real-time speed (60 fps) with a similar amount of accuracy. Since the UAV123 benchmark is utilized for real-life applications, the tracker should be performed in real-time to complete the preferred demand. Another version of the ECO tracker named ECOhc [6], which can operate in real-time but shown 2.48% and 2.96% less performance for precision and success plots, respectively, than our proposed method.

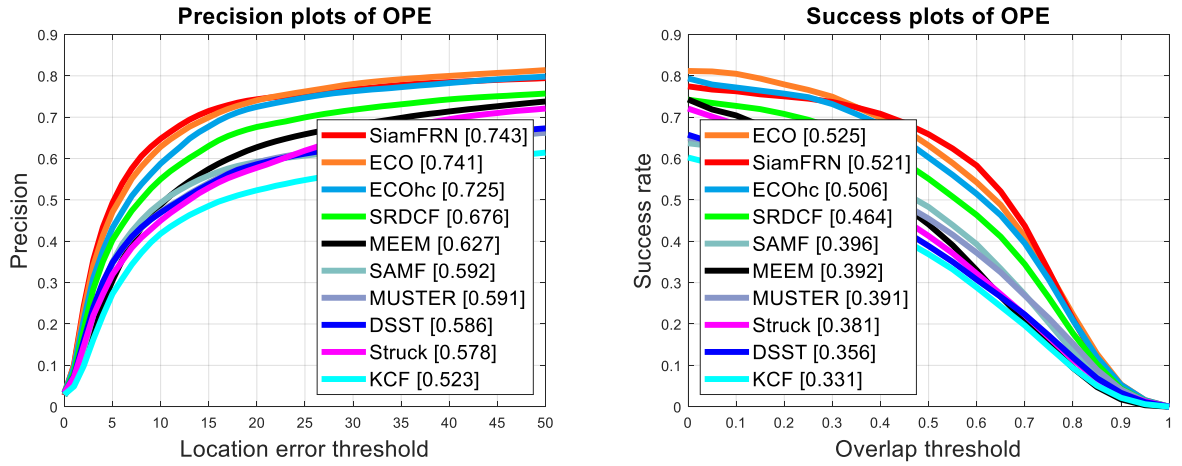

Figure 8. Compared overall precision and success plots over UAV123 benchmark.

Therefore, the compared results validate that the proposed tracker will be more usable for real-time applications than others in terms of high speed and accuracy.

4.2.4. Experiments on TC128 Benchmark

TC-128 [19] dataset introduced to integrate color information providing rich discriminative clues on the gray-scale images. We perform one-pass evaluation on TempleColor-128 (TC-128) benchmark containing 128 fully-annotated image sequences. Similar to the OTB benchmark evaluation, we adopt success and precision plots to compare with the state-of-the-art trackers. We compare our tracker with SCSAtt [69], MEEM [63], SRDCF [61], MUSTER [65], SAMF [64], KCF [68], DSST [66], truck [67], and CSK [70] on this benchmark.

The compared trackers are presented in Figure 9 that shows the success and precision plots. The proposed tracker SiamFRN achieves a 71.9% precision score and a 52.8% success score over the TC128 benchmark. SiamFRN has shown dominant performance among compared trackers other than the SCSAtt tracker. Our tracker achieves 8.61%, 12.70%, and 29.60% improvement on precision score and 8.64%, 12.10%, and 37.14% improvement from SRDCF, MUSTER, and KCF, respectively. Thus, the proposed tracker shows comparable results among other trackers over this benchmark.

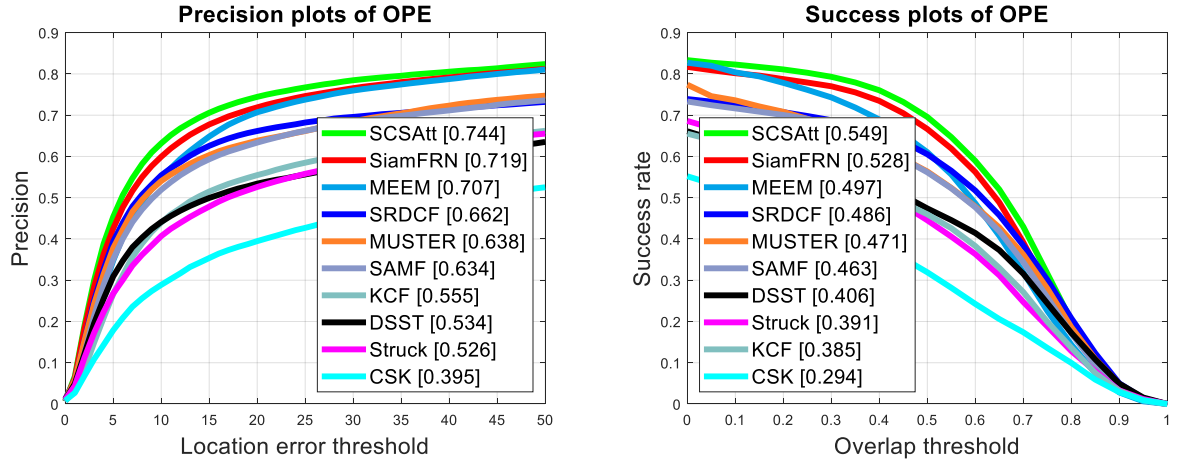

Figure 9. Compared overall precision and success plots over TC128 benchmark.

4.2.5. Experiments on VOT2017 and VOT2018 Benchmark

Both VOT2017 [20] and VOT2018 [21] benchmarks included identical 60 video sequences for tracking. So, we consider the trackers from both VOT2017 and VOT2018 challenges to our comparison framework. However, the VOT evaluation metrics are different from the above benchmarks (OTB100, OTB50, TC-128, and UAV123). The VOT benchmarks are computed on three important evaluation metrics, including accuracy, robustness, and expected average overlap (EAO). Accuracy is calculated using the average overlap of the successful tracking, while robustness is evaluated by utilizing failure times during tracking. And, EAO is measured by considering both accuracy and robustness into a single metric. However, a good tracker represents higher accuracy and EAO scores but a lower robustness score. We encourage interested readers to look at the article [20,21] for obtaining more information about VOT benchmarks and challenges.

Table 2 shows the compared trackers and their corresponding accuracy, robustness, EAO score, and tracking speed. We consider GradNet [71], SCS-Siam [72], MemTrack [14], ECOhc [6], SATIN [73], DSiam [11], CSRDCF [74], SiamFC [10], DCFNet [75], DensSiam [76], DSST [66], and SRDCF [61] trackers for the comparison. The top three performed trackers in terms of VOT measure metrics are highlighted using red, green, and blue colors, respectively. From Table 2, it is clear that the proposed tracker achieves highest score in terms of accuracy and robustness and second highest score for EAO score with real-time tracking speed.

Table 2. Compared our proposed SiamFRN tracker over VOT2017 and VOT2018 benchmarks with state-of-the-art trackers in terms of accuracy, robustness, EAO, and tracking Speed. The top three results are highlighted for a better view.

Tracker	Accuracy (↑)	Robustness(↓)	EAO (↑)	Speed (fps)
Ours	0.54	0.22	0.25	60
GradNet [71]	0.50	0.37	0.24	80
SCS-Siam [72]	0.52	0.29	0.24	73
MemTrack [14]	0.49	1.77	0.24	50
ECOhc [6]	0.49	0.44	0.24	60
SATIN [73]	0.49	1.34	0.28	24
DSiam [11]	0.51	0.67	0.20	6
CSRDCF[74]	0.49	0.49	0.25	13
SiamFC [10]	0.50	0.59	0.19	86
DCFNet [75]	0.47	0.54	0.18	60
DensSiam [76]	0.46	0.69	0.17	60
DSST [66]	0.39	1.45	0.08	24
SRDCF [61]	0.49	0.97	0.12	6

4.3. Ablation Analysis

In this section, we investigate the effectiveness of the individual components of the proposed tracker. To verify the contribution of individual design choice, we carried out an extensive ablation study on OTB100 benchmark [16,17]. To preserve the rational comparison among the tracker variants, we considered the same training datasets (ImageNet-2015 [58] and GOT-10k [59]) to train them.

Table 3 shows the overview of the ablation results of several variants of the proposed tracker. The last row from Table 3 corresponds to the baseline SiamFC network that we re-implemented and trained from scratch to utilize as a backbone network of the proposed method, which performed 59.7% and 79.8% in success and precision metrics, respectively. The two main components of the proposed framework are the residual refine module (RRM) and the attention module (AM). The RRM is responsible for improving the network feature representation power. It helps to apprehend the important target feature index, enabling the prediction of the target location precisely. On the other hand, we utilized the channel attention module as AM because each feature channel's contribution is

different. It prioritizes the important channels from the feature map by embedding relevant weights to the features.

Table 3. Ablation analysis results on the OTB100 benchmark. L1–L5 indicates feature for convolutional layers from 1 to 5, and checkmark represents the feature selection for the evaluation. The proposed model variants are evaluated on the OTB100 benchmark, and the highest result is highlighted.

Low-Level			High-Level		FRN Mechanism		Performance Results	
L1	**L2**	**L3**	**L4**	**L5**	**Residual Refine Module**	**Attention Module**	**Success Score**	**Precision Score**
✓	✓	✓			✓		58.5	78.8
✓	✓	✓				✓	57.9	77.3
✓	✓	✓			✓	✓	61.7	81.9
			✓	✓	✓		62.0	82.3
			✓	✓		✓	62.3	82.5
			✓	✓	✓	✓	**63.6**	**84.0**
✓	✓	✓	✓	✓	✓	✓	62.9	83.2
		✓	✓		✓	✓	62.2	82.0
	✓	✓			✓	✓	61.3	80.7
		✓		✓	✓	✓	63.0	82.8
	SiamFC-baseline				✓	✓	62.5	82.7
	SiamFC-baseline				✓		61.2	80.9
	SiamFC-baseline					✓	61.9	81.9
				SiamFC-baseline			59.7	79.8

Firstly, in the ablation analysis, we verify individual RRM and AM effectiveness by incorporating them separately at the end of the baseline siamFC network. The variant of baseline SiamFC with the RRM performs 61.2% success score and 80.9% precision score, which shows 2.51% and 1.38% progress in success and precision score from SiamFC-baseline tracker. The AM module integration achieves 3.69% and 2.63% improvement from the baseline siamFC for success and precision score. When we embedded both modules together, we noticed that the combination of both modules able to perform better than the individual module. The fourth row from the bottom of the table presented the combined RRM and AM module performance results. We have not considered the concatenated-feature map from the convolutional layers during RRM and AM modules integration with the baseline SiamFC.

Then we carried our experiments to explore the effectiveness of combined features. To utilized fused features, we concatenated features from different convolution layers. Choosing the appropriate feature combination from convolutional layers may directly affect the tracker performance. To ensure the fused feature effects on network performance, we verified RRM and AM impact individually and combinedly. In our experiments, we considered convolutional layers as a low-level (layer 1 to 3) and high-level (layer 4 to 5) feature map in our analysis. Usually, the higher convolutional layers of a network learn more semantic features than lower layers. We found that the low-level feature-based tracker variants cannot perform well. Even the performance of integrating a low-level fused feature map with RRM or AM are relatively worse than considering a single convolutional layer feature.

Furthermore, we validate the effectiveness of high-level feature integration with RRM and AM individually and combinedly. From our experimental results, we observed that the high-level fused feature-based network boosts the tracker's performance. We also perceived that combining RRM and AM with the high-level fused feature hugely augments the network performance, whereas employing a single feature refine module performance is relatively lower. Similarly, to prove the efficiency of several proposed tracker variants, we investigate features from layers 2 and 3, 3 and 4, and 3 and 5 together are used to incorporate with RRM and AM. These variants are performed inferior to selecting the feature from the last two convolutional layers (high-level features). Besides, we considered all convolutional layers for embedding RRM and AM together, which performed 62.9% in success and 83.2% in the precision score. However, we finalized the combination of high-level (layer 4 and 5) fused features with the combined RRM, and AM offered the best performance among all variants from the extensive ablation studies. The combination of RRM and AM are represented as the feature refine

mechanism (FRN) in our proposed framework. Overall, the FRN mechanism improves the feature representation and discrimination ability of the underlying siamese network. We consider it our proposed SiamFRN method that achieves 63.6% and 84.0% success and precision score, respectively.

Finally, to analyze the further effectiveness of SiamFRN, we performed experiments on several popular tracking datasets with state-of-the-art trackers. Each compared dataset are different from each other. The performance comparison results of the proposed method are demonstrated in Figures 3–9 for OTB, UAV123, and TC128 datasets. Table 2 shows the performance comparison (accuracy, robustness, and EAO) for VOT2017 and VOT2018 datasets. We also analyzed tracking speed, and we observed that our tracker operates more than the real-time frame rate, which demonstrates that the proposed SiamFRN can be applicable for real-time applications. Overall, the proposed method reported a balanced performance over the tracking speed (60 fps) with accuracy in both success and precision metrics, which the efficacy of the proposed framework.

4.4. Discussion

This work utilized an FRN mechanism that helps to improve the underlying Siamese network's feature representation and discriminative ability. The FRM mechanism comprises of RRM and AM. The RRM enhances the network's representation ability by utilizing high-level semantic feature information for salient target regions. On the other hand, AM is responsible for improving the proposed network discriminative power by re-calibrating the target-specific channel weights. The feature learning of individual channels is not identical during training the network. AM helps to provide more weights to prioritize important features while reducing the importance of irrelevant channels by embedding lower weights. Therefore, by combining these two modules network learns better representation power for the template image and more discrimination ability between target and background, enhancing the overall tracking performance on challenging sequences.

We conducted a detailed ablation study to verify our network design before finalizing the proposed framework. Table 3 summarizes the comprehensive empirical analysis of the several variants of the proposed tracker on the OTB100 benchmark. As each module of the SiamFRN has different tasks, choosing an appropriate combination of the convolutional features impacts tracking performance. After validating the effectiveness of individual components of the proposed method on benchmark datasets, we embedded it with other network modules to analyze the effect on overall network performance. From our ablation analysis, we observed that the integration of the FRN mechanism with high-level features from the backbone network achieves the best performance among the tracker variants, which is our proposed SiamFRN tracker. SiamFRN works better than the other variants because it utilized the high-level features from the backbone network that are useful to learn semantic information for target representation by avoiding the non-salient detail information from the low-level feature map.

To analyze the proposed method, we compared our tracking results with other state-of-the-art trackers. We conducted several experiments on OTB100 [16,17], OTB50 [16,17], UAV123 [18], TC128 [19], VOT2017 [20], and VOT2018 [21] benchmark datasets. The proposed tracker's overall performance seems better; however, it shows some limitations during the performance comparison. We noticed that the proposed method weakly performs in the low-resolution sequences depicted in Figures 4 and 5 compared with other trackers. We also observed that our method has a deficiency on TC128 benchmark against the SCSAtt [69] tracker that introduced stacked channel-spatial attention based Siamese network for tracking. SCSAtt utilized the spatial feature map to locate the target besides focusing on the object's informative part. Our ablation analysis shows that the high-level feature with the FRN module improves the tracker accuracy. However, the only high-level feature becomes less effective when target appearance changes frequently and appears on low-resolution sequences. We investigated that the proposed method does not consider the low-level feature map for refining through the FRN module. Therefore, we presume it may lose some essential positional details information of the target, which could be useful for locating the target position more precisely.

Furthermore, we used a single template image while tracking the entire image sequence without further updating is susceptible to tracker reliability. In this case, the proposed tracker shows the weakness of handling such challenging scenarios, particularly when the target object appearance changes continuously. For improving the tracker reliability, MDNet [8], DSAR-CF [77], and MemDTC [78] were proposed template update mechanism within their tracking framework. Usually, these trackers updated the template image frequently with their updating mechanisms during tracking. However, integrating an effective updating mechanism hinders the tracker's simplicity and consumes more time to accurately update the target template. Such trackers increase the model complexity and compromise the high tracking speed that is one of the major concerns for real-time applications.

In contrast, to preserve the simplicity of designing the tracker, we have not introduced the template updating mechanism while building our proposed framework. Therefore, we compute the target image once rather than introducing any updating mechanism or considering multiple image frames as our reference template image to maintain a balanced performance in terms of speed and accuracy using a simple tracking framework. That is the main objective of the proposed tracker. Based on the above discussion, our proposed tracker performs better with some limitations, requiring more investigation to overcome.

5. Conclusions and Future Works

This work proposed a fully convolutional feature refined Siamese network to improve the overall tracking performance. The proposed framework extended the underlying Siamese network by integrating an FRN mechanism enabling the network to learn more generalized feature representation using the high-level semantic feature map. To improve the network's discriminative power, we employed a channel attention mechanism, and altogether it enhanced the robustness of our proposed tracker against challenges without compromising the real-time tracking speed. Furthermore, to verify the effectiveness of the proposed tracker, we performed extensive experiments on large datasets include OTB100 [16,17], OTB50 [16,17], UAV123 [18], TC128 [19], VOT2017 [20], and VOT2018 [21] that shown our tracker outperformed the state-of-the-art results. The experimental results in Section 4 demonstrated the proposed tracker efficiency over several benchmarks, which can operate at (60 fps). Therefore, the proposed tracker preserves balance performance between speed and accuracy, and we believe that it would be useful for real-time applications.

However, the proposed method shows difficulties in handling the frequent appearance changed targets and low-resolution target sequences. Although our method has such disadvantages, it is still outperformed the compared trackers on almost all challenging attributes shown in Figures 4 and 5. In our future work, we are planning to focus on resolving such issues for the proposed tracker with preserving the balanced performance in terms of tracking speed and accuracy by updating the overall tracking framework.

Author Contributions: Conceptualization, M.M.R.; methodology, M.M.R.; software, M.M.R.; validation, M.M.R., M.R.A. and S.K.J.; formal analysis, M.M.R., M.R.A. and L.L.; investigation, S.K.J., L.L. and S.H.K.; resources, L.L. and S.H.K.; data curation, M.M.R.; writing—original draft preparation, M.M.R. and M.R.A.; writing—review and editing, M.M.R., M.R.A. and S.K.J.; visualization, L.L. and S.H.K.; supervision, S.K.J.; project administration, S.K.J.; funding acquisition, S.K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2019R1A2C1010786).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Yao, H.; Cavallaro, A.; Bouwmans, T.; Zhang, Z. Guest Editorial Introduction to the Special Issue on Group and Crowd Behavior Analysis for Intelligent Multicamera Video Surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 405–408. [CrossRef]
2. Lu, W.L.; Ting, J.A.; Little, J.J.; Murphy, K.P. Learning to track and identify players from broadcast sports videos. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1704–1716. [PubMed]
3. Gupta, M.; Kumar, S.; Behera, L.; Subramanian, V.K. A Novel Vision-Based Tracking Algorithm for a Human-Following Mobile Robot. *IEEE Trans. Syst. Man, Cybern. Syst.* **2017**, *47*, 1415–1427. [CrossRef]
4. Menze, M.; Geiger, A. Object scene flow for autonomous vehicles. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3061–3070.
5. Liang, Z.; Shen, J. Local semantic siamese networks for fast tracking. *IEEE Trans. Image Process.* **2019**, *29*, 3351–3364. [CrossRef]
6. 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.
7. Choi, J.; Jin Chang, H.; Fischer, T.; Yun, S.; Lee, K.; Jeong, J.; Demiris, Y.; Young Choi, J. Context-aware deep feature compression for high-speed visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 479–488.
8. Nam, H.; Han, B. Learning Multi-Domain Convolutional Neural Networks for Visual Tracking. *arXiv* **2015**, arXiv:1510.07945.
9. Bromley, J.; Guyon, I.; LeCun, Y.; Säckinger, E.; Shah, R. Signature verification using a "siamese" time delay neural network. *Adv. Neural Inf. Process. Syst.* **1994**, 737–744. [CrossRef]
10. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*; Springer: Amsterdam, The Netherlands, 2016; pp. 850–865.
11. 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.
12. Dong, X.; Shen, J. Triplet loss in siamese network for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 459–474.
13. 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.
14. Yang, T.; Chan, A.B. Learning dynamic memory networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 152–167.
15. Fiaz, M.; Rahman, M.M.; Mahmood, A.; Farooq, S.S.; Baek, K.Y.; Jung, S.K. Adaptive Feature Selection Siamese Networks for Visual Tracking. In *International Workshop on Frontiers of Computer Vision*; Springer: Ibusuki, Japan, 2020; pp. 167–179.
16. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef]
17. 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, 23–28 June 2013; pp. 2411–2418.
18. 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.
19. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef]
20. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Hager, G.; Lukezic, A.; Eldesokey, A.; et al. The visual object tracking vot2017 challenge results. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–27 October 2017; pp. 1949–1972.

21. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; Cehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. The sixth visual object tracking vot2018 challenge results. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
22. Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 1442–1468.
23. Fiaz, M.; Mahmood, A.; Javed, S.; Jung, S.K. Handcrafted and deep trackers: Recent visual object tracking approaches and trends. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–44. [CrossRef]
24. Brendel, W.; Bethge, M. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. *arXiv* **2019**, arXiv:1904.00760.
25. Ahmed, M.R.; Zhang, Y.; Liu, Y.; Liao, H. Single Volume Image Generator and Deep Learning-based ASD Classification. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 3044–3054. [CrossRef] [PubMed]
26. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
27. Shaban, M.; Mahmood, A.; Al-maadeed, S.; Rajpoot, N. Multi-person Head Segmentation in Low Resolution Crowd Scenes Using Convolutional Encoder-Decoder Framework. In *Representations, Analysis and Recognition of Shape and Motion from Imaging Data*; Chen, L., Ben Amor, B., Ghorbel, F., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 82–92.
28. Gao, F.; Wang, C. Hybrid strategy for traffic light detection by combining classical and self-learning detectors. *IET Intell. Transp. Syst.* **2020**, *14*, 735–741. [CrossRef]
29. Shen, J.; Yu, D.; Deng, L.; Dong, X. Fast Online Tracking With Detection Refinement. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 162–173. [CrossRef]
30. Shen, J.; Liang, Z.; Liu, J.; Sun, H.; Shao, L.; Tao, D. Multiobject Tracking by Submodular Optimization. *IEEE Trans. Cybern.* **2019**, *49*, 1990–2001. [CrossRef]
31. Shen, J.; Peng, J.; Dong, X.; Shao, L.; Porikli, F. Higher Order Energies for Image Segmentation. *IEEE Trans. Image Process.* **2017**, *26*, 4911–4922. [CrossRef] [PubMed]
32. Ross, D.; Lim, J.; Lin, R.S.; Yang, M.H. Incremental Learning for Robust Visual Tracking. *Int. J. Comput. Vis.* **2008**, *77*, 125–141. [CrossRef]
33. 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; pp. 2544–2550.
34. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical convolutional features for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
35. Doulamis, A.; Doulamis, N.; Ntalianis, K.; Kollias, S. An efficient fully unsupervised video object segmentation scheme using an adaptive neural-network classifier architecture. *IEEE Trans. Neural Netw.* **2003**, *14*, 616–630. [CrossRef]
36. 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.
37. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
38. Luo, W.; Schwing, A.G.; Urtasun, R. Efficient Deep Learning for Stereo Matching. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5695–5703.
39. Chen, K.; Tao, W. Once for All: A Two-Flow Convolutional Neural Network for Visual Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *28*, 3377–3386. [CrossRef]
40. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
41. Fiaz, M.; Mahmood, A.; Baek, K.Y.; Farooq, S.S.; Jung, S.K. Improving Object Tracking by Added Noise and Channel Attention. *Sensors* **2020**, *20*, 3780. [CrossRef]

42. Rahman, M.M. A DWT, DCT and SVD based watermarking technique to protect the image piracy. *arXiv* **2013**, arXiv:1307.3294.
43. Rahman, M.M.; Ahammed, M.S.; Ahmed, M.R.; Izhar, M.N. A semi blind watermarking technique for copyright protection of image based on DCT and SVD domain. *Glob. J. Res. Eng.* **2017**, *16*, 9–16.
44. Xu, K.; Ba, J.L.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.S.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 7–9 July 2015; pp. 2048–2057.
45. Woo, S.; Park, J.; Lee, J.Y.; So Kweon, I. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
46. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
47. Zhao, Y.; Liu, Z.; Yang, L.; Cheng, H. Combing RGB and Depth Map Features for human activity recognition. In Proceedings of the 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, Lanzhou, China, 18–21 November 2012; pp. 1–4.
48. Cui, Z.; Xiao, S.; Feng, J.; Yan, S. Recurrently Target-Attending Tracking. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1449–1458.
49. Choi, J.; Chang, H.J.; Jeong, J.; Demiris, Y.; Choi, J.Y. Visual Tracking Using Attention-Modulated Disintegration and Integration. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4321–4330.
50. Choi, J.; Chang, H.J.; Yun, S.; Fischer, T.; Demiris, Y.; Choi, J.Y. Attentional Correlation Filter Network for Adaptive Visual Tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4828–4837.
51. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold siamese network for real-time object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4834–4843.
52. Qin, X.; Fan, Z. Initial Matting-Guided Visual Tracking with Siamese Network. *IEEE Access* **2019**, *7*, 41669–41677. [CrossRef]
53. Deng, Z.; Hu, X.; Zhu, L.; Xu, X.; Qin, J.; Han, G.; Heng, P.A. R3net: Recurrent residual refinement network for saliency detection. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 684–690.
54. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
55. 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.
56. Li, C.; Yang, B. Adaptive weighted CNN features integration for correlation filter tracking. *IEEE Access* **2019**, *7*, 76416–76427. [CrossRef]
57. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
58. 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]
59. 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**. [CrossRef]
60. Choi, J.; Kwon, J.; Lee, K.M. Deep meta learning for real-time target-aware visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 911–920.

61. 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.
62. Wang, N.; Song, Y.; Ma, C.; Zhou, W.; Liu, W.; Li, H. Unsupervised deep tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1308–1317.
63. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust tracking via multiple experts using entropy minimization. In *European Conference on Computer Vision*; Springer: Zurich, Switzerland, 2014; pp. 188–203.
64. Kristan, M.; Pflugfelder, R.; Leonardis, A.; Matas, J.; Cehovin, L.; Nebehay, G.; Vojir, T.; Fernandez, G.; Lukezic, A. The visual object tracking vot2014 challenge results. In Proceedings of the Visual Object Tracking Workshop 2014 at ECCV, Zurich, Switzerland, 6–7, 12 September 2014; Volume 1, p. 6.
65. Hong, Z.; Chen, Z.; Wang, C.; Mei, X.; Prokhorov, D.; Tao, D. Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 749–758.
66. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef]
67. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.M.; Hicks, S.L.; Torr, P.H. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2096–2109. [CrossRef]
68. 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]
69. Rahman, M.M.; Fiaz, M.; Jung, S.K. Efficient Visual Tracking with Stacked Channel-Spatial Attention Learning. *IEEE Access* **2020**, *8*, 100857–100869. [CrossRef]
70. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision*; Springer: Firenze, Italy, 2012; pp. 702–715.
71. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–29 October 2019; pp. 6162–6171.
72. Fiaz, M.; Mahmood, A.; Jung, S.K. Learning soft mask based feature fusion with channel and spatial attention for robust visual object tracking. *Sensors* **2020**, *20*, 4021. [CrossRef] [PubMed]
73. Gao, P.; Yuan, R.; Wang, F.; Xiao, L.; Fujita, H.; Zhang, Y. Siamese attentional keypoint network for high performance visual tracking. *Knowl. Based Syst.* **2020**, *193*, 105448. [CrossRef]
74. Lukezic, A.; Vojir, T.; Cehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
75. Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; Hu, W. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv* **2017**, arXiv:1704.04057.
76. Abdelpakey, M.H.; Shehata, M.S.; Mohamed, M.M. Denssiam: End-to-end densely-siamese network with self-attention model for object tracking. In *International Symposium on Visual Computing*; Springer: Las Vegas, NV, USA, 2018; pp. 463–473.
77. Feng, W.; Han, R.; Guo, Q.; Zhu, J.; Wang, S. Dynamic saliency-aware regularization for correlation filter-based object tracking. *IEEE Trans. Image Process.* **2019**, *28*, 3232–3245. [CrossRef]
78. Yang, T.; Chan, A.B. Visual Tracking via Dynamic Memory Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]