sensors

MDPI

Article

Improving Object Tracking by Added Noise and Channel Attention

Mustansar Fiaz [1], Arif Mahmood [2], Ki Yeol Baek [1], Sehar Shahzad Farooq [1] and Soon Ki Jung [1,*]

[1] School of Computer Science and Engineering, Kyungpook National University, Daegu 41566, Korea; mustansar@knu.ac.kr (M.F.); qkkndq@knu.ac.kr (K.Y.B.); sehar146@knu.ac.kr (S.S.F.)

[2] Department of Computer Science, Information Technology University, Lahore 54000, Pakistan; arif.mahmood@itu.edu.pk

* Correspondence: skjung@knu.ac.kr

Received: 3 June 2020; Accepted: 28 June 2020; Published: 6 July 2020

Abstract: CNN-based trackers, especially those based on Siamese networks, have recently attracted considerable attention because of their relatively good performance and low computational cost. For many Siamese trackers, learning a generic object model from a large-scale dataset is still a challenging task. In the current study, we introduce input noise as regularization in the training data to improve generalization of the learned model. We propose an Input-Regularized Channel Attentional Siamese (IRCA-Siam) tracker which exhibits improved generalization compared to the current state-of-the-art trackers. In particular, we exploit offline learning by introducing additive noise for input data augmentation to mitigate the overfitting problem. We propose feature fusion from noisy and clean input channels which improves the target localization. Channel attention integrated with our framework helps finding more useful target features resulting in further performance improvement. Our proposed IRCA-Siam enhances the discrimination of the tracker/background and improves fault tolerance and generalization. An extensive experimental evaluation on six benchmark datasets including OTB2013, OTB2015, TC128, UAV123, VOT2016 and VOT2017 demonstrate superior performance of the proposed IRCA-Siam tracker compared to the 30 existing state-of-the-art trackers.

Keywords: Siamese networks; convolutional neural network; visual tracking; noise regularization; attentional mechanism

1. Introduction

Visual Object Tracking (VOT) is a promising and fundamental research area in computer vision applications including robotics [1], video understanding [2], video surveillance [3] and autonomous driving [4]. Given the initial state of a target object (generally specified by a bounding box) in a video, the aim of an object tracker is to estimate the spatial trajectory of the target object in the upcoming frames. Despite a significant progress made in the field of VOT, it remains a challenging problem owing to diverse real-world challenges such as scale variations, occlusion, background clutter, fast motion, and illumination variations.

Deep trackers take the benefits from pretrained deep neural networks and have shown outstanding performance [5–10]. These deep trackers extract features from off-the-shelf pretrained models as a backbone feature extractor known as deep features for better discrimination. The pretrained models are trained over ImageNet [11] for image classification tasks such as VGGNet and AlexNet. Many computer vision sub-fields employ pretrained models to benefit from transfer learning [12,13]. However, it can be observed that during tracking, these models may not fully adapt the specific target features and online learning may steer to overfitting [14]. Recently, deep Siamese-based trackers [10,15–17] have become popular since they achieve good performance with relatively low

computational cost. Deep neural networks are composed of multiple hidden layers, which enable learning complex relationships between the inputs and outputs. However, due to limited training data, deep network models are prone to over-learn the training dataset which may lead to overfitting problem [18]. Dropout [18] and additive noise [19] can be employed to handle this issue in deep neural networks. There exists many approaches to handle overfitting problem by using spatio-residual modules [20], regularizers [5,21], integrating context information [22], or factorized convolution [23]. Another limitation is that online learning is an expensive process and requires more computational resources.

Our proposed methodology tackles aforementioned challenges by adopting an input regularization using similarity matching learning function. To validate the basic concept, we used SiameseFC [15] as our baseline tracker. It is important to improve the training process by including novel data augmentation techniques to enhance the generalization ability of deep trackers. We propose data augmentation by introducing noise into the training dataset. Introducing noise is similar to instructing a network not to change its output and it may be considered a special kind of input regularization. The proposed data augmentation method increases the accuracy and reduces the generalization error and overfitting problem.

Certain convolutional feature channels contribute more than the others, employing a channel attention mechanism can enhance the tracking performance. A channel attention mechanism is considered to be a process of weighting specific features because of their potential to model context information. The inclusion of attention mechanism has already been shown to be beneficial in visual object tracking. ACFN [24] used spatial attention to select a subset of correlation filters for tracking. RASNet [25] employs different kinds of attention models to highlight the dominance or weakness of channels. It is necessary to suppress irrelevant channels while providing higher weights to the more useful channels. Based on these observations, we incorporate a channel attentional mechanism within Siamese framework to enhance the tracking performance. We adapted feature fusion and a special kind of attention mechanism in our tracking framework to generate more discriminative target features.

In the current work, we propose an Input-Regularized Channel Attentional Siamese (IRCA-Siam) network to learn efficient target boosted features and enhance its discriminative ability. Early feature fusion is helpful for encoding adaptive target representations while suppressing noisy information. Moreover, the proposed network exploits the relationship among feature channels at a high level to learn informative and meaningful channels while suppressing trifling channels. The proposed tracker is evaluated over OTB2013 [26], OTB2015 [27], temple color-128 [28], UAV123 [29], VOT2016 [30], and VOT2017 [31] datasets and compared with 30 state-of-the-art methods. The proposed tracker has consistently shown improved performance compared to these trackers.

We summarize our main contributions as follows:

- We propose an additive noise as input regularization to improve deep network generalization.
- Early feature fusion mechanism is proposed to learn better target feature representation.
- An adaptive channel attention mechanism is integrated to give more weight to the important channels compared to the less important ones using a skip connection.
- Robustness of the proposed tracker is evaluated on the six benchmark datasets. Our experiments demonstrate better performance of the proposed tracker compared to the 30 state-of-the-art methods.

2. Related Work

In this section, we review different deep learning methods using additive noise. We also discuss closely related tracking approaches including deep features based trackers, Siamese-based and attention-based trackers. A detailed study may be found in recent surveys [6,9,32,33].

2.1. Deep Learning with Noise

Deep Neural Network (DNN) models have shown significant importance due to improved performance in various computer vision problems such as image classification, semantic segmentation,

and action recognition. However, due to limited training data, networks may lead to overfitting. Dropout is an often used method to handle overfitting issue by randomly dropping out values in the hidden units in the network model [18]. However, it is still unclear how to select the best dropout rate to perform well and how can we maximize the benefit from optimization as well as preventing model from overfitting [19]. Instead of using dropout, many researchers used additive noise to handle overfitting problem [19,34,35]. Increased dropout rate may cause information loss especially when target size is small and decreased dropout may not be able to avoid overfitting. Noh et al. [19] used additive noise as regularizer from marginalized noise instead of dropout approach. Bishop et al. [34] showed that additive noise effect is similar to Tikhonove regularization. Liu et al. [36] used noise layer to prevent their network from adversarial attacks. Fiaz et al. [6] studied the performance of trackers on noisy inputs during tracking. In contrast, we propose an additive noise as input regularization to improve the generalization error in the visual object tracking domain. Proposed regularization improves the tracking performance during the inference. We also verified the performance of our framework by inducing a noise layer before each convolutional layer. Experimental results showed that inducing a noise layer for each convolutional layer reduces the tracking performance compared to adding noise in the input data.

2.2. Deep Feature-Based Trackers

Recently, deep learning approaches have boosted the tracking performance due to their inherent characteristics. However, employing deep learning in visual tracking have several limitations. For example, deep learning requires more computational resources and have higher time complexity. The ground truth for the reference target object is provided only on the first frame of the video. To benefit from deep learning and limited available training data, deep features are combined into correlation filter tracking to boost the tracking performance. For instance, DeepSRDCF [5], CF2 [8], and FCNT [37] take the leverage from deep learning by extracting deep features at multiple layers from pretrained models such as VGG [38] or AlexNet [39]. Deep features from different layers were exploited to enable the capabilities of accuracy and robustness for the visual tracking [7,23,40,41]. Bhat et al. [41] revealed that pretrained models do not always fetch performance boost due to incompatible resolutions, unseen target objects and increasing dimensions. On the other hand, deep learning can also be used as classification or regression networks for visual tracking [22,42,43]. CNN-SVM [44] employs CNN model and performs classification task using SVM with saliency map. The TSN tracker [45] used CNN to encode temporal and spatial information for classification. The MDNet [21] is a multi-domain online deep tracker performing tracking as classification task, and capturing the domain dependent information during online tracking within a particle filter framework.

The online model update is performed to adapt different appearance variations of the target, but it may lose target under scenarios such as occlusion, deformation, or background clutter. Online learning requires extra computational cost to update the model parameters. Although CNN-based models have fewer parameters than RNN-based models, frequent model update incur extra computational cost therefore, such trackers may have limited real-world applications.

2.3. Siamese Network-Based Trackers

A Siamese network comprises of two parallel Convolutional Neural Networks (CNN) streams that are used to learn the similarity between input images in embedded space and to fuse them to produce an output [46]. Owing to their inherent characteristics such as accuracy and speed, Siamese networks are popular in the visual tracking community [10,15–17,47]. A SiameseFC [15] extracts input image features using an embedded CNN model and fuses them by using a correlation layer, to generate a response map. CFNet [10] is an improved version of the SiameseFC and it integrates a correlation filter layer as a differentiable layer within template branch. On the other hand, GOTURN [16] involves the use of a Siamese network as a feature extractor and the use of fully connected layers for fusing embedded features. The GOTURN tracker performs regression between two consecutive frames.

The SINT [17] formulates the tracking problem as a verification task to learn the similarity between inputs. These approaches have secured much importance due to their performance, but overfitting might occur if trained on small datasets. The proposed tracking algorithm enhances the discriminative ability of Siamese tracking framework by exploring data augmentation using additive noise.

2.4. Attention Mechanism-Based Trackers

Recently, attention mechanisms have become popular owing to their improved learning capabilities. CSRDCF [48] constructs a unique spatial reliability map to impose constraints on correlation filters within a correlation tracking framework. AFS-Siam [49] selects the discriminative kernels from different convolutional layers. Choi et al. [24] proposed ACFN and used spatial attention to select a subset of correlation filters for visual object tracking. RTT [50] used multi-directional recurrent filters to learn the target object appearance. The objective of using a channel attention mechanism has enabled the tracker to learn the most critical information to adapt the target appearance. However, the attention mechanism within convolutional layers has not been fully exploited. On the basis of these considerations, we introduced a channel attention mechanism to highlight the importance of discriminative features. Our technique showed high performance by offline learning efficient discriminative features.

3. The Proposed Input-Regularized Channel Attentional Siamese (IRCA-Siam) Network

Overall framework of the proposed IRCA-Siam network is shown in Figure 1. Compared to the previous deep trackers, IRCA-Siam exploits additive noise in the input data within Siamese framework to handle overfitting problem. We propose an early feature fusion mechanism for better target localization. We also integrate a channel attention mechanism within IRCA-Siam to highlight the more useful and discriminative features for improved tracking.

3.1. Fully Convolutional Siamese Network

The building block of the proposed framework is SiameseFC tracker proposed by Bertinetto et al. [15]. SiameseFC formulates the tracking problem, to learn a similarity map from embedded CNN models, as a cross-correlation problem within a Siamese network architecture. The embedded CNN model consists of two parallel branches, one representing the target and the other representing the search region. In visual tracking, the target template is provided in the first frame of the as an exemplar z. The objective of SiameseFC is to find the most similar region from the search region (larger in size than the template) x for subsequent frames as:

$$g(z, x) = \theta(z) * \theta(x) + b, \tag{1}$$

where $*$ represents the cross-correlation, $\theta(\cdot)$ denotes the embedded space, and b represents the offset of the similarity value. From Equation (1), we note that SiameseFC uses feature representation and discriminative learning to produce a similarity map by using a single function $\theta(\cdot)$. The performance of both tasks may lead to overfitting the model to the training data. We therefore propose noisy regularized feature fusion to overcome the challenges faced by SiameseFC and to improve the generalization capability of the tracker. We also highlight the importance of discriminative channel feature information.

3.2. Input Regularization and Feature Fusion

In the current study, a data augmentation mechanism is introduced for Siamese networks to overcome their limitations. Existing Siamese trackers suffer due to low fidelity of the target representation. We propose an input regularization during the training of Siamese trackers. Introducing noise into the input can be regarded as input regularization, and it encourages the model to learn various aspects of the object and increases its robustness against noise during testing. The features from both branches are fused (as shown in Figure 1) such that the model can learn the

target features under noise or disturbance to enhance its accuracy in real-world noisy environment. It may be noted that during tracking, a target may observe noise leading to performance degradation. The proposed feature fusion mechanism helps to overcome this limitation.

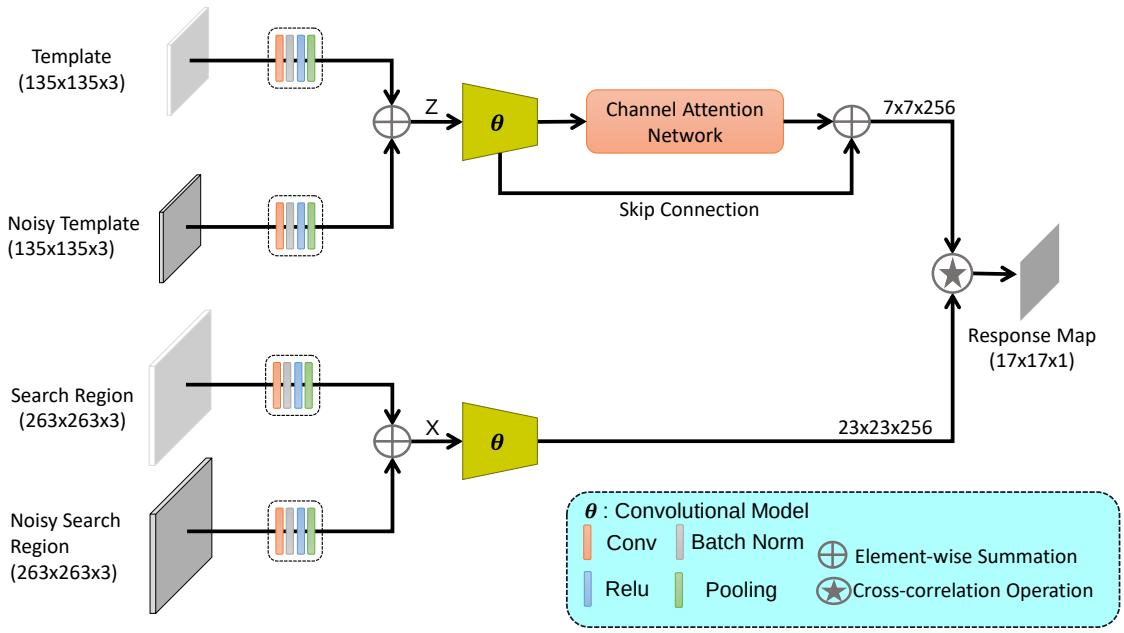


Figure 1. Proposed IRCA-Siam tracking framework. The inputs are fused after MaxPool layer for exemplar and search branches. Channel attentional network is integrated for exemplar branch using a skip connection.

We induce random Gaussian noise into the input patches to obtain noisy images with mean μ and standard deviation σ. A Gaussian noise map $G \hookrightarrow \texttt{Rand}_G(\mu, \sigma^2)$ is constructed and added with the input, where $\texttt{Rand}_G(\cdot)$ is a random number generator function based on Gaussian density function. In contrast to existing Siamese networks, the proposed model accepts four inputs, namely a target patch (z), a noisy target patch ($G + z$), a search patch (x), and a noisy search patch ($G + x$). Low-level features from noisy and clean images are fused to encode the spatial target information for better localization.

In practice, we fuse features from target patch and noisy target patch as:

$$Z = \mathcal{B}(z) + \mathcal{B}(G + z), \tag{2}$$

where $\mathcal{B}$ represents a convolutional block including a convolutional layer, a normalization layer, a rectifier layer, and a pooling layer. Similarly, features from search and noisy search patches are fused as:

$$X = \mathcal{B}(x) + \mathcal{B}(G + x), \tag{3}$$

The proposed framework is summarized as:

$$g(z, x) = (\Delta(\theta(Z)) \oplus \theta(Z)) * \theta(X) + b, \tag{4}$$

where $\Delta(.)$ denotes the channel attention and $\oplus$ represents the element-wise addition operation. The channel attention network is explained in Section 3.3.

During testing, we do not require noisy template and noisy search region. Instead, we provide the same template and search region that are provided to the other two inputs.

3.3. Channel Attention Network

A convolutional feature channel can be considered to be equivalent to a specific type of visual pattern. SiameseFC treats the feature channels for both the exemplar and search branches equally, which leads to performance degradation. However, the proposed channel attention mechanism exploits the relationship among channels and assigns more weights to channels that contribute more to target discrimination and localization. The objective is to enhance the adaptation capacity of the model to capture target variations. We incorporate a channel attention mechanism in the template branch as shown in Figure 1. There exists many channel attentional networks to calibrate the channel information such as SENet [51] and SA-Siam [52] which employ only global max-pooling and multi-perceptron layer. Choi et al. [24] proposed ACFN and used spatial attention to select a subset of correlation filters for visual object tracking. On the other hand, our channel network fuses the channel coefficients from global max-pooling and global average pooling and then forwards to convolutional layer. The global max-pooling exploits the finer and distinctive target information while global average pooling reflects the overall knowledge of the target for proposed channel attention.

The proposed channel attention mechanism is a lightweight network, as depicted in Figure 2. The input for this network is the output features $\theta(z)$ from the last convolutional layers. This network passes the inputs to Global Average Pooling (GAP) and Global Maximum Pooling (GMP) layers. The outputs of these layers are fused using an element-wise operation to form a Global Descriptor (GD). The GD is feed forwarded to a dimensionality reduction layer, a rectifier activation layer, and a dimensionality increasing layer and then relayed to a Sigmoid activation layer to provide the final weights of the input features.

The input to the channel attentional mechanism is represented as $C = \theta(Z)$ from Equation (4). The Global Descriptor (GD) is calculated using element-wise operation ($\oplus$) between the outputs from $\texttt{GAP}$ and $\texttt{GMP}$ layers as:

$$GD = \texttt{GAP}(C) \oplus \texttt{GMP}(C). \tag{5}$$

The weights for input features are computed as:

$$\alpha = \sigma(\texttt{fc}_2(\texttt{Relu}(\texttt{fc}_1(GD))))), \tag{6}$$

where $\texttt{fc}_1$ and $\texttt{fc}_2$ denote fully connected layers, $\texttt{Relu}$ represents rectifiers layer, and σ is the Sigmoid function as $f(x) = \frac{1}{1+e^{-x}}$. It is assumed that C has k feature channels such that $C = [c_1, c_2, ...c_k]$.

$$\hat{c_k} = \alpha_k \times c_k, \tag{7}$$

where α_k represents the kth weight for channel c_k. Then the final output of channel attention will be $\Delta(C) = \Delta(\theta(Z)) = [\hat{c_1}, \hat{c_2}, ...\hat{c_k}]$.

The output of proposed channel attention element-wise is added to the $\theta(Z)$ using skip connection as shown in proposed framework Figure 1. Proposed channel attention is only applied in the template branch of our framework to exploit the target feature channels.

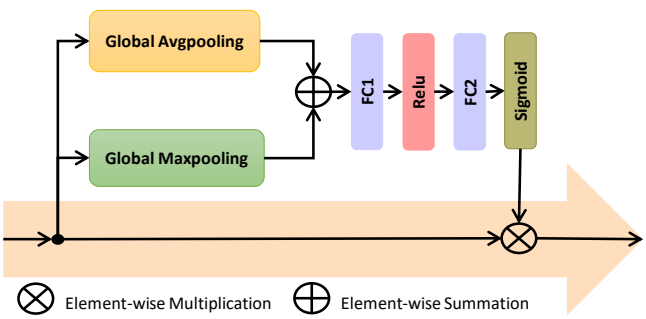


Figure 2. Channel attention network.

4. Experiments

4.1. Implementation Details

We train proposed model over GOT-10K dataset [53] which contains more than 10,000 video sequences. The proposed network accepts four input image patches. During offline training, the input size for the template and noisy template is $127 \times 127 \times 3$, while that for search region and noisy search region is $255 \times 255 \times 3$. For noisy images, μ is fixed at zero and σ is set to 0.09 which is obtained empirically and discussed in Section 4.3. During data curation, we crop the input patches such that the target object resides at the center as it reflects the most influential region for tracking performance. During training, we regularize our input using Gaussian additive noise such that it refrains to distract against noise at inference time. The model was trained offline end-to-end using a stochastic gradient method for 50 epochs. We set the momentum to 0.9 and the weight decay to 5×10^{-4}, while the learning rate started at 10^{-2} and later decreased to 10^{-5}. During training, we adopt the following loss function to update the model parameters:

$$L(g, y) = \frac{1}{|\delta|} \sum_{(k) \in \delta} log(1 + exp(-g(k) \times y(k)), \tag{8}$$

where g represents the response map, $y \in \{+1, -1\}$ denotes ground-truth label, k shows the position in the response, and δ indicates the set of positions in the search window on the score map.

During testing, we set template and noisy template is $135 \times 135 \times 3$, while that for search region and noisy search region is $263 \times 263 \times 3$. During the inference, the maximum location on the response map represents the new estimated target location. To overcome the problem of scale variations, we constructed a pyramid over three scales $(0.963, 1, 1.0375)$ based on previously estimated location for the current frame and selected the best score for target scale estimation. The code was implemented in python 3.7 and PyTorch 1.0.1 and all the experiments were performed using 1 GPU NVIDIA TITAN X_p over i7 3.6GHz CPU (PRIME Z370-A II) with 32G memory.

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

An extensive experimental evaluation is performed for six datasets including Object Tracking Benchmark 2013 (OTB2013) [26], OTB2015 [27] TempleColor128 (TC-128) [28], UAV123 [29], VOT2016 [30], and VOT2017 [31]. OTB2013 [26] comprises 50 different challenging videos, while OTB2015 [27] is an extended version containing 100 sequences. TC-128 [28] contains 128 colored challenging sequences. UAV123 contains 123 videos captured from Unmanned Aerial Vehicle (UAV) at a low-altitude [29]. Precision and success metrics were used to perform a comparison for aforementioned datasets. The precision is computed using the Euclidean distance between the ground-truth center and the predicted center as:

$$P_{gp} = \sqrt{(x_g - x_p)^2 + (y_g - y_p)^2}, \tag{9}$$

where (x_g, y_g) denote the ground-truth center location, and (x_p, y_p) represent the predicted target center position in a frame. A frame is considered successful if the precision is within a threshold of P_{gp} which is set P_{gp} equal to 20 pixels in the current work. Similarly, success is determined from the overlap score between the ground-truth bounding box r_g and the predicted bounding box r_t as:

$$OS = \frac{|r_t \cap r_g|}{|r_t \cup r_g|}, \tag{10}$$

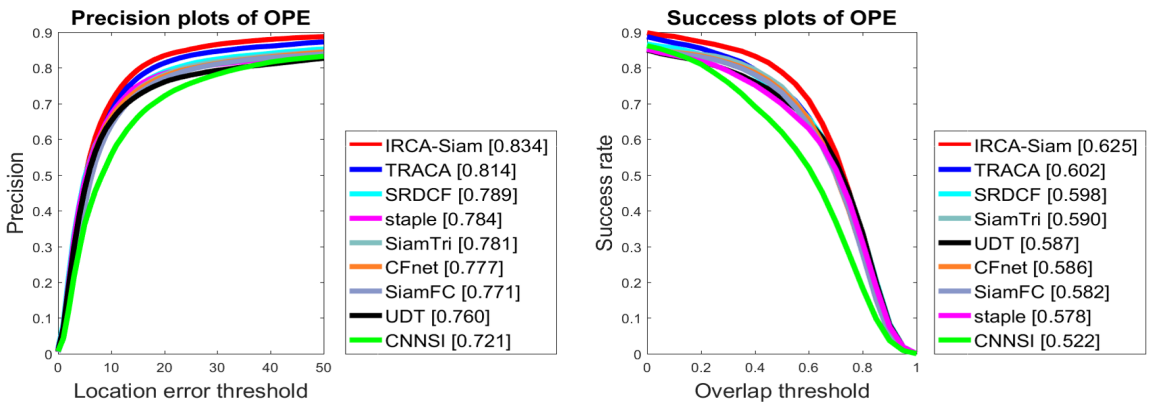
where $|\cdot|$ indicate the number of pixels, $\cap$ shows the intersection of two regions while $\cup$ indicates the union of two regions. If the overlap score (OS) exceeded 0.5, the frame is classified as having been tracked successfully; otherwise, the tracking is classified as failure. We performed One Pass Evaluation (OPE) to

validate our tracking method [9]. We also performed evaluation over VOT2016 [30] and VOT2017 [31]. The tracker is re-initialized during the evaluation if it encounters failure. We used the Expected Average Overlap (EAO), Accuracy (A), and Robustness (R) parameters for the evaluation for VOT2016 and VOT2017 datasets. Accuracy represents the average overlap score between estimated bounding box and ground truth. Robustness means the number of times a tracker failed. EAO computes the expected overlap score for typical short-term sequence lengths over an interval by averaging the scores for the expected average overlap curve [54].

We compared our method with 30 state-of-the-art trackers including SiamTri [55], CSRDCF [48], CNNSI [56], SRDCF [57], Staple [58], TRACA [59], SiameseFC [15], CFNet [10], ACFN [24], SiamFc-lu [60], HASiam [61], SiamFCRes22 [62], Kuai et al. [63], MSN [64], MLT [65], KCF [66], SCT [67], OA-LSTM [68], ECOhc [23], DSiam [69], MEEM [70], CCOT [40], SAMF [71], CMKCF [72], SATIN [73], GradNet [74], SiameseRPN [75] DSST [76], MemTrack [14], MemDTC [77], and UDT [78].

4.2.1. Evaluation over OTB Datasets

We present precision and success plots for the OTB2015. We compared IRCA-Siam with other state-of-the-art methods including TRACA, SRDCF, staple, SiamTri, CFNet, SiamFC, UDT, and CNNSI. Figure 3 demonstrates that the proposed algorithm IRCA-Siam showed better tracking performance compared to other trackers. IRCA-Siam achieved 62.5% and 83.5% success and precision respectively, which is 3.9% and 6.3% gain in performance compared to baseline SiamFC tracker. We compared our method with Siamese-based trackers including SiamTri, SiameseFC, CFNet, UDT, and CNNSI as shown in Figure 3. These tracking approaches take two inputs, but our approach takes four inputs. During training, we train our model such that it withholds discriminative ability for better localization. Our method has achieved 2.1% and 2.3%, 4.5% and 2.7%, and 5.3% and 4.7% superior performance in terms of precision and success, respectively, compared to correlation filter-based trackers such as TRACA, SRDCF, and Staple, respectively.



(**a**) Precision plot over OTB2015. (**b**) Success plot over OTB2015.

Figure 3. Performance comparison over OTB2015.

We also present the success scores for OTB2013 and OTB2015 in Table 1. The table also displays the average speed in units of Frames Per Second (FPS). The table shows that MSN [64] and HASiam [61] achieved success score more than 63.0 for OTB2013. Compared to these trackers, IRCA-Siam secured superior success score of 65.3. We also observed that our algorithm surpassed the other methods over OTB2015. Futhermore, our algorithm performs tracking at 77 FPS and is a real-time tracker. Although TRACA [59], SiamTri [55], Staple [58], SiamFC-lu [60], and SiameseFC [15] show higher tracking speed than our algorithm, they are less successful for OTB2013 and OTB2015.

Table 1. Performance comparison of IRCA-Siam with other trackers over OTB2013 and OTB2015 using success and speed in FPS.

Tracker	OTB2013	OTB2015	FPS	Real-Time
TRACA [59]	65.2	60.3	**101**	Yes
SiamTri [55]	61.5	59.0	85	Yes
CSRDCF [48]	59.9	58.2	24	No
ACFN [24]	60.7	57.5	15	No
CNNSI [56]	53.9	52.2	<1	No
SRDCF [57]	62.6	59.8	6	No
Staple [58]	59.3	57.8	80	Yes
SiamFc-lu [60]	-	62.0	82	Yes
HASiam [61]	64.0	61.1	30	Yes
Kuai et al. [63]	-	62.2	25	No
MSN [64]	64.3	59.7	40	Yes
MLT [65]	62.1	61.1	48	Yes
SiameseFC [15]	60.7	58.2	86	Yes
CFNet [10]	58.9	58.6	43	Yes
UDT [78]	61.9	58.7	70	Yes
IRCA-Siam	**65.3**	**62.5**	77	Yes

4.2.2. Challenge-Based Comparison

We present the evaluation of IRCA-Siam for various tracking challenges and compared with other state-of-the-art methods including TRACA, SRDCF, staple, SiamTri, CFNet, SiamFC, UDT, and CNNSI over OTB2015 in terms of success and precision in Figures 4 and 5 respectively. IRCA-Siam showed the best performance over fast motion, motion blur, deformation, in-planar rotation, out-of-planar rotation, occlusion, illumination variations, and scale variations challenges in terms of success. IRCA-Siam did not perform well over low-resolution videos and background clutter but ranked second with a minor difference as shown in Figure 4. SiamTri and TRACA surpassed our method with less than 1.0% for low-resolution and background clutter. However, overall, our tracker performed best for most of the challenges in terms of success.

We present precision plots for different challenges in Figure 5. Our algorithm showed better performance for eight challenges including fast motion, scale variations, illumination variations, occlusion, deformation, motion blur, in-plane rotation and low resolution. IRCA-Siam showed second best performance for out-of-view, low resolution, and background clutter. However, the difference between the top ranked compared with our method is less than 1.0%. As our approach ranked best for the rest of the challenges, such a minor difference can be ignored. We notice that other Siamese-based trackers are trained from raw images and do not perform well against different challenges. However, we train our model with regularized input such that it preserves the discriminative ability for better localization against noise during test time. This approach helped our method to perform better for most of the challenges in terms of both success and precision as shown in Figures 4 and 5 respectively.

4.2.3. Qualitative Analysis

We performed the qualitative analysis of the proposed method over *CarScale*, *FaceOcc1*, *Skiing*, and *Jogging-1* sequences as shown in Figure 6. In *CarScale* sequence, IRCA-Siam performed better compared to others as its bounding box enclose most region of the vehicle while others less. Almost all the trackers tackled the *FaceOcc1* sequence successfully. However, IRCA-Siam and TRACA succeeded to track the skier in *Skiing* sequence. The proposed method also performed efficiently for occlusion in *Jogging-1* sequence.

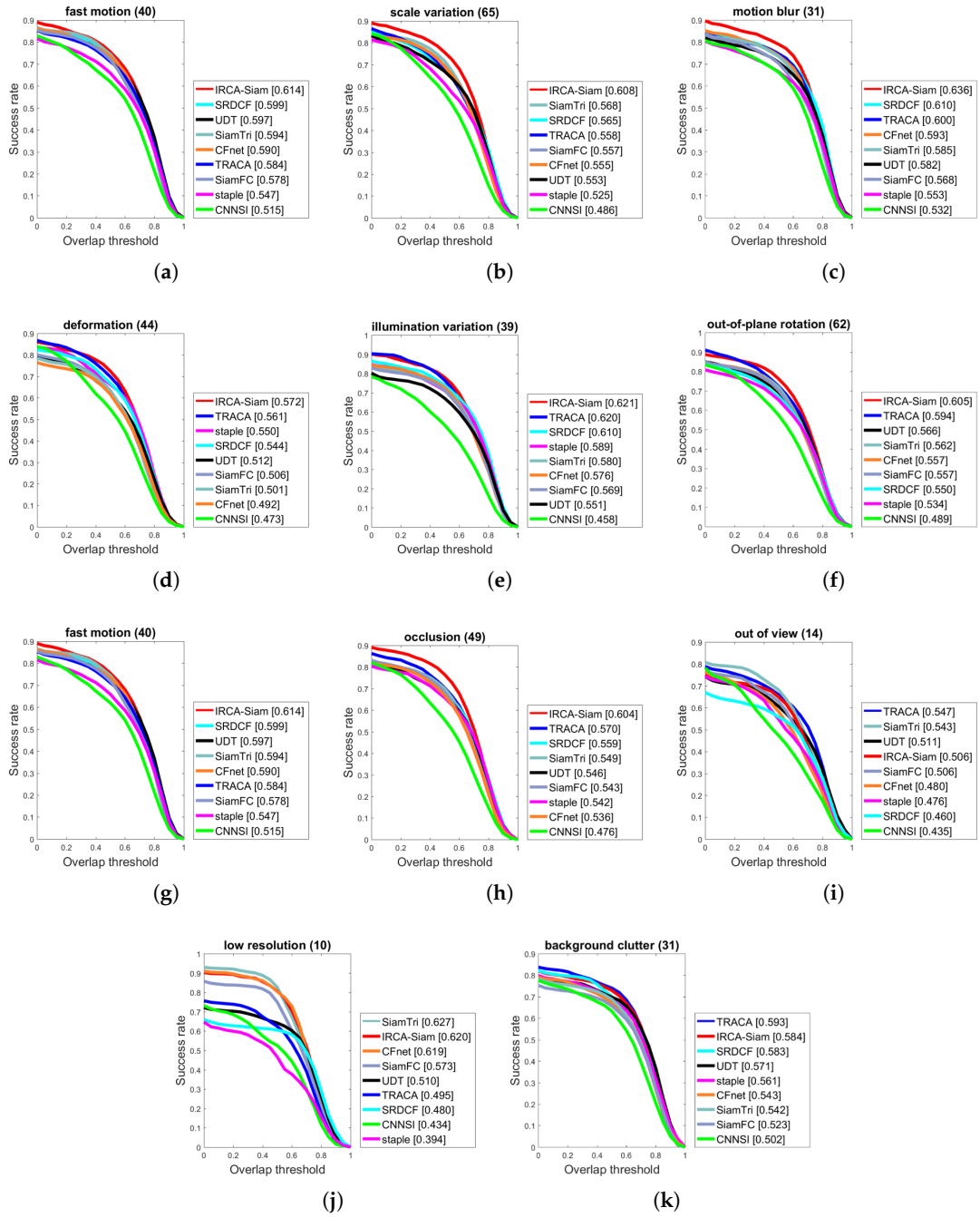


Figure 4. Success plots over OTB2015 for different challenges such as (**a**) fast motion, (**b**) scale variation, (**c**) motion blur, (**d**) deformation, (**e**) illumination variation, (**f**) out-of-plane rotation, (**g**) fast motion, (**h**) occlusion, (**i**) out-of-view, (**j**) low resolution, and (**k**) background clutter.

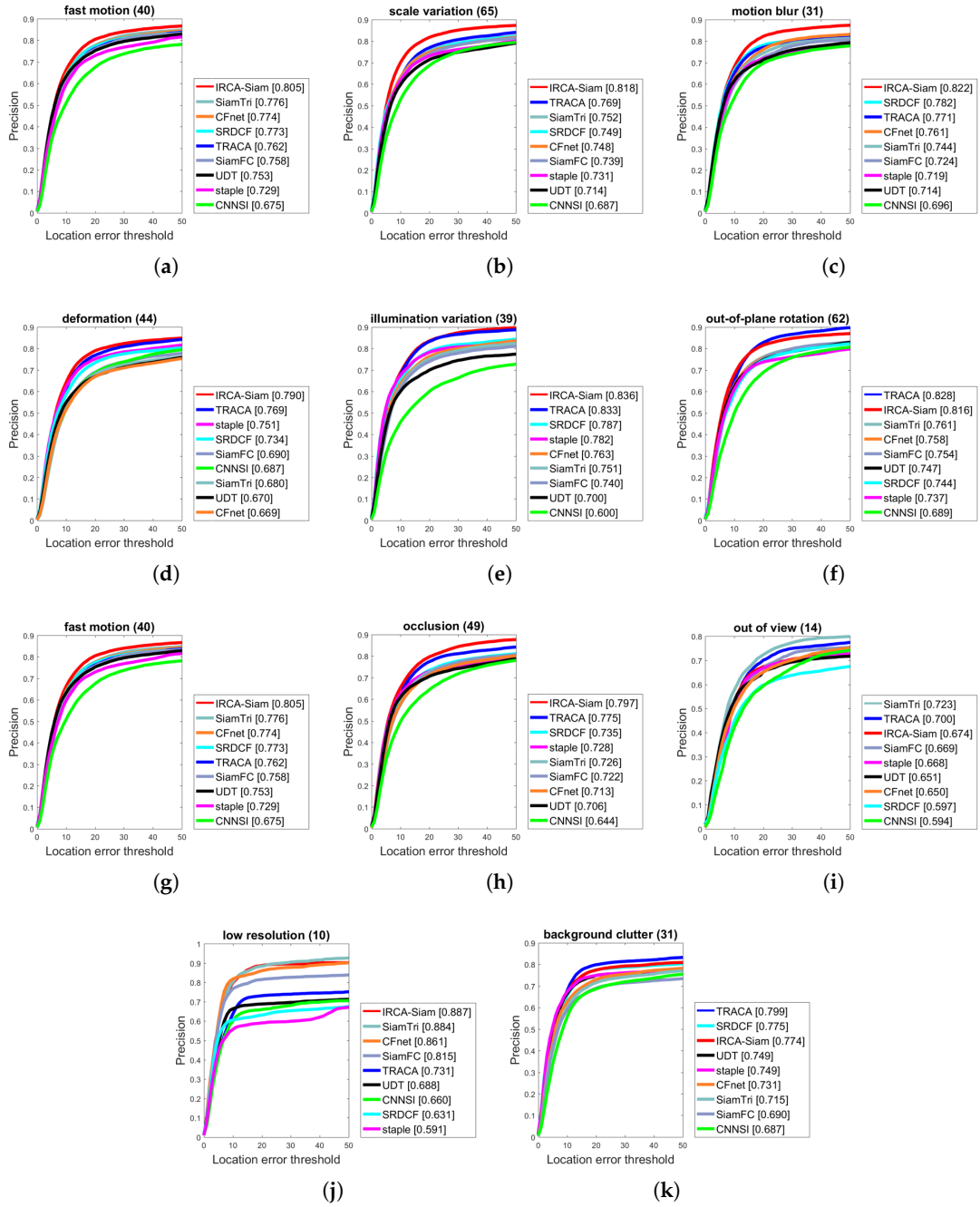


Figure 5. Precision plots over OTB2015 for different challenges such as (**a**) fast motion, (**b**) scale variation, (**c**) motion blur, (**d**) deformation, (**e**) illumination variation, (**f**) out-of-plane rotation, (**g**) fast motion, (**h**) occlusion, (**i**) out-of-view, (**j**) low resolution, and (**k**) background clutter.

4.2.4. Evaluation over TC128 Dataset

We validate the proposed IRCA-Siam tracker over TC128 benchmark dataset and showed the precision and success in Table 2. We compared our method with UDT [78], Kuai et al. [63], KC [66], MLT [65], SCT [67], SiameseFC [15], CFNet [10], Staple [58], CNNSI [56], OA-LSTM [68], and SRDCF [57]. The proposed method secured the first rank compared to other trackers with maximum precision score 74.5 and success 55.0.

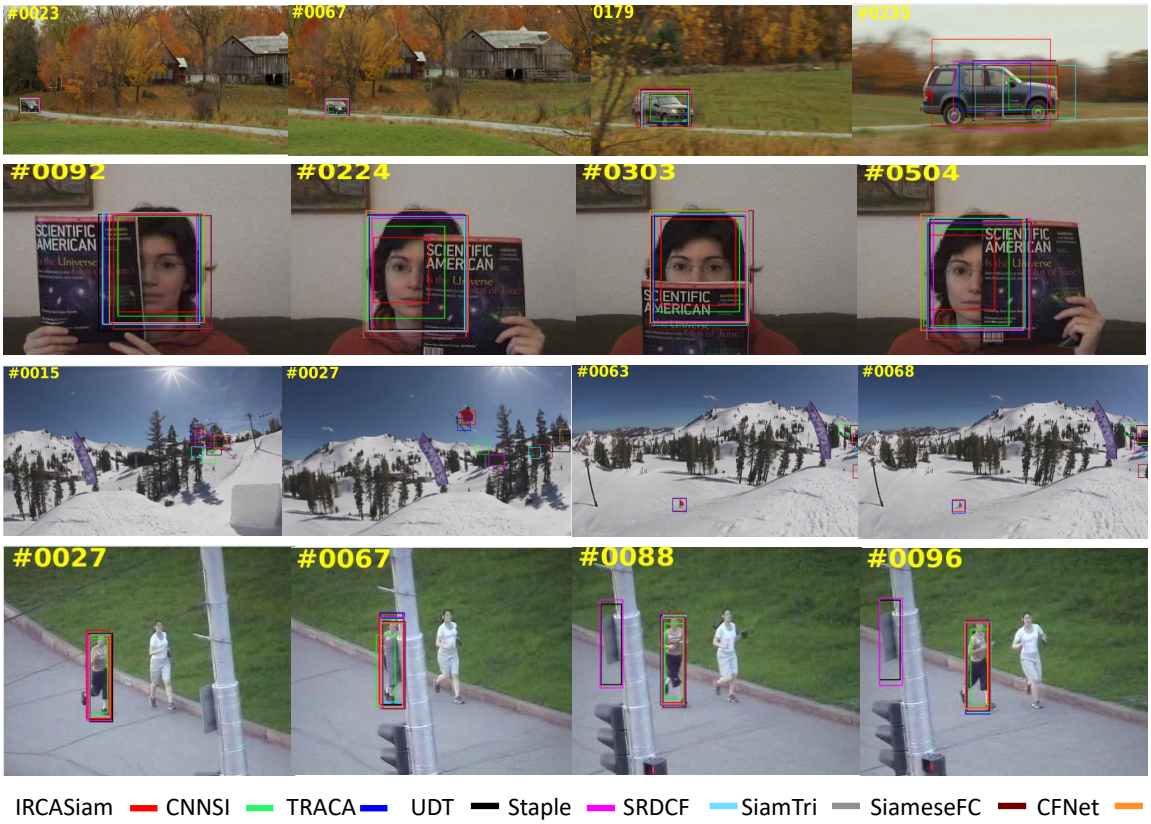


Figure 6. Qualitative analysis over *CarScale*, *FaceOcc1*, *Skiing*, and *Jogging-1* sequences.

Table 2. Comparison of the proposed method with various state-of-the-art methods over TC128 using precision, success and speed in FPS.

Trackers	Precision	Success	FPS
UDT [78]	71.7	50.7	70
Kuai et al. [63]	71.6	52.3	25
KCF [66]	54.9	38.7	160
MLT [65]	-	49.8	48
SCT [67]	62.7	46.6	40
SiameseFC [15]	68.8	50.3	86
CFNet [10]	60.7	45.6	43
Staple [58]		49.8	80
CNNSI [56]	63.8	44.8	<1
OA-LSTM [68]	70.8	49.5	11.5
SRDCF [57]	-	50.9	6
IRCA-Siam	74.5	55.0	77

4.2.5. Evaluation over UAV123 Dataset

This benchmark contains 123 videos captured from an Unmanned Aerial Vehicle (UAV) at a low-altitude. We opted to validate the proposed method over UAV123 dataset and showed the precision and success in Table 3. We compared IRCA-Siam with trackers including MLT [65], Kuai et al. [63], KCF [66], SRDCF [57], ECOhc [23], MEEM [70], SAMF [71], and DSST [76]. The results showed that IRCA-Siam demonstrated outstanding performance compared to other methods and secured best precision 74.5 and success 52.0.

Table 3. Comparison of the proposed method with various state-of-the-art methods over UAV123 using precision and success.

Trackers	Precision	Success
MLT [65]	-	43.5
Kuai et al. [63]	73.0	50.9
KCF [66]	54.9	38.7
SRDCF [57]	67.7	46.4
ECOhc [23]	72.5	50.6
MEEM [70]	62.7	39.2
SAMF [71]	59.2	39.6
DSST [76]	58.6	35.6
IRCA-Siam	74.5	52.0

4.2.6. Evaluation over VOT2016 and VOT2017 Dataset

We present the performance comparison over VOT2016 and VOT2017 in Tables 4 and 5 respectively. We compared our method over VOT2016 with various state-of-the-art trackers such as MemTrack [14], MemDTC [77], ECO [23], HASiam [61], Staple [58], SRDCF [57], DSiam [69], MLT [65], CCOT [40], UDT [78], SiameseFC [15], CMKCF [72], and SiamFCRes22 [62]. We observe that CCOT [40] secured best EAO 0.33 but our IRCA-Siam algorithm showed better accuracy and robustness for VOT2016 dataset. CMKCF [72] have shown lower robustness compared to our method but its accuracy is lower than ours. Moreover, our method showed best accuracy 0.56 compared to other state-of-the-art methods for VOT2016.

Performance comparison over VOT2017 is shown in Table 5. We compared our tracker with other trackers over VOT2017 dataset are CSRDCF [48], MemTrack [14], MemDTC [77], SRDCF [57], MSN [64], DSST [76], SATIN [73], SiameseFC [15], GradNet [74], SiameseRPN [75], and SiamFCRes22 [62]. We note that SATIN [73] showed best EAO score but its accuracy and robustness it not better than our algorithm. Furthermore, our algorithm showed best accuracy 0.52 and robustness 0.29 compared to other state-of-the-art algorithms.

Table 4. Performance comparison for different trackers over VOT2016.

Trackers	Overlap (↑)	Robustness (↓)	EAO (↑)
MemTrack [14]	0.53	1.44	0.27
MemDTC [77]	0.51	1.82	0.27
ECO [23]	0.54	-	0.37
HASiam [61]	-	-	0.27
Staple [58]	0.53	0.38	0.29
SRDCF [57]	0.54	0.42	0.25
DSiam [69]	0.49	2.93	0.18
MLT [65]	0.53	-	-
CCOT [40]	0.54	0.24	**0.33**
UDT [78]	0.54	-	0.22
SiameseFC [15]	0.53	0.46	0.23
CMKCF [72]	0.53	**0.18**	0.30
SiamFCRes22 [62]	0.54	0.38	0.30
IRCA-Siam	**0.56**	0.19	0.30

Table 5. Performance comparison for different trackers over VOT2017.

Trackers	Overlap (↑)	Robustness (↓)	EAO (↑)	FPS
CSRDCF [48]	0.49	0.49	0.25	13
MemTrack [14]	0.49	1.77	0.24	50
MemDTC [77]	0.49	1.77	0.25	40
SRDCF [57]	0.49	0.97	0.12	6
MSN [64]	0.50	0.46	0.26	40
DSST [76]	0.39	1.45	0.08	24
SATIN [73]	0.49	1.34	**0.28**	24
SiameseFC [15]	0.50	0.59	0.19	86
GradNet [74]	0.50	0.37	0.24	80
SiameseRPN [75]	0.49	0.46	0.24	200
SiamFCRes22 [62]	0.50	0.49	0.23	70
IRCA-Siam	**0.52**	**0.29**	0.25	76

4.3. Ablation Study

In this section, we investigate the effect of input additive noise and noise layers before convolution layers during the training. During testing, we neither provide input noise nor noise layers. We performed different experiments for SiameseFC and proposed (IR-Siam) method as shown in Figure 7. We also evaluated the performance of the proposed channel attention with additive noise named IRCA-Siam as shown in Figure 1. We performed our ablation study over OTB2015 dataset and showed the performance in precision and success.

In our framework, noise is added to inputs as regularization instead of dropout approach. Liu et al. [36] used noise layer to prevent their network from adversarial attacks. Therefore, we also used noise layer before convolutional layers to verify the improvement of generalization error using noise layer within convolutional model θ. We present the additive noise as input regularization as well as noise layer within Siamese tracking framework as shown in Figure 7. In Figure 7a shows the baseline SiameseFC tracking framework. We used a noise layer and placed before each convolutional layer to learn the noisy gradients during back propagation. Figure 7b presents the SiameseFC with noise layer before each convolutional layer. Similarly, Figure 7c,d represents the proposed framework without channel attention and, with and without noise layer, respectively. In our ablation study, we preformed different experiments to show the impact of addition of noise layer within Siamese framework.

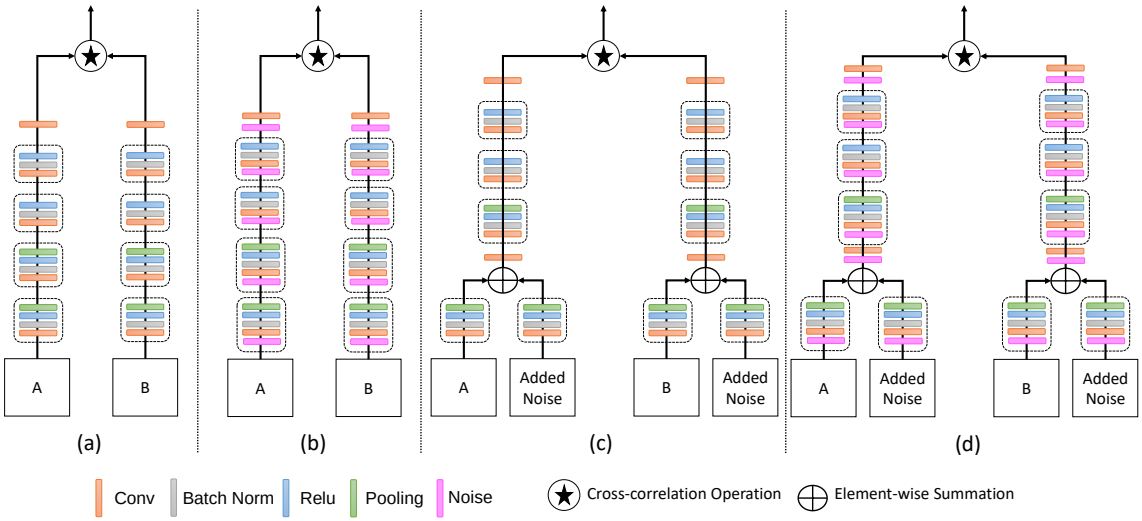


Figure 7. Added noise to inputs and different convolutional layers for SiameseFC and proposed framework. (**a**) shows the baseline SiameseFC, (**b**) indicates the SiameseFC with noise layers before convolutional layers, (**c**) represents the proposed framework without channel attention, and (**d**) shows the proposed framework with noise layers before convolutional layers and without channel attention.

First, we evaluate the performance of additive input noise. In this study, we used Salt and Pepper (S&P) and Gaussian noise as input noise. For S&P noise, we use three different probabilities (0.09, 0.05, and 0.03), similarly we use three different σ (0.09, 0.05, and 0.03) with mean (μ) zero for Gaussian input noise computation as shown in Figure 8. We observe that SiameseFC showed better performance without addition of noise. On the other hand, our IR-Siam without channel attention improved the tracking performance in the addition of Gaussian noise with σ = 0.09 and achieved precision = 81.9 and success = 61.9.

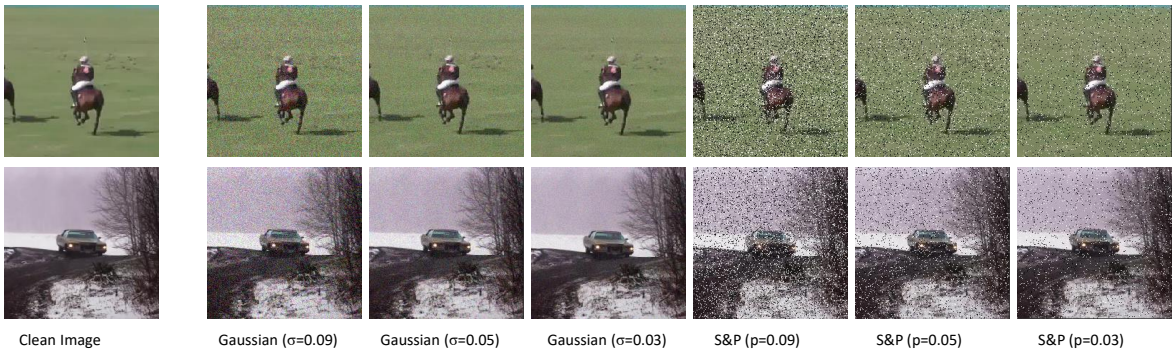


Figure 8. Illustration of additive noises to inputs. Here σ represents the variance of Gaussian noise while p denotes the probability for Salt and Pepper (S&P) noise.

We investigate the addition of input layers within the network architecture. We only added Gaussian noise layers before convolution layers as shown in Figure 7. We observe that the added noise layer degrades the performance for SiameseFC as well as our IR-Siam tracker. From Table 6, we note that IR-Siam shows the tracking improvement when noise is added as input. Moreover, we also find that the added channel attentional module shows tracking performance improvement. Proposed IRCA-Siam with channel attention achieved best precision = 83.4 and success = 62.5. The improved performance of IRCA-Siam reflects the importance of proposed channel attention network as it efficiently highlights the important feature channels and reduces the significance of the irrelevant ones.

Table 6. Ablation study performed over OTB2015 using precision and success.

Tracker	Additive Input Noise	Added Noise Layer before	Added Noise Layer Type	Precision	Success
SiameseFC	-	-	-	77.1	58.2
SiameseFC	S&P (p = 0.09)	-	-	76.5	57.2
SiameseFC	S&P (p = 0.05)	-	-	75.2	54.8
SiameseFC	S&P (p = 0.03)	-	-	73.5	52.9
SiameseFC	Gaussian ($\mu = 0, \sigma = 0.09$)		-	76.9	57.8
SiameseFC	Gaussian ($\mu = 0, \sigma = 0.05$)	-	-	75.7	56.4
SiameseFC	Gaussian ($\mu = 0, \sigma = 0.03$)	-	-	75.1	55.3
SiameseFC	-	Conv5	Gaussian ($\mu = 0, \sigma = 0.09$)	76.8	56.5
SiameseFC	-	Conv5	Gaussian ($\mu = 0, \sigma = 0.05$)	75.2	55.7
SiameseFC	-	Conv5	Gaussian ($\mu = 0, \sigma = 0.03$)	74.1	53.9
SiameseFC	-	Conv1, Conv2, Conv3, Conv4, Conv5	Gaussian ($\mu = 0, \sigma = 0.09$)	75.5	55.9
SiameseFC	Gaussian ($\mu = 0, \sigma = 0.09$)	Conv1, Conv2, Conv3, Conv4, Conv5	Gaussian ($\mu = 0, \sigma = 0.09$)	76.7	57.9
IR-Siam	-	-	-	80.8	60..6
IR-Siam	S&P (p = 0.09)	-	-	81.6	61.5
IR-Siam	S&P (p = 0.05)	-	-	80.3	61.0
IR-Siam	S&P (p = 0.03)	-	-	79.9	59.3
IR-Siam	Gaussian ($\mu = 0, \sigma = 0.09$)	-	-	81.9	61.9
IR-Siam	Gaussian ($\mu = 0, \sigma = 0.05$)	-	-	81.2	61.3
IR-Siam	Gaussian ($\mu = 0, \sigma = 0.03$)	-	-	80.1	60.4
IR-Siam	-	Conv6	Gaussian ($\mu = 0, \sigma = 0.09$)	80.9	60.6
IR-Siam	-	Conv6	Gaussian ($\mu = 0, \sigma = 0.05$)	80.2	60.1
IR-Siam	-	Conv6	Gaussian ($\mu = 0, \sigma = 0.03$)	78.9	58.7
IR-Siam	-	Conv1, Conv2, Conv3, Conv4, Conv5, Conv6	Gaussian ($\mu = 0, \sigma = 0.09$)	81.2	60.7
IR-Siam	Gaussian ($\mu = 0, \sigma = 0.09$)	Conv1, Conv2, Conv3, Conv4, Conv5, Conv6	Gaussian ($\mu = 0, \sigma = 0.09$)	80.5	59.5
IR-Siam	-	Conv1, Conv2, Conv6	Gaussian ($\mu = 0, \sigma = 0.09$)	81.5	60.5
IR-Siam	Gaussian ($\mu = 0, \sigma = 0.09$)	Conv1, Conv2, Conv6	Gaussian ($\mu = 0, \sigma = 0.09$)	82.1	60.8
IRCA-Siam	S&P (p = 0.09)	-	-	82.7	62.3
IRCA-Siam	Gaussian ($\mu = 0, \sigma = 0.09$)	-	-	83.4	62.5

5. Conclusions

In this work, input-noise-based regularization is proposed to improve tracking generalization. In addition, early feature fusion of noisy and clean channels is also proposed for better target localization. In the same framework, channel attention has been proposed to select more informative target features to improve tracking performance. For input-noise regularization, Gaussian noise has been added to both the template and the search patches during the training. Feature fusion is performed at low-level layers to make the tracking process more robust to noise and to improve target localization. Channel attention has been used to highlight more descriptive features and to suppress the noisy features. The proposed tracker has shown superior performance compared to 18 Siamese trackers and 12 other existing trackers. The proposed tracker has shown promising performance for fast motion, motion blur, deformation, in-plane rotation, out-of-plane rotation, occlusion, illumination variations, and scale variation challenges.

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

Acknowledgments: This study was supported by the BK21 Plus project (SW Human Resource Development Program for Supporting Smart Life) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (21A20131600005).

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. 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. Part A Syst. Hum.* **2016**, *47*, 1415–1427. [CrossRef]
2. Renoust, B.; Le, D.D.; Satoh, S. Visual analytics of political networks from face-tracking of news video. *IEEE Trans. Multimed.* **2016**, *18*, 2184–2195. [CrossRef]
3. 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]
4. Menze, M.; Geiger, A. Object scene flow for autonomous vehicles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015; pp. 3061–3070.
5. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV) Workshop, Santiago, Chile, 13–16 December 2015; pp. 58–66.
6. Fiaz, M.; Mahmood, A.; Jung, S.K. Tracking noisy targets: A review of recent object tracking approaches. *arXiv* **2018**, arXiv:1802.03098.
7. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q.; Lim, J.; Yang, M.H. Hedged deep tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4303–4311.
8. 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 Workshops (ICCV), Santiago, Chile, 13–16 December 2015; pp. 3074–3082.
9. 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*, 43. [CrossRef]
10. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.

11. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
12. Zhou, X.; Yao, C.; Wen, H.; Wang, Y.; Zhou, S.; He, W.; Liang, J. East: An efficient and accurate scene text detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5551–5560.
13. Jeon, M.; Jeong, Y.S. Compact and Accurate Scene Text Detector. *Appl. Sci.* **2020**, *10*, 2096. [CrossRef]
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. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
16. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.
17. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1420–1429.
18. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
19. Noh, H.; You, T.; Mun, J.; Han, B. Regularizing deep neural networks by noise: Its interpretation and optimization. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 5109–5118.
20. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.; Yang, M.H. Crest: Convolutional residual learning for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision(ICCV), Venice, Italy, 22–29 October 2017; pp. 2555–2564.
21. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4293–4302.
22. Mueller, M.; Smith, N.; Ghanem, B. Context-aware correlation filter tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1396–1404.
23. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
24. Choi, J.; Jin Chang, H.; Yun, S.; Fischer, T.; Demiris, Y.; Choi, J.Y. Attentional correlation filter network for adaptive visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4807–4816.
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 (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 4854–4863.
26. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 25–27 June 2013; pp. 2411–2418.
27. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE TPAMI* **2015**, *37*, 1834–1848. [CrossRef]
28. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef]
29. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 445–461.
30. Kristan, M.; Pflugfelder, R.; Lebeda, K. The Visual Object Tracking VOT2016 challenge results. In Proceedings of the European Conference on Computer Vision (ECCV) Workshop, Amsterdam, The Netherlands, 8–10 October 2016; pp. 777–823.

31. 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 (ICCV), Venice, Italy, 22–29 October 2017; pp. 1949–1972.
32. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep learning for visual tracking: A comprehensive survey. *arXiv* **2019**, arXiv:1912.00535.
33. Li, P.; Wang, D.; Wang, L.; Lu, H. Deep visual tracking: Review and experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338. [CrossRef]
34. Bishop, C.M. Training with noise is equivalent to Tikhonov regularization. *Neural Comput.* **1995**, *7*, 108–116. [CrossRef]
35. Rifai, S.; Glorot, X.; Bengio, Y.; Vincent, P. Adding noise to the input of a model trained with a regularized objective. *arXiv* **2011**, arXiv:1104.3250.
36. Liu, X.; Cheng, M.; Zhang, H.; Hsieh, C.J. Towards robust neural networks via random self-ensemble. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 369–385.
37. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision(ICCV), Santiago, Chile, 13–16 December 2015; pp. 3119–3127.
38. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 30, Lake Tahoe, CA, USA, 3–6 December 2012; pp. 1097–1105.
40. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488.
41. Bhat, G.; Johnander, J.; Danelljan, M.; Shahbaz Khan, F.; Felsberg, M. Unveiling the power of deep tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 483–498.
42. Han, B.; Sim, J.; Adam, H. Branchout: Regularization for online ensemble tracking with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3356–3365.
43. Yun, S.; Choi, J.; Yoo, Y.; Yun, K.; Young Choi, J. Action-decision networks for visual tracking with deep reinforcement learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2711–2720.
44. Hong, S.; You, T.; Kwak, S.; Han, B. Online tracking by learning discriminative saliency map with convolutional neural network. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 597–606.
45. Teng, Z.; Xing, J.; Wang, Q.; Lang, C.; Feng, S.; Jin, Y. Robust object tracking based on temporal and spatial deep networks. In Proceedings of the IEEE International Conference on Computer Vision(ICCV), Venice, Italy, 22–29 October 2017; pp. 1144–1153.
46. Fiaz, M.; Mahmood, A.; Jung, S.K. Deep Siamese Networks toward Robust Visual Tracking. In *Visual Object Tracking in the Deep Neural Networks Era*; IntechOpen: Rijeka, Croatia, 2019; ISBN 978-1-78985-157-1.
47. Rahman, M.M.; Fiaz, M.; Jung, S.J. Efficient Visual Tracking with Stacked Channel-Spatial Attention Learning. *IEEE Access* **2020**, *8*, 100857–100869. [CrossRef]
48. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.
49. 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 Proceedings of the International Workshop on Frontiers of Computer Vision, Kagoshima, Japan, 20–22 February 2020; pp. 167–179.
50. Cui, Z.; Xiao, S.; Feng, J.; Yan, S. Recurrently target-attending tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1449–1458.

51. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 7132–7141.
52. 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 (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 4834–4843.
53. Huang, L.; Zhao, X.; Huang, K. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *arXiv* **2018**, arXiv:1810.11981.
54. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; Pflugfelder, R. The visual object tracking vot2015 challenge results. In Proceedings of the IEEE International Conference on Computer Vision(ICCV) Workshop, Santiago, Chile, 13–16 December 2015; pp. 1–23.
55. 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.
56. Fiaz, M.; Mahmood, A.; Jung, S.K. Convolutional neural network with structural input for visual object tracking. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 1345–1352.
57. 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(ICCV), Santiago, Chile, 13–16 December 2015; pp. 4310–4318.
58. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1401–1409.
59. Choi, J.; Chang, H.J.; 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 (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 479–488.
60. Li, B.; Xie, W.; Zeng, W.; Liu, W. Learning to Update for Object Tracking With Recurrent Meta-Learner. *IEEE Trans. Image Process.* **2019**, *28*, 3624–3635. [CrossRef]
61. Shen, J.; Tang, X.; Dong, X.; Shao, L. Visual object tracking by hierarchical attention siamese network. *IEEE Trans. Cybern.* **2019**, *50*, 3068–3080. [CrossRef]
62. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
63. Kuai, Y.; Wen, G.; Li, D. Masked and dynamic Siamese network for robust visual tracking. *Inf. Sci.* **2019**, *503*, 169–182. [CrossRef]
64. Gao, M.; Jin, L.; Jiang, Y.; Guo, B. Manifold Siamese Network: A Novel Visual Tracking ConvNet for Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1612–1623. [CrossRef]
65. 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 (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 911–920.
66. 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]
67. Choi, J.; Jin Chang, H.; Jeong, J.; Demiris, Y.; Young Choi, J. Visual tracking using attention-modulated disintegration and integration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 4321–4330.
68. Du, Y.; Yan, Y.; Chen, S.; Hua, Y.; Wang, H. Object-Adaptive LSTM Network for Visual Tracking. In Proceedings of the 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 1719–1724.
69. 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 (ICCV), Venice, Italy, 22–29 October 2017; pp. 1763–1771.
70. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust tracking via multiple experts using entropy minimization. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 188–203.

71. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 254–265.
72. Huang, B.; Xu, T.; Jiang, S.; Chen, Y.; Bai, Y. Robust Visual Tracking via Constrained Multi-Kernel Correlation Filters. *IEEE Trans. Multimed.* **2020**. [CrossRef]
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.* **2019**, *2019*, 105448. [CrossRef]
74. 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(ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6162–6171.
75. 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 (CVPR), Salt Late City, UT, USA, 18–22 June 2018; pp. 8971–8980.
76. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef]
77. Yang, T.; Chan, A.B. Visual Tracking via Dynamic Memory Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]
78. 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 (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 1308–1317.