Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Multi-Model U-Net: An Adversarial Defense Mechanism for Robust Visual Tracking

Wattanapong Suttapak

wattanapong.su@outlook.com

Shanghai Jiao Tong University

Jianfu Zhang

Shanghai Jiao Tong University

Haohuo Zhao

Shanghai Jiao Tong University

Liqing Zhang

Shanghai Jiao Tong University

Research Article

Keywords: MU-Net, pixel-wise defense, feature-wise defense, adversarial defense, adversarial robust, visual object-tracking

Posted Date: March 15th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2675105/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Neural Processing Letters on April 1st, 2024. See the published version at https://doi.org/10.1007/s11063-024-11592-2.

Multi-Model U-Net: An Adversarial Defense Mechanism for Robust Visual Tracking

Wattanapong Suttapak[1*], Jianfu Zhang[1†], Haohuo Zhao[1†] and Liqing Zhang[1†]

[1*]Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dong Chuan Rd, Minhang, 200240, Shanghai, China.

*Corresponding author(s). E-mail(s): wattanapong_su@sjtu.edu.cn;
Contributing authors: c.sis@sjtu.edu.cn; haoh.zhao@sjtu.edu.cn; zhang-lq@cs.sjtu.edu.cn;
†These authors contributed equally to this work.

Abstract

Currently, state-of-the-art object-tracking algorithms are facing a severe threat from adversarial attacks, which can significantly undermine their performance. In this work, we present the first defensive model for visual tracking, called MU-Net. The proposed model generates denoised images that successfully defend against attacks while requiring minimal computing resources. To achieve this, MU-Net models are explored in five different models, with the smallest pair layers being 3. Each model is trained end-to-end on random pair images with both clean and adversarial noise images by utilizing pixel-wise and feature-wise denoisers. Our proposed models significantly enhance tracking performance even when the target tracker is attacked or the target frame is clean. Additionally, MU-Net can simultaneously share its parameters on both template and search regions. In experimental results, the proposed models successfully defend against top attackers on six benchmark datasets, including OTB100, LaSOT, UAV123, VOT2018, VOT2019, and GOT-10k. Performance results on all datasets show a significant improvement over all attackers, with a decline of less than 4.6% for every benchmark metric compared to the original tracker.

Keywords: MU-Net, pixel-wise defense, feature-wise defense, adversarial defense, adversarial robust, visual object-tracking.

1 Introduction

Visual tracking is a beneficial application in computer vision that aims to track the movement of objects within a scene in real-time. It has numerous applications, for example, video surveillance, robotics, and autonomous driving. Visual tracking typically involves detecting the location of an object in the first frame of a video and then tracking it through subsequent frames, such as SiamFC[1], SiamRPN[2], [3], SiamRPN++[4], ATOM[5], DiMP[6], and lighttrack [7]. However, this task is challenging due to factors such as changes in lighting conditions, occlusion, and background clutter.

In recent years, the performance of visual tracking models have been threatened by adversarial attacks, for example, FGSM [8], iFGSM [9], and miFGSM [10]. These attacks are designed to manipulate the input data and cause the model to malfunction, resulting in incorrect object tracking. Adversarial attacks are a growing concern in the field of machine learning, as they can undermine the accuracy and reliability of models in critical applications.

Visual tracking attacks involve generating adversarial noise that is added to the frames, causing the tracking model to fail [11–16]. The attackers use sophisticated tactics such as model generators to infiltrate the noise into the template frames or search frames. The perturbation caused by visual object-tracking (VOT) attackers is particularly alarming, and models such as CSA [13], FBA [14], and DFA [15] can generate real-time adversarial noise, leading to complete disruption of top trackers.

To address this challenge, researchers have proposed various countermeasures to improve the robustness of deep learning models against adversarial attacks [17, 18]. However, these approaches are often not effective in visual tracking due to fundamental differences in the network structures of classification and visual tracking models.

To overcome this limitation, we have proposed a novel defense model for visual tracking attackers, specifically designed to denoise the adversarial noise before feeding it to the tracking model. Our approach leverages the power of denoising techniques to remove adversarial noise and improve the quality of the input data. This helps to ensure that the tracking model receives clean and accurate data, making it more robust against adversarial attacks. We further propose a feature-wise denoiser on a foundation of deep learning, leveraging the power of convolutional neural networks (CNNs) for feature extraction and denoising. The proposed defense model is trained on a large dataset of clean and adversarial images. The model learns to distinguish between clean and adversarial images and to denoise the adversarial images to improve the performance of the visual tracking model. We evaluate the effectiveness of our

defense model on several benchmark datasets and demonstrate that it significantly improves the robustness of visual tracking models against adversarial attacks.

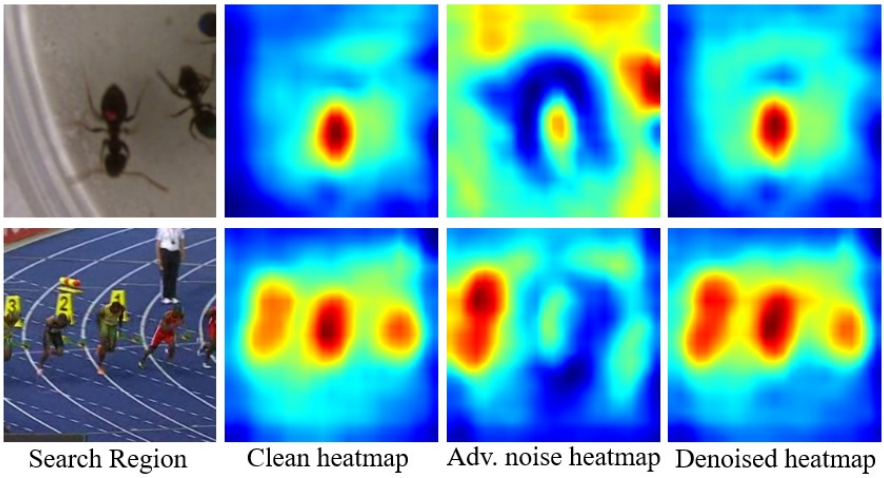


Fig. 1 Comparison tracking heatmap of clean images, adversarial noise images, and our denoised images.

Our proposed method's effectiveness is demonstrated in Fig. 1 through a comparison of search region frames and their corresponding heatmaps on a clean tracker, an adversarial noise attack, and our proposed defensive model. The contributions are described as follows:

1. We propose the first defensive method against visual tracking attackers by generating defensive noise and sharing parameters on both template and search frames.
2. We present MU-Net architectures, which generate denoised images to defend against adversarial attacks on visual tracking. Although MU-Net models provide rich defensive performance, they require subtle inference time due to the compact layers within the modified U-Net structures.
3. Our models achieve in defending VOT-threatening adversarial attacks on various benchmark datasets and are robust to whether the suspicious frame is clean or malicious.

2 Related Works

2.1 Visual-Tracking-Based Siamese Network

In the past decade, the field of object tracking has seen significant advancements in accuracy and speed. One of the earliest successful approaches was the Siamese fully convolutional network (SiamFC) [1] that learned the similarity between pairs of images, allowing it to track an object in a video sequence.

Expanding on the success of SiamFC, the Siamese region proposal network (SiamRPN) [2] improved the accuracy of the tracking algorithm by introducing a region proposal network [19] that generated candidate object locations.

SiamRPN++ [4] further improved upon SiamRPN by introducing a deeper and wider backbone network based on ResNet-50 [20], along with a multi-scale feature fusion mechanism to handle scale variations in the tracked object. Another extension to the Siamese architecture is the Siamese Box Adaptive Network (SiamBAN) that uses an attention mechanism to weight the feature maps based on their relevance to the object being tracked, thereby improving the accuracy of bounding box predictions. SiamMask [3] expanded the capabilities of the Siamese architecture to include instance segmentation in addition to object tracking. This was accomplished by incorporating a mask branch into the network that generated pixel-level segmentation masks for the tracked object.

To achieve real-time tracking on CPU devices, the Siamese Cascaded Architecture for Real-time object tracking (SiamCAR) [21] uses a lightweight feature extractor and a proposal generator to achieve high accuracy and speed. SiamFC++ [22] further extends the SiamFC architecture by upgrading head network structure and adding the regression structure to increase model accuracy. Moreover, they replace the AlexNet [23] backbone with GoogLeNet [24] backbone, which achieves superior performance with fewer computing resources.

In addition to these architecture-based approaches, One-Shot Neural Architecture Search (One-Shot NAS) is a method for automatically designing deep neural network architectures for visual tracking tasks. It optimizes the network architecture for a given tracking task using a reinforcement learning-based search strategy, resulting in a high-performance network with reduced manual effort.

2.2 Adversarial Attacks for Visual Tracking

Recently, adversarial attacks on VOT have received increasing attention in research as a means to anticipate against top trackers. Adversarial attacks are not limited to classification tasks, and in visual tracking, researchers have sought to generate imperceptible noise to disrupt object tracking. Recently, online learning attacks have been proposed that can successfully disturb top trackers, for example, Hijacking tracker [11], one-shot adversarial attacks [12], and IoU attack [16]. However, online training is impractical in visual tracking due to its time-consuming nature; for example, the IoU attack [16] takes 0.625 seconds per frame in inference time, as illustrated in Table 4. In offline training, Yan *et al.* [13] proposed offline training based on SiamRPN++ [4] with the GOT-10k dataset [25], named cooling-shrinking attack (CSA). This attacking model addressed a tracking weakness by decreasing the target heatmap density, misleading the predicted bounding box to be shrunken. The feature-based attack [14] and the diminishing-feature attack (DFA) [15] were introduced to attack the sole template frame by minimizing the variance of its feature

channel. This perturbed effect leads to tracker malfunction on both template and search frames. Apart from that, DFA perfectly transfers the attacking effect to unseen top trackers without learning from their parameters. It is explicit that adversarial attacking models can harm visual tracking models. Therefore, the adversarial defense against adversarial attack is essential to invest and urgently invent an adversarial defensive model.

We propose the first adversarial defense on visual tracking, named multi-model U-Net (MU-Net), to denoise the suspicious image, whether clean or malicious, before feeding it to the tracker. MU-Net models comprise five pair layers, such as 3, 4, 5, 6, and 7. Our MU-Net models can vary in parameter size and inference speed based on the number of pair layers. The defending performance of our largest model is outstanding, while our fewer-layer models are also efficient and very close to the most significant model.

3 Defensive Model Methodology

Our proposed model generates an adversarial denoised image by obtaining and preprocessing the suspicious frame, whether a clean or malicious image, before feeding it to the target tracker that we utilize SiamRPN++ [4]. This section describes defensive models, pixel-wise denoiser, and feature-wise denoiser. The pixel-wise denoiser obtains an adversarial noise image and generates a denoised image by removing the noise. The feature-wise denoiser uses the same principle but operates on the feature maps extracted by the SiamRPN++.

3.1 Pixel-wise and Feature-wise denoiser

Liao *et al.* [17] proposed a defensive model named pixel-guide denoiser and high-level guide denoiser that successfully improved image classification performance. In this work, we adapted their models to defend against adversarial attacks on visual tracking by recovering the original performance through the use of pixel-wise denoiser and feature-wise denoiser, PD and FD, respectively. Their loss functions can describe as follows:

$$\begin{aligned} L_{PD} &= \frac{\|x - \tilde{x}\|_2}{3 \times w \times h}, \\ L_{FD} &= \frac{\|f(x) - f(\tilde{x})\|_2}{M \times N \times O_1 \times O_2}. \end{aligned} \tag{1}$$

When an adversarial attacked image $x' = x + \Delta x$ for $x \in \mathbb{R}^{3 \times W \times H}$ is an input image, $\Delta x = G(x)$ is an adversarial noise of CSA [13] or DFA [15] when $G(x)$ is an adversarial noise generator. Consider a predicted noise generator function $D(x') = \Delta\tilde{x}$ that produces a predicted noise image $\tilde{x}$ as the output of a defensive model. Let $\tilde{x} = x' - \Delta\tilde{x}$ be the denoised image obtained by subtracting the predicted noise image $\Delta\tilde{x}$ from an adversarial image x'. In this approach, the defensive model attempts to generate a noise image that can

effectively remove the adversarial perturbations from the input image, thereby producing a denoised image $\tilde{x}$ that closely resembles the original image x.

In the pixel-wise denoiser, a predicted noise image $\Delta\tilde{x}$ can be generated by learning the noise pattern from the adversarial noise Δx. The pixel-wise loss function L_{PD} can be learned by minimizing the mean L2-norm of the difference between the clean input image x and the denoised image $\tilde{x}$. In the feature-wise denoiser, $f(\cdot) \in \mathbb{R}^{M\times N\times O_1\times O_2}$ denotes the adjusted convolution layers of SiamRPN++ based ResNet-50 [20]. M is the number of aggregated layers of SiamRPN++, N is the number of feature channels, and O_1 and O_2 are the width and height of each feature channel when $f(x)$ and $f(\tilde{x})$ are the SiamRPN++ features of the clean image x and the denoised image $\tilde{x}$. The feature-wise loss function L_{FD} aims to reshape the feature of the adversarial image to closely match the clean feature. The L_{FD} loss is expressed as the mean L2-norm of the difference between the feature of the clean image $f(x)$ and the feature of the denoised image $f(\tilde{x})$.

In the following section, network architecture will be explained more detail for the defensive model generator.

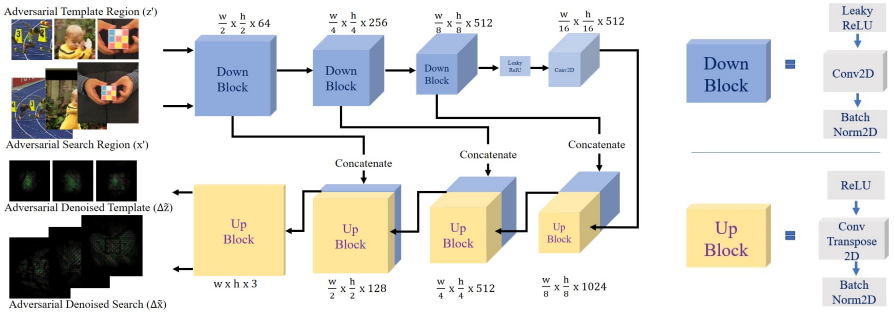


Fig. 2 Network architecture of our proposed MU-Net.

3.2 Network Architecture

The U-Net [26] model is utilized in many applications based on generative adversarial purposes, for example, image translation [27, 28], adversarial attack on visual tracking [13, 15], and adversarial defense on image classification [17]. This work is inspired by DUNET [17] due to its defending performance on the classification model. However, the inference time is crucial in visual tracking. We propose the MU-Net by modifying it from U-Net [26]. Our MU-Net provides independently adopt U-Net models. Each model performance depends on the variant pairs' convolution layers n explored from downward and upward operations as illustrated in the equation 2 and 3, respectively.

$$\tilde{x}_i^d = f_i^d(\tilde{x}_{i-1}^d). \tag{2}$$

The output of the denoising process for a given layer i, denoted as $\tilde{x}_i^d$, is obtained by convolving the denoised output of the previous layer $\tilde{x}_{i-1}^d$, denoted as $\tilde{x}_{i-1}^d$, with a convolution function f_i^d. The denoising process consists of n pairs of convolutional layers, where the first layer takes as input a frame of either clean x or adversarial image x', denoted as $\tilde{x}_0^d$. We repeatedly operate the downward direction for i = 1 to n.

To obtain the final denoised output in the upward direction, denoted as $\tilde{x}_1^u$, we apply the denoising process in the reverse direction. Specifically, we start from the convolution function f_{n-1}^u of the last corresponding layer in the downward direction, denoted as $\tilde{x}_n^d$. We then concatenate it with the denoised output of the previous layer in the downward direction $\tilde{x}_{n-1}^d$ to obtain $\tilde{x}_{n-1}^u$. We repeat this process in reverse order by feeding the corresponding layer in the upward direction, denoted as $\tilde{x}_{i+1}^u$, until we obtain the denoised output of the first layer, denoted as $\tilde{x}_1^u$, which corresponds to the predicted adversarial noise, denoted as $\Delta\tilde{x}$.

$$\tilde{x}_i^u = \tilde{x}_i^d \oplus f_i^u(\tilde{x}_{i+1}^u). \tag{3}$$

The network architecture of MU-Net is illustrated in Fig. 2. Each down-sampling and up-sampling blocks are scaled down and up by two times, respectively. Each down-sampling and up-sampling block is down-scaled and up-scaled by a factor of two, respectively. These blocks consist of a LeakyReLU operation, a 3x3 convolutional kernel with 2x2 strides and 1x1 paddings, and 2D batch normalization, except for the final down-sampling block, which does not operate batch normalization. We choose LeakyReLU in down-sampling layers because slightly negative values are allowed to leak, thereby the output values of down-sampling blocks are slightly flexible than ReLU in up-sampling blocks.

The denoised template region $\tilde{z} = z' - \Delta\tilde{z}$ and the denoised search region $\tilde{x} = x' - \Delta\tilde{x}$ are calculated in pixel-wise loss L_{PD} and feature-wise loss L_{FD} as seen in (1). The challenge of MU-Net is the weight sharing between a template region and a search region; their sizes differ by 127×127 and 255×255 pixels. Thereby it requires a sophisticated network architecture. The comparison of defending performance between different n-layer is explained in the next section.

4 Experimental results

4.1 Training and Evaluation

Training: We generated the adversarial noise offline for both template-region and search-region frames based on CSA [13] and DFA [15] noise generators. The models were trained end-to-end on the GOT-10k dataset [25]. We explore five models for each pair of convolution layers, such as 3, 4, 5, 6, and 7 pair layers.

Each model's iteration consists of 2 clean template frames, 2 CSA template-attack frames, 4 DFA frames, and 8 CSA search-attack frames for template-region learning and 8 CSA search-attack frames, 8 DFA frames for search-region learning by setting batch size as 16 pair-images. The training dataset comprises 500,000 pairs of images, and each epoch consists of 100,000 pairs. The adaptive moment estimation (ADAM) [29] is chosen to be the optimizer with momentum decay of 10^{-4} and a fixed learning rate of 10^{-3} in every epoch. Our proposed model converges after five epochs, which takes 7 hours of training on CPU Intel i9-12900KF, Nvidia RTX 3090, and RAM 128GB based on the PyTorch framework [30].

Evaluation: We evaluate MU-Net models for all five different pair layers based on PD and FD denoisers, with and without CSA [13] and DFA [15] attacking models. The baseline tracker for all experiments is SiamRPN++ [4]. We utilize benchmark datasets including OTB100 [31], LaSOT [32], UAV123 [33], VOT2018 [34], VOT2019 [35], and GOT-10k [25].

4.2 Performance Analysis in Different Pair Layers

We compare the defending performance against DFA and CSA attackers [15] among five different pair layers competes on OTB100. The smallest network consists of three pairs of layers, and the most extensive network has seven pairs of layers. Although the performance of the 7-pair-layer model is outstanding, all five pair-layer combinations have slightly different results against both DFA and CSA attackers. Based on DFA attacker, the lowest performance is the 3-pair-layer model, which has a 4.9% drop in success and a 3.6% drop in precision compares to the original SiamRPN++. The experimental result is displayed on Fig. 3, and the qualitative analysis of one pass evaluation is displayed on Fig. 5. On the other hand, the lowest performance against the CSA attacker is observed in the 4-pair-layer model, which has a 2.5% drop in success and a 1.4% drop in precision compares to the original SiamRPN++. In contrast, the 7-pair-layer model has a 2.5% drop in success and a 1.8% drop in precision, based on DFA attacker. Meanwhile, the 5-pair-layer model has 0.4% drop in success and full recovers in precision against the CSA attacker, as illustrated in Fig 4, and the qualitative analysis of one pass evaluation is displayed on Fig. 6.

The MU-Net contains five pair-layer models, including 3, 4, 5, 6, and 7, while the target adversarial attacking model consists of seven pair layers (14 layers) with 41.83 million parameters. Our MU-Net models are highly effective in defending against adversarial attacks. The most miniature model can defend 1626 FPS on the template region and 1205 FPS on the search region. More details are provided in Table 1.

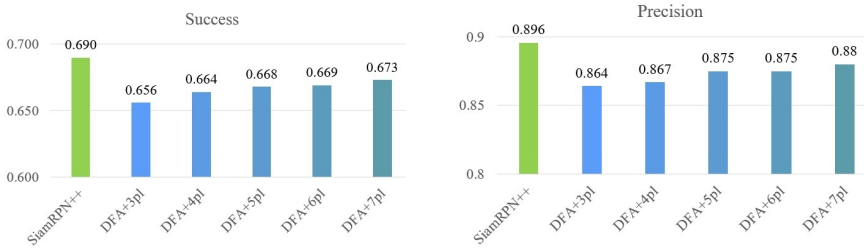


Fig. 3 Comparison of FD-defensive results of different n-pair-layer (pl) against DFA attackers' both-region attacks on OTB100.

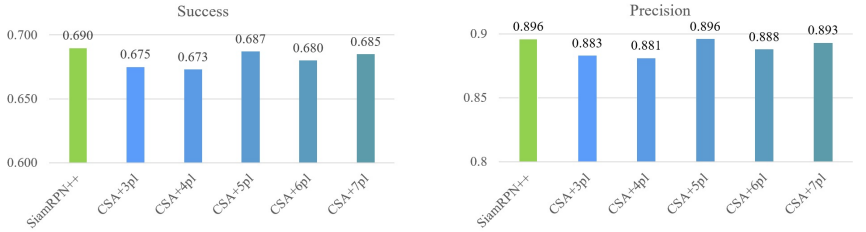


Fig. 4 Comparison FD defensive results of different n-pair-layer (pl) against CSA attackers' both-region attacks on OTB100.

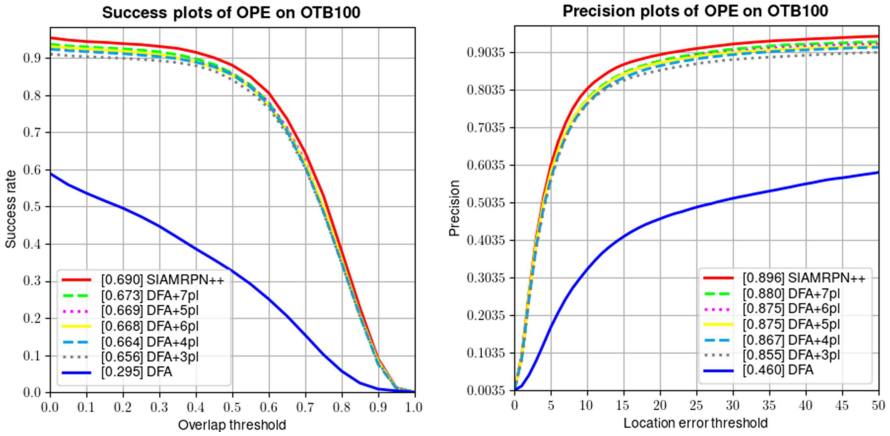


Fig. 5 Qualitative analysis of our FD defensive models against both-region attacking of DFA attacker on OTB100.

4.3 Comparison results of PD and FD denoisers against CSA and DFA attacking models

We evaluate pixel-wise (PD) and feature-wise (FD) denoisers against CSA and DFA attacking models based on SiamRPN++ by choosing the best performance of five different pair-layer models and using OTB100 as benchmark dataset. As illustrated in Table 2, our PD and FD models successfully improve

Table 1 Comparison number of parameters, convolution layers, and inference time of our MU-Net.

#Pairs	#Layers	#Params.	Inference Time			
			Milliseconds/Frames		Frames/Second	
			Template	Search	Template	Search
3	6	5.6M	0.62	0.83	1626	1205
4	8	8.7M	0.65	0.91	1529	1096
5	10	9.4M	0.83	1.29	1209	776
6	12	16.5M	1.06	1.61	946	620
7	14	41.8M	1.90	2.60	526	384

the attacking performance on both CSA and DFA attacks for all three types; template region, search region, and both-region attacks. The qualitative analysis among PD and FD denoisers against CSA and DFA attackers is displayed in Fig 7, and example tracking images are displayed in Fig 9. The superior performance of the FD method over the PD approach can be attributed to the network architecture of SiamRPN++, in which features have a higher impact than the similarity of the original image. This effect leads to adversarial features being more closely aligned with the features of the original image than with those of the denoised image by using PD method.

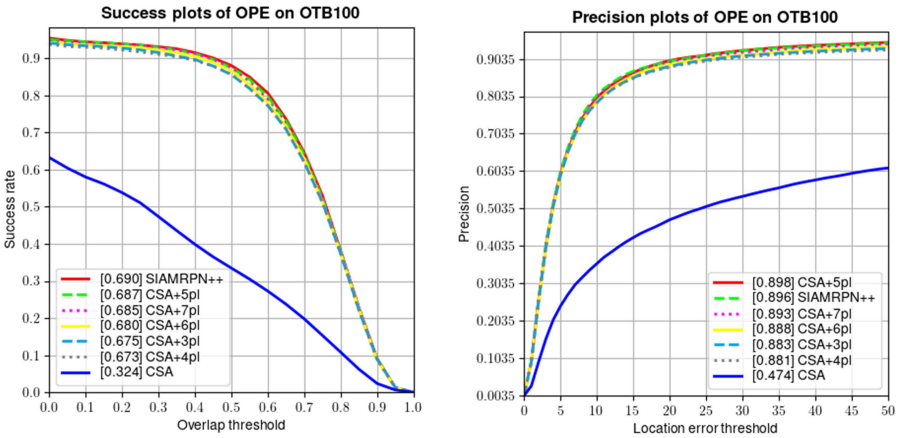


Fig. 6 Qualitative analysis of our FD defensive models against both-region attacking of CSA attacker on OTB100.

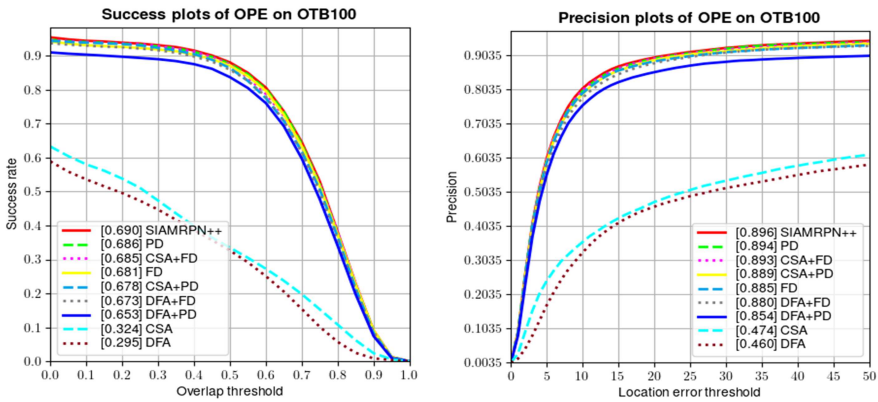


Fig. 7 Qualitative analysis of our PD and FD defensive models with and without both-region attacking of CSA and DFA attackers on OTB100.

Table 2 Comparison of defensive performance of our PD and FD models with and without CSA and DFA attackers using different attacking methods, including template-region attack, search-region attack, and both-region attack on the OTB100 dataset.

	Success (%Drop)				**Precision (%Drop)**			
	Clean	**Attack-T**	**Attack-SR**	**Attack-Both**	**Clean**	**Attack-T**	**Attack-SR**	**Attack-Both**
CSA	0.690	0.593 (14.1)	0.351 (49.1)	0.324 (53.0)	0.896	0.786 (12.3)	0.492 (45.1)	0.477 (46.8)
CSA+PD	0.686 (0.6)	0.687 (0.4)	0.681 (1.3)	0.678 (1.7)	0.894 (0.2)	0.896 (0.0)	0.891 (0.6)	0.889 (0.8)
CSA+FD	0.681 (1.3)	0.686 (0.6)	0.681 (1.3)	0.687 (0.4)	0.885 (1.2)	0.896 (0.0)	0.887 (1.0)	0.893 (0.3)
DFA	0.690	0.467 (32.3)	0.263 (61.9)	0.295 (57.2)	0.896	0.656 (26.8)	0.391 (56.4)	0.460 (48.7)
DFA+PD	0.686 (0.6)	0.679 (1.6)	0.659 (4.5)	0.653 (5.4)	0.894 (0.2)	0.892 (0.4)	0.858 (4.2)	0.854 (4.7)
DFA+FD	0.681 (1.3)	0.687 (0.4)	0.680 (1.4)	0.673 (2.5)	0.885 (1.2)	0.896 (0.0)	0.884 (1.3)	0.880 (1.8)

4.4 Comparison results on various tracking datasets

We conduct experiments to evaluate the effectiveness of three different MU-Net models, which have three, five, and seven pair layers respectively, against both DFA and CSA attackers using FD denoiser on six benchmark datasets: OTB100, LaSOT, UAV123, VOT2018, VOT2019, and Got-10k. Our results, shown in Table 3, demonstrate that the FD model successfully defends against DFA and CSA attacks on all benchmark datasets.

To determine the best defensive performance of the three variation pair layers, we evaluate their defensive performance against DFA and CSA attackers. Our results indicate that the defensive performance against DFA attacker slightly decrease from the original tracker by 0.6% success on LaSOT, 1.5% success on UAV123, 4.6% EAO on VOT2018, 1.4% AO on GOT-10k, and completely resile the EAO on VOT2019 when the best pair-layer is chosen. Similarly, the defensive performance results against CSA attacker slightly decrease from the original tracker by 2.2% success on LaSOT, 1.2% success on VOT2018, 3.2% EAO on GOT-10k, and completely resile the EAO on UAV123 and VOT2019.

Table 3 Comparison of our FD defensive model in different pair layers against DFA and CSA attackers. The best defending performance for each benchmark is highlighted in red color.

	LaSOT (%Drop)		UAV123 (%Drop)		VOT2018 (%Drop)		VOT2019 (%Drop)		Got-10k (%Drop)	
	Success	Norm-Prec.	Success	Precision	Accuracy	EAO	Accuracy	EAO	SR 0.75	AO
SiamRPN++	0.504	0.583	0.611	0.798	0.607	0.388	0.604	0.275	0.329	0.510
DFA	0.123 (75.6)	0.158 (72.9)	0.242 (60.4)	0.417 (47.7)	0.471 (22.4)	0.099 (74.5)	0.446 (26.2)	0.087 (68.4)	0.035 (89.4)	0.228 (55.3)
DFA+3pl	0.490 (2.8)	0.574 (1.5)	0.602 (1.5)	0.793 (0.6)	0.590 (2.8)	0.342 (11.9)	0.593 (1.8)	0.258 (6.2)	0.241 (26.7)	0.461 (9.6)
DFA+5pl	0.501 (0.6)	0.583 (0.0)	0.586 (4.1)	0.774 (3.0)	0.592 (2.5)	0.370 (4.6)	0.589 (2.5)	0.275 (0.0)	0.267 (18.8)	0.470 (7.8)
DFA+7pl	0.487 (3.4)	0.565 (3.1)	0.588 (3.8)	0.779 (2.4)	0.587 (3.3)	0.348 (10.6)	0.570 (5.6)	0.265 (3.6)	0.329 (0.0)	0.503 (1.4)
CSA	0.172 (65.9)	0.207 (64.5)	0.294 (51.9)	0.466 (41.6)	0.467 (23.1)	0.077 (80.2)	0.456 (24.5)	0.084 (69.5)	0.076 (76.9)	0.235 (53.9)
CSA+3pl	0.480 (4.8)	0.565 (3.1)	0.611 (0.0)	0.798 (0.0)	0.605 (0.3)	0.343 (11.6)	0.604 (0.0)	0.267 (2.9)	0.314 (4.6)	0.491 (3.7)
CSA+5pl	0.474 (6.0)	0.558 (4.3)	0.609 (0.3)	0.798 (0.0)	0.600 (1.2)	0.382 (1.5)	0.604 (0.0)	0.275 (0.0)	0.319 (3.0)	0.500 (2.0)
CSA+7pl	0.493 (2.2)	0.576 (1.2)	0.610 (0.2)	0.798 (0.0)	0.604 (0.5)	0.376 (3.1)	0.604 (0.0)	0.275 (0.0)	0.283 (14.0)	0.488 (4.3)

4.5 Ablation Study

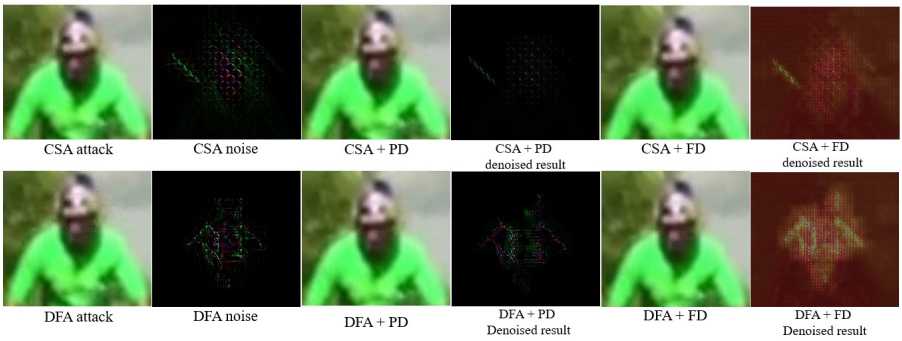


Fig. 8 Both the noise and denoised images are amplified by a factor of 10 for enhanced visibility. The first, third, and fifth columns show adversarial images, while the second, fourth, and sixth columns show their corresponding adversarial noise results and denoised results, respectively.

Noise pattern: Adversarial noise and denoised results are imperceptible by human vision. In best visibility, adversarial noise and denoised results are magnified 10 times, as demonstrated in Fig. 8. PD demonstrates superior pixel recovery when applied to clean input. This observation suggests that the FD approach causes some degradation of the clean image, while the PD method does not. On the other hand, when defending against adversarial noise attacks, the FD method outperforms PD due to the higher impact of features on the network architecture of SiamRPN++ than the similarity of the original image and tends to be extensive because the difference between clean and denoised images is limitless in feature-wise denoiser.

Table 4 Comparison of inference times of attackers with and without defenders.

		attacker (frames/s)		
		CSA	DFA	IOU_attack
defender pair layers	-	57.8	55.3	1.6
	3	53.2	54.0	-
	5	52.2	50.8	-
	7	48.1	48.7	-

Inference times of attacking and defending models: In this work, we proposed the defensive model that successfully defends the SiamRPN++ [4] model with offline attacking models, such as CSA [13] and DFA [15]. Despite the efficacious dropping performance of online learning, its inference time is prolonged. We compare the inference times of attacking models for offline (CSA [13], DFA [15]) and online (IoU attacker [16]) learning in Table 4. The

inference times of our defenders range from 48.1 to 54 FPS, which represents an increase of 4-9 FPS compared to CSA and DFA. In contrast, the inference time of the IoU attack is 1.6 FPS which is unlikely to attack or be defended in a practical application.

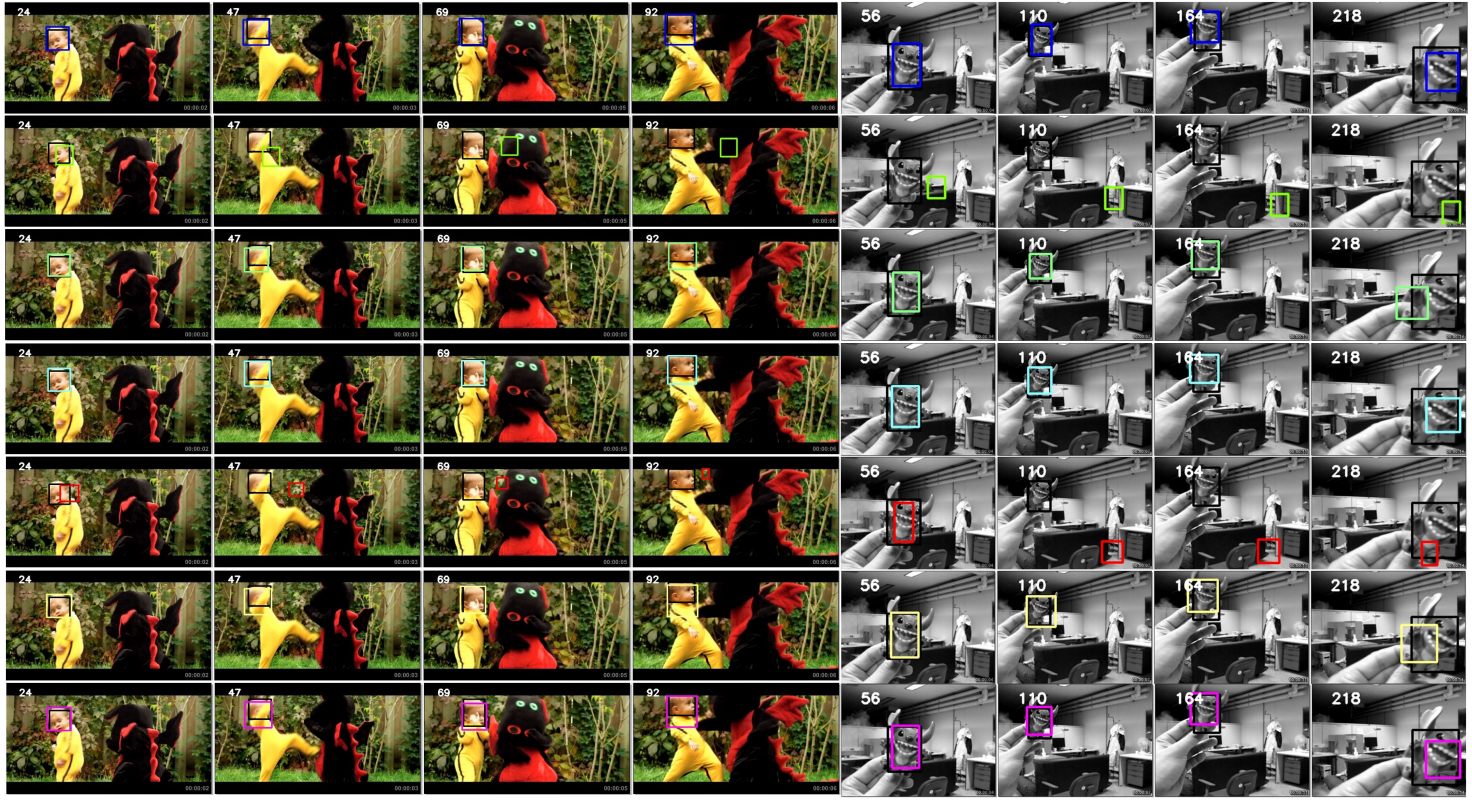


Fig. 9 The initial row of images depicts the original SiamRPN++. The second, third, and fourth rows illustrate the CSA attacker with and without PD, and FD denoisers, respectively. The fifth, sixth, and seventh rows show the DFA attacker with and without PD, and FD denoisers, respectively. In all images, the ground-truth bounding box is drawn in black.

Springer Nature 2021 LaTeX template

5 Conclusion

In conclusion, our proposed MU-Net architectures effectively address the problem of adversarial attacks on visual tracking by using pixel-wise and feature-wise denoisers to generate robust denoised images against attacks. Our experimental results demonstrate that our models perform well across various benchmark datasets, with the ability to defend against both attacked and clean frames. Importantly, our models achieve these results while maintaining a subtle execution time, making it practical for real-world applications.

Our work contributes to the state-of-the-art in adversarial defense for visual tracking, achieving a remarkable range of performance results with only marginal drops in accuracy (ranging from 0% to 4.6%) across the benchmark datasets evaluated. Our proposed approach has significant potential to enhance the robustness and security of visual tracking systems in various real-world settings.

References

[1] 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, pp. 850–865 (2016). Springer

[2] 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, pp. 8971–8980 (2018)

[3] Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: A unifying approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1328–1338 (2019)

[4] Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)

[5] Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: ATOM: Accurate Tracking by Overlap Maximization. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4660–4669 (2019). http://arxiv.org/abs/1811.07628

[6] Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 6182–6191 (2019)

[7] Yan, B., Peng, H., Wu, K., Wang, D., Fu, J., Lu, H.: Lighttrack: Finding

lightweight neural networks for object tracking via one-shot architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15180–15189 (2021)

[8] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2014). http://arxiv.org/abs/1412.6572

[9] Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)

[10] Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9185–9193 (2018)

[11] Yan, X., Chen, X., Jiang, Y., Xia, S.-T., Zhao, Y., Zheng, F.: Hijacking tracker: A powerful adversarial attack on visual tracking. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2897–2901 (2020). IEEE

[12] Chen, X., Yan, X., Zheng, F., Jiang, Y., Xia, S.-T., Zhao, Y., Ji, R.: One-shot Adversarial Attacks on Visual Tracking with Dual Attention. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 10176–10185 (2020)

[13] Yan, B., Wang, D., Lu, H., Yang, X.: Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 990–999 (2020)

[14] Suttapak, W.: Feature-based attack on visual tracking. In: 2021 25th International Computer Science and Engineering Conference (ICSEC), pp. 45–50 (2021). IEEE

[15] Suttapak, W., Zhang, J., Zhang, L.: Diminishing-feature attack: The adversarial infiltration on visual tracking. Neurocomputing (2022)

[16] Jia, S., Song, Y., Ma, C., Yang, X.: Iou attack: Towards temporally coherent black-box adversarial attack for visual object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6709–6718 (2021)

[17] Liao, F., Liang, M., Dong, Y., Pang, T., Hu, X., Zhu, J.: Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition, pp. 1778–1787 (2018)

[18] Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12299–12310 (2021)

[19] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

[21] Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S.: Siamcar: Siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6269–6277 (2020)

[22] Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G.: Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12549–12556 (2020)

[23] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)

[24] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)

[25] Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5374–5383 (2019)

[26] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241 (2015). Springer

[27] Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networkss. In: Computer Vision (ICCV), 2017 IEEE International Conference On (2017)

[28] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference On (2017)

[29] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[30] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)

[31] Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)

[32] Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5374–5383 (2019)

[33] Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for uav tracking. In: European Conference on Computer Vision, pp. 445–461 (2016). Springer

[34] Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., ˇCehovin 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), pp. 0–0 (2018)

[35] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.-K., ˇCehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., *et al.*: The seventh visual object tracking vot2019 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 0–0 (2019)