Real-time Siamese visual object tracking using attention and anchor-free mechanism

Hoang Dinh Thang[1], Do Ngoc Tuan[1], Thai Trung Kien[1], Tran Quoc Long[2*]

[1]Institute of Information Technology, Academy of Military Science and Technology, Hanoi, Vietnam;
[2]VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam.
[*]Corresponding author: tqlong@vnu.edu.vn
Received 11 April 2022; Revised 21 May 2022; Accepted 10 June 2022; Published 28 June 2022.
DOI: https://doi.org/10.54939/1859-1043.j.mst.80.2022.132-141

ABSTRACT

Trackers based on Siamese have consistently demonstrated superior performance in tracking visual objects. The majority of existing trackers calculate the features of the target template and search image independently and then estimate the target's scale and aspect ratio using either a multi-scale searching scheme or pre-defined anchor boxes. This paper proposed a Siamese attention network for tracking visual objects. An attention fusion mechanism is generated using pixel-level matching of template and search features. The framework proposed is anchor-free, making it both simple and effective. Extensive experiments on visual tracking benchmark VOT2018 and UAV123 demonstrate that our tracker operates at 42 fps and achieves state-of-the-art performance.

Keywords: Visual object tracking; Attention mechanism; Anchor-free mechanism.

1. INTRODUCTION

Visual object tracking is a critical task in computer vision because it enables the state of a target object to be determined within a video sequence. Due to the breadth of its applications, including intelligent surveillance, human-machine interaction, and unmanned vehicles, visual object tracking has garnered considerable attention. Visual tracking has come a long way. However, it remains a difficult task, particularly for real-world applications, because objects in unconstrained recording conditions frequently exhibit large variations in illumination, scale, background clutter, and occlusions, among other characteristics.

The Siamese network-based [1-7] has attracted significant community interest. These Siamese trackers formulate the visual object tracking problem in terms of learning a general similarity map via cross-correlation between the feature representations of the target template and the feature representations of the search region. In recent years, anchor boxes have been widely used in visual tracking [2-4, 7] and have demonstrated a reasonable trade-off between speed and accuracy. However, because anchors are used to propose regions, these trackers are extremely sensitive to the number, size, and aspect ratio of anchor boxes, necessitating expertise in hyperparameter tuning to achieve successful tracking with these trackers. SiamCAR [5] and SiamBAN[6] applied the FCOS [8] concept to tracking and developed an anchor-free tracker that obviates the need for intricate anchor setting parameters. SiamCorners [9] took the tracking concept from CornerNet[10] and created a simple yet effective anchor-free tracker. However, these Siamese-based trackers typically localize the target by simultaneously optimizing the classification and regression branches, which can result in a tracking procedure mismatch.

Visual object tracking has recently been introduced with the attention mechanism [7, 11]. However, [11] computes the attention and deep features of the target template and search images independently, limiting the potential performance of the Siamese architecture. SiamAtnn [7] is an anchor-based tracker that investigates both self-and cross-attention in order to improve the discriminative capacity of the template and search features prior to performing depth-wise cross-correlation on the fusion feature. On the other hand, cross-correlation is a linear matching

procedure, which limits the tracker's ability to capture the complex non-linear interaction between the template and search patch. While these studies demonstrated an increase in tracking accuracy via the attention mechanism, the channel spatial compositional attention blocks did not demonstrate significant overall benefits over recent attentions.

We present an attention fusion mechanism in this work that is based on pixel-level matching of template and search features. To be more precise, key and value maps are generated from features, which serve as an encoding of visual semantics for matching and a repository of detailed appearance information for prediction.

Our primary contributions are as follows:

• We propose a novel framework for Siamese tracking that consists of feature extraction, fusion, and head prediction modules. The network's prediction head is anchor-free, allowing for more precise localization and classification.

• The proposed fusion mechanism to fuse features from two Siamese network branches with an attention mechanism based on pixel-level matching.

• Using the benchmark datasets VOT2018 and UAV123, we conduct extensive experiments and demonstrate that our tracker outperforms state-of-the-art findings while operating at real-time speeds.

2. RELATED WORKS

Over the last few decades, visual tracking has been one of the most active research areas in computer vision. Due to the fact that a comprehensive review of related trackers is beyond the scope of this paper, we will concentrate on three areas that are particularly relevant to our work: visual tracking, attention mechanisms, and anchor-free mechanisms.

2.1. Visual tracking

Tracking researchers devote considerable effort to developing more accurate and faster trackers from a variety of perspectives, including feature extraction, classifier design, template updating, and bounding box regression. The extraction of early features is primarily based on color, texture, or other hand-crafted features. Due to the rapid advancement of deep learning, convolutional neural network (CNN)-based features have gained widespread adoption.

Recently, the tracking community has paid close attention to Siamese network-based methods. A Siamese tracker learns a similarity matching function through cross-correlation between feature representations learned from a template and test images. These trackers are typically quite efficient, as they reuse the target from the initial frame and do not require online updates.

SiamFC [1] is a ground-breaking work that calculates the similarity prediction of a template and search image at a rate of over 100 frames per second using a full-convolutional network. Encouraged by the success of the original model, numerous researchers have continued the work and proposed several updated models, including [2-6].

The SiamRPN [2] tracker is inspired by the region proposal network (RPN [12]) for object detection. It extracts region proposals from the output of the Siamese network. By jointly learning a classification and regression branch for region proposals, SiamRPN avoids the time-consuming step of obtaining multi-scale feature maps. It has difficulty, however, detecting distractor objects that appear to be identical to the target object. Based on SiamRPN, DaSiamRPN[3] increases the amount of negative hard training data during the training phase. They enhance the discrimination capabilities of the tracker through data enhancement, resulting in a much more robust tracking result. Additionally, the tracker is extended to support visual tracking over an extended period of time. SiamRPN++ [4] substitutes ResNet [13] for feature extraction in order to take advantage of the powerful deep feature extracted by the deep network.

While anchor-based trackers are capable of adapting to changes in scale and aspect ratio, caution should be exercised when designing and fixing the anchor box parameters. Frequently, design parameters require heuristic adjustments and a variety of tricks to optimize performance. In comparison to anchor-based trackers, our tracker is more flexible and generic due to the absence of hyper-parameters associated with anchor boxes.

2.2. Attention mechanism

Numerous visual tasks incorporate attention mechanisms to compensate for the limitations of convolutional neural networks in their standard configuration [14, 15]. Each input tensor is used to compute an attention tensor, which is then used to reweight the self-attention mechanism. Self-attention augmented convolutional models [14], and standalone self-attention models [15] have demonstrated significant performance gains in a variety of vision tasks, including object detection and image classification. [16] pioneered the use of self-attention to collect contextual information for semantic segmentation. Numerous attempts to incorporate an attention mechanism into the field of tracking have been made. Wang et al. [11] proposed a RASNet by developing an attention mechanism for Siamese trackers, but it is entirely based on template information, limiting its representation capability. [17] in particular, makes use of channel-wise attention to feed the matching network with target-specific information. SiamAttn [7] investigates self- and cross-branch attention in order to improve the discriminative ability of target features. SiamAtnn is an anchor-based tracker.

2.3. Anchor-free mechanism

CornerNet [10] pioneered the concept of a corner-based detector by converting the target's bounding box to a pair of corner predictions. Without referencing an anchor, FCOS [8] proposed predicting an object's existence and bounding box coordinates. SiamBAN and SiamCAR adapted the FCOS concept for tracking, whereas SiamCorners adapted the CornerNet concept for tracking, resulting in a simple yet effective anchorless tracker. While these works have simplified and improved the accuracy of tracking using anchorless techniques, they continue to rely heavily on the correlation operation fusion of template and search region features.

We use the fundamental concepts of Attention and FCOS to create a new Siamese network for visual tracking in this work. The network uses an anchor-free head prediction mechanism combined with attention feature enhancement and fusion.

3. PROPOSED METHOD

This section describes the specifics of our proposed networks. As illustrated in figure 1, it is composed of four major components: a mechanism for feature extraction, a mechanism for fusion, and a head for prediction.

3.1. Feature extraction

In this work, we construct the Siamese subnetwork for visual feature extraction using a fully convolutional network. Two identical branches comprise the Siamese network. The template branch is one of them, while the search branch is the other. We use a modified version of ECA-Net [18] pre-trained on [19] as the backbone network for feature extraction. We use the fourth stage's (layer3) outputs as final outputs in our tracker. The backbone processes the template patch (denoted by $\boldsymbol{z} \in \mathbb{R}^{3\times H_{z_0}\times W_{z_0}}$) and the search patch (denoted by $\boldsymbol{x} \in \mathbb{R}^{3\times H_{x_0}\times W_{x_0}}$) to obtain their respective feature maps $\boldsymbol{F}_z \in \mathbb{R}^{C_z\times H_z\times W_z}$ and $\boldsymbol{F}_x \in \mathbb{R}^{C_x\times H_x\times W_x}$, respectively, where $H_z, W_z = \frac{H_{z_0}}{8}, \frac{W_{z_0}}{8}$, $H_x, W_x = \frac{H_{x_0}}{8}, \frac{W_{x_0}}{8}$, and $C_z = C_x = 1024$. Following that, we use a neck with 1×1 convolution to reduce the output features channel to 256, and follow by [2] we only use the features from the template branch center 7×7 areas, which still capture the entire target region. Our network's output features are defined as $\boldsymbol{Z} \in \mathbb{R}^{C\times h\times w}$ and $\boldsymbol{X} \in \mathbb{R}^{C\times H\times W}$, with C equal to 256.

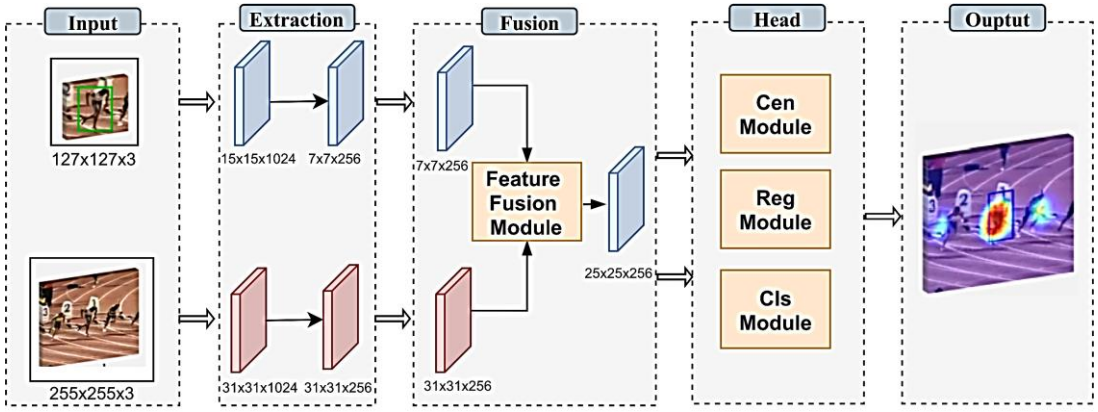

Figure 1. *An overview of the proposed Networks. It consists of an input, feature extraction, an attention feature fusion, and a prediction head for classification and regresstion.*

3.2. Fusion mechanism

When the appearance of a target template or search patch changes or is occluded, detailed local features become more important for matching the target template and search patch. Thus, rather than relying solely on depth-wise correlation, we propose an attention fusion mechanism in which template and search features are matched pixel-by-pixel, as illustrated in figure 2. To be more precise, key and value maps are generated from features, which are used to encode visual semantics for matching and detailed appearance information for prediction. We generate a key and value features map using **Z** and **X** from the backbone by:

$$\begin{aligned} \boldsymbol{K}_T &= \sigma_1\big(\boldsymbol{W}_{kt}(\boldsymbol{Z})\big) \in \mathbb{R}^{\frac{C}{8}\times hw} \\ \boldsymbol{V}_T &= \sigma_2\big(\boldsymbol{W}_{vt}(\boldsymbol{Z})\big) \in \mathbb{R}^{hw\times C/2} \\ \boldsymbol{K}_S &= \sigma_3\big(\boldsymbol{W}_{ks}(\boldsymbol{Z})\big) \in \mathbb{R}^{HW\times C/8} \\ \boldsymbol{V}_S &= \sigma_4\big(\boldsymbol{W}_{vs}(\boldsymbol{Z})\big) \in \mathbb{R}^{HW\times C/2} \end{aligned} \tag{1}$$

where $\boldsymbol{W}_{kt}, \boldsymbol{W}_{vt}, \boldsymbol{W}_{ks}$, and $\boldsymbol{W}_{vs}$ are 3 × 3 convolution layer, respectively, σ_1, σ_2, σ_3 and σ_4 are our tensors reshape operators. Then we calulate the similarities between key maps of template feature and search feature by:

$$\boldsymbol{SM} = \boldsymbol{K}_T \times \boldsymbol{K}_S \tag{2}$$

where "×" is the matrix dot-product operation.

After computing the similarity of pixel features, we perform softmax normalization as:

$$\boldsymbol{S} = F_{SM}(\boldsymbol{SM}) \in \mathbb{R}^{HW\times hw} \tag{3}$$

Then calculate embedding value and concat this value with the value map of search feature to generate attention fusion feature as:

$$\boldsymbol{E} = \boldsymbol{S} \times \boldsymbol{V}_T \in \mathbb{R}^{HW\times C/2} \tag{4}$$

$$\boldsymbol{R} = concat(\boldsymbol{V}_S, \boldsymbol{E}) \in \mathbb{R}^{H\times W\times C} \tag{5}$$

Suppose the input features are $\boldsymbol{R} \in \mathbb{R}^{H\times W\times C}$, we can generate a spatial self-attention map $\boldsymbol{A}^{sp}(\boldsymbol{R}) \in \mathbb{R}^{1\times H\times W}$ [20]:

$$\boldsymbol{A}^{sp}(\boldsymbol{R}) = F_{SG}\big[\sigma_3(F_{SM}(\sigma_1(F_{GP}(\boldsymbol{W}_q(\boldsymbol{R})))) \times \sigma_2(\boldsymbol{W}_v(\boldsymbol{R})))\big] \tag{6}$$

where $\boldsymbol{W}_q$ and $\boldsymbol{W}_v$ are 1 × 1 convolution layer respectively, s$F_{SG}(.)$ is Sigmoid operator, and $F_{SM}(.)$ is SoftMax operator. F_{GP} is a global pooling operator, $F_{GP}(\boldsymbol{R}) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{R}(:,i,j))$ and "×" is the matrix dot-product operation. The output of spatial self attention is:

$$\boldsymbol{R}^* = \boldsymbol{A}^{sp}(\boldsymbol{R}) \odot \boldsymbol{R} \in \mathbb{R}^{H \times W \times C} \quad (7)$$

where $\odot$ is a channel-wise multiplication operator.

The generated response map $\boldsymbol{R}^*$ contains massive information for classification and regression.

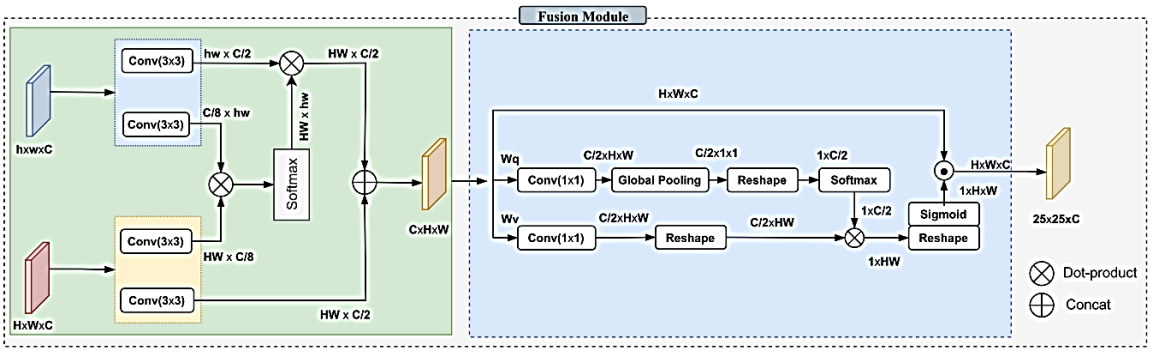

Figure 2. *The proposed feature fusion using attention mechanism.*

3.3. Prediction head network

As illustrated in figure 3, the prediction head is composed of three subnetworks: a classification branch that predicts the category for each location, a regression branch that computes the target bounding box for this location, and a centeredness branch that runs concurrently with the regression branch to eliminate outliers. For a response map $\boldsymbol{R}^*$ constructed using the Siamese subnetwork, the classification branch generates a classification feature map $P^{cls} \in \mathbb{R}^{25\times25\times2}$, the regression branch generates a regression feature map $P^{reg} \in \mathbb{R}^{25\times25\times4}$, and the branch generates a centeredness feature map $P^{cen} \in \mathbb{R}^{25\times25\times1}$, where each point value represents the location's centerness score.

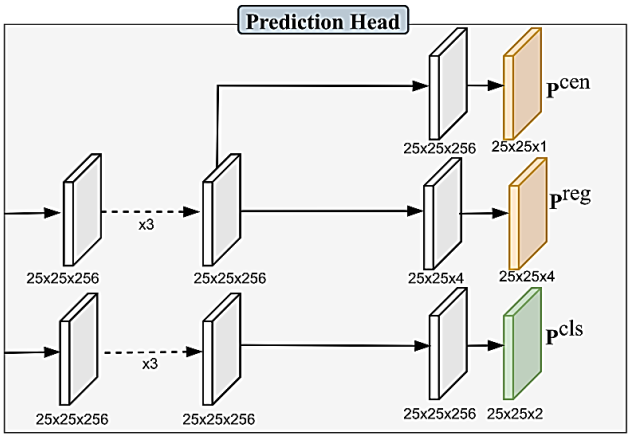

Figure 3. *The prediction head, which consists of three subnetworks: localization, classification and centerness subnet.*

3.4. Loss function

Let (g_{x_c}, g_{y_c}), and (g_w, g_h) denote the center position and size of the target box, respectively. We create positive and negative labels in this work by utilizing the ellipse figure region, as described in [6]. Two ellipses are present, denoted by the letters E_1 and E_2.

We can map each location (i, j) in feature map $\boldsymbol{R}^*$ to the input frame to obtain the image location that corresponds to it (p_i, p_j). If (p_i, p_j) lies within the ellipse E_2, it is regarded as positive; if (p_i, p_j) lies outside the ellipse E_1, it is regarded as negative; and if (p_i, p_j) lies between the ellipses E_1 and E_2, it is ignored.

The box localization losses are calculated by multiplying the IoU loss by the following formula:

$$L_{reg} = 1 - \frac{1}{N_{pos}} \sum_{i,j} \mathbb{I}_{\{(p_i,p_j)\in E_2\}} L_{IoU}(p_{i,j}^{reg}, g_{i,j}) \quad (8)$$

where N_{pos} denotes the number of positive samples, $\mathbb{I}_{\{(p_i,p_j)\in E_2\}}$denotes an indicator function that equals 1 when $(p_i, p_j) \in E_2$ is positive and 0 otherwise, L_{IoU} denotes the IoU loss as UnitBox [21], $g_{i,j}$ denotes the ground-truth box, and $P_{i,j}^{reg}$ denotes the regression bounding box.

The cross-entropy loss, which is used to calculate the classification score loss for each box, is defined as follows:

$$L_{cls} = y_o \log(p_o) + (1 - y_o)(1 - p_o) + y_b \log(p_b) + (1 - y_b)(1 - p_b) \quad (9)$$

where y_o and y_b denote the target object and background object labels, respectively, p_o and p_b denote the probability tracker predicted target object and background.

The centerness loss L_{cen} defined as follows:

$$L_{cen} = \frac{1}{\sum \mathbb{I}(\tilde{t}_{(i,j)})} \sum_{\mathbb{I}(\tilde{t}_{(i,j)})==1} C(i,j) * log A_{w\times h\times 1}^{cen}(i,j) + \big(1 - C(i,j)\big) * \log\left(1 - A_{w\times h\times 1}^{cen}(i,j)\right) \quad (10)$$

where where $C(i,j)$ is in contrast with the distance between the corresponding location (x, y) and the object center in the search region. If (x, y) is a location within background, the value of C(i, j) is set to 0.

$$C(i,j) = \mathbb{I}\big(\tilde{t}_{(i,j)}\big) * \sqrt{\frac{\min(\tilde{l},\tilde{r})}{\max(\tilde{l},\tilde{r})} \times \frac{\min(\tilde{t},\tilde{b})}{\max(\tilde{t},\tilde{b})}} \quad (11)$$

Using the aforementioned losses, we define our multi-task loss function as follows:

$$L = \lambda_1 L_{cls} + \lambda_2 L_{reg} + \lambda_3 L_{cen} \quad (12)$$

During training, we empirically set $\lambda_1 = \lambda_3 = 1$, and $\lambda_2 = 2$ for all experiments.

3.5. Inference

We crop the first frame's template patch and feed it to the network during inference. For subsequent frames, we trim the search patch and extract features based on the target position of the previous frame, followed by prediction in the search region to generate the classification map $P_{h\times w\times 2}^{cls}$ and the refined regression map $P_{h\times w\times 4}^{reg}$. Following that, we can generate prediction boxes using the following procedure by performing the following:

$$\begin{aligned} p_{x_1} &= p_i - d_l^{reg} \\ p_{y_1} &= p_j - d_t^{reg} \\ p_{x_2} &= p_i - d_r^{reg} \\ p_{y_2} &= p_j - d_b^{reg} \end{aligned} \quad (13)$$

where p_i, p_j as mentioned in 3.4 aboves, d_l^{reg}, d_t^{reg}, d_r^{reg}, and d_b^{reg} denote the box regression map's prediction values, and (p_{x_1}, p_{y_1}) and (p_{x_2}, p_{y_2}) denote the prediction box's top-left and bottom-right corners, respectively.

Following the generation of prediction boxes, the prediction box with the highest score is chosen and its size is updated using linear interpolation with the previous frame's state using the cosine window and scale change penalty to smooth target movements and changes.

4. EXPERIMENTS

To verify the proposed components' effects, we conduct extensive experiments on the benchmark database VOT2018 [22], as well as ablation studies.

4.1. Validation datasets

VOT2018. VOT2018 [22] are widely used visual object tracking benchmarks. VOT2018 contains 60 sequences ranging in difficulty. Rotated bounding boxes are used in the dataset, and evaluation is performed using a reset-based methodology. The accuracy (A), robustness (R), and expected average overlap (EAO) of trackers against both benchmarks are all evaluated.

UAV123. UAV123 [23] contains 123 video sequences and over 100,000 images captured by unmanned aerial vehicles flying at low altitudes. Unlike other tracking datasets, UAV123 has an aerial perspective and typically tracks small targets. Each Sequence is fully annotated using upright bounding boxes. The objects in this dataset move quickly and have a large scale, variable illumination, and occlusion, all of which complicate tracking. The success plot and precision plot of OPE are used to evaluate the overall performance of UAV123 [24].

4.2. Implementation details

We train the network on the COCO [25], ImageNet DET [26], ImageNet VID [26], and GOT10k [27] training sets in order to develop a general understanding of how to compare general objects for visual tracking. On ImageNet, the network is pre-trained [19]. To ensure a fair comparison, we set the size of the template patch to 127×127 pixels and the search patch to 255×255 pixels, as described in [4]. Our approach is implemented in Python using PyTorch on a computer with 2 Intel(R) Xeon(R) Bronze 3104 CPU @ 1.70GHz, 96G RAM and 2 Nvidia GTX 1080ti.

4.3. Comparisons with the State-of-the-art

4.3.1. Result on VOT2018

The following table compares the VOT2018 results. Our tracker outperforms state-of-the-art methods on all metrics on VOT2018, achieving 0.591 accuracy, 0.180 robustness, 32 lost numbers, and 0.447 EAO. In comparison to SiamBAN's tracker, our tracker achieves comparable EAO while utilizing a simpler and faster network. SiamBAN tracker is trained on a larger dataset (extra added LaSOT [28], YouTube-BB [29]) with three backbone layers as input features. The proposed method achieves real-time performance at 42 fps (real-time speed greater than 30 fps). The results of the evaluation of other methods are taken from the publication of these methods on GitHub or on the VOT challenge page.

Table 1. *Results on VOT2018, with Accuracy (A), Robustness (R), Lost Number (LN), and Expected Average Overlap (EAO).*

Tracker	Source	A(↑)	R(↓)	LN(↓)	EAO(↑)	GPU	FPS(↑)
Ours		0.591	0.180	32.0	**0.447**	**GTX 1080ti**	**42**
SiamBAN[6]	CVPR 2020	0.590	0.178	38.0	0.447	GTX 1080ti	40
DiMP-50[30]	ICCV 2019	0.597	0.152	32.5	0.439	GTX 1080	43
SiamRPN++[4]	CVPR 2019	0.600	0.234	50.0	0.415	Titan Xp Pascal	35
ATOM[31]	CVPR 2019	0.590	0.203	43.4	0.400	GTX 1080	30
SiamRPN[2]	CVPR 2018	0.586	0.276	59.0	0.383	GTX 1060	160
UPDT[32]	ECCV 2018	0.536	0.184	39.2	0.378	-	-
DeepCSRDCF[32]	ICCV 2015	0.489	0.276	59.0	0.293	-	-
CCOT[34]	ECCV 2016	0.494	0.318	68.0	0.267	-	-

Each frame in VOT2018 has the following visual attributes annotated: change in illumination, occlusion, camera movement, size change, and motion change. Unassigned frames are those that do not have any of the five attributes assigned to them. We compare the EAO of the visual attributes of the top-performing trackers. As illustrated in figure 4, our tracker is ranked first for occlusion tributes, second for camera motion, and third for unassigned. This demonstrates the tracker's resistance to occlusion and camera motion.

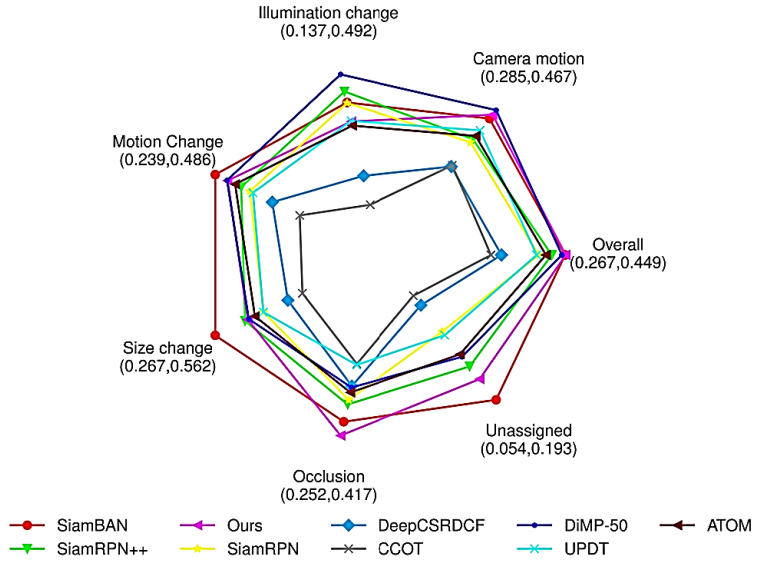

Figure 4. *The following visual characteristics of EAO on VOT2018 were compared: camera motion, illumination change, occlusion, size change, and motion change.*

4.3.2. Result on UAV123

Our tracker is compared to seven state-of-the-art approaches, including SiamRPN [2], DaSiamRPN [3], SiamBAN [6], SiamCAR [5], ECO [35], and SiamLT. As illustrated in figure 5, our tracker outperforms all other trackers in terms of success and precision, scoring 63.2 percent for success and 83.3 percent for precision. Our tracker outperforms recent trackers SiamCAR [6] and SiamCAR [5] by 0.1 percent and 0.9 percent, respectively, in terms of AUC. In comparison to the SiamBAN and SiamCAR trackers, our tracker achieves the best results through the use of a simpler and faster network. These trackers are trained on a larger dataset (additional LaSOT [28], YouTube-BB [29]), with three backbone layers as input features.

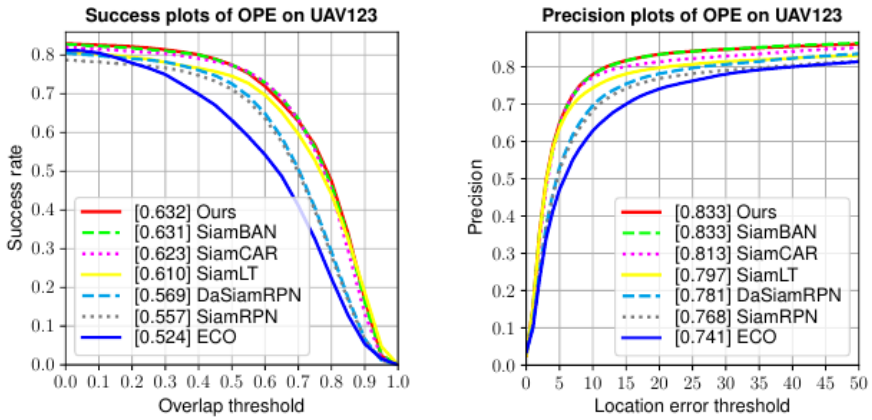

Figure 5. *Comparisons between the state-of-the-art tracker on UAV123 and the state-of-the-art tracker in terms of success and precision plots of OPE.*

4.4. Ablation study

To compare with cross-correlation-based methods, we replace the FM with the DW-Xcorr layer [4], which has the best performance among cross-correlation-based methods. As shown in table 2, on VOT2018, the FM improved the EAO score by 7.3% when compared to DW-Xcorr.

Table 2. *Quantitative comparison results of our trackerand its variants with different fusion mechanism on VOT2018.*

Fusion	**Accuracy** (↑)	**EAO**(↑)
DW-Xcorr	0.591	0.374
FM	**0.591**	**0.447**

5. CONCLUSIONS

This paper presents a novel Siamese attention network for visual object tracking. We present an attention fusion mechanism that generates fusion features by matching template and search features at the pixel level. Additionally, a prediction head anchor-free network is used to improve the accuracy of the tracking. The new Siamese network may enhance robustness against occlusion and camera motion. Extensive experiments on the VOT2018 and UAV123 benchmarks demonstrate that our method achieves a new state-of-the-art performance at 42 frames per second in real-time.

REFERENCES

[1]. L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "*Fully-convolutional siamese networks for object tracking*", in ECCV Workshops, (2016).

[2]. B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "*High performance visual tracking with siamese region proposal network*", in CVPR, (2018).

[3]. Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "*Distractor-aware siamese networks for visual object tracking*", in ECCV, (2018).

[4]. B. Li, W. Wu, Q. Wang, F. Y. Zhang, J. L. Xing, and J. J. Yan, "*SiamRPN++: Evolution of siamese visual tracking with very deep networks*", in CVPR, (2019).

[5]. D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "*SiamCAR: Siamese fully convolutional classification and regression for visual tracking"*, in CVPR, (2020).

[6]. Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "*Siamese box adaptive network for visual tracking*", in CVPR, (2020).

[7]. Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "*Deformable siamese attention networks for visual object tracking"*, in CVPR, (2020).

[8]. Z. Tian, C. Shen, H. Chen, and T. He, "*FCOS: Fully convolutional one-stage object detection"*, in ICCV, pp. 9626–9635, (2019).

[9]. H. Law, Y. Teng, O. Russakovsky, and J. Deng, "*SiamCorners: Siamese Corner Networks for Visual Tracking*", in arXiv:1904.08900, (2021).

[10]. H. Law and J. Deng, "*Cornernet: Detecting objects as paired keypoints"*, in CVPR, (2018).

[11]. Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "*Learning attentions: residual attentional siamese network for high performance online visual tracking*", in CVPR, (2018).

[12]. S.Q. Ren, K.M. He, R. Girshick, and J. Sun. "*Faster r-cnn: Towards real-time object detection with region proposal networks*", in NIPS, (2015).

[13]. K. He,X. Zhang, S. Ren, J. Sun, "*Deep residual learning for image recognition*", in CVPR, (2016).

[14]. I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "*Attention augmented convolutional networks"*, in ICCV, (2019).

[15]. P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "*Stand-alone self-attention in vision models*", in NIPS, (2019).

[16]. J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "*Dual attention network for scene segmentation*", in CVPR, (2019).

[17]. J. Choi, J. Kwon, and K. M. Lee, "*Deep meta learning for real-time target-aware visual tracking*", in ICCV, (2019).

[18]. K. Wang, B. Wu, P. Zhu, P. Li, W. Zuo and Q. Hu, "*ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks*", in CVPR, (2020).
[19]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "*Imagenet classification with deep convolutional neural networks"*, in NIPS, (2012).
[20]. H. Liu, F. Liu, X. Fan, and D. Huang, "*Polarized self-attention: towards high-quality pixel-wise regression*", in in arXiv:2107.00782, (2021).
[21]. J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "*Unitbox: An advanced object detection network*", in Proceedings of the 24th ACM international conference on Multimedia, pp. 516-520, (2016).
[22]. M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, and et al, "*The sixth visual object tracking vot2018 challenge results*", ECCV Workshops, (2018).
[23]. M. Muller,N.Smith, B. Ghanem, "*A benchmark and simulator for uav tracking*", in ECCV, (2016).
[24]. Y. Wu, J. Lim, M. Yang, "*Online object tracking: A benchmark*", in CVPR, pp.2411-2418, (2013).
[25]. T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "*Microsoft coco: Common objects in context*", in ECCV, pages 740–755, Springer, (2014).
[26]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, "*ImageNet Large Scale Visual Recognition Challenge*", in IJCV, (2015).
[27]. L. Huang, X. Zhao, and K. Huang, "*GOT-10k: A large high-diversity benchmark for generic object tracking in the wild*", in IEEE Transactions on Pattern Analysis and Machine Intelligence, (2019).
[28]. H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "*LaSOT: A high-quality benchmark for large-scale single object tracking*", (2018).
[29]. E. Real, J. Shlens, S Mazzocchi, X. Pan, and V. Vanhoucke, *"YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video"* in CVPR, (2017).
[30]. G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "*Learning Discriminative Model Prediction for Tracking*", in ECCV, (2019).
[31]. M. Danelljan, G. Bhat, F.S. Khan, and M. Felsberg, "*Atom: Accurate tracking by overlap maximization*", in CVPR, (2019).
[32]. G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "*Unveiling the power of deep tracking*", in ECCV, (2018).
[33]. A. LuNežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan, "*Discriminative correlation filter tracker with channel and spatial reliability*", in IJCV, (2018).
[34]. M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "*Beyond correlation filters: learning continuous convolution operators for visual tracking*", in ECCV, (2016).
[35]. M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "*Eco: Efficient convolution operators for tracking*CVPR, (2017).

TÓM TẮT

Mạng Siamese theo dõi đối tượng trực quan trong thời gian thực sử dụng cơ chế tự chú ý và không dùng neo

Trình theo dõi dựa trên Siamese đã chứng minh hiệu suất vượt trội trong việc theo dõi các đối tượng trực quan. Phần lớn các trình theo dõi hiện tại tính toán các đặc trưng của mẫu mục tiêu và hình ảnh tìm kiếm một cách độc lập, sau đó ước tính quy mô và tỷ lệ khung hình của mục tiêu bằng cách sử dụng lược đồ tìm kiếm đa tỷ lệ hoặc các hộp liên kết được xác định trước. Bài báo này đã đề xuất một mạng chú ý Siames để theo dõi các đối tượng trực quan. Một cơ chế kết hợp sự chú ý được tạo ra bằng cách sử dụng đối sánh cấp pixel của mẫu và các đặc trưng ảnh chứa đối tượng. Mô hình được đề xuất không sử dụng neo, nên đơn giản và hiệu quả. Các thử nghiệm mở rộng trên các bộ dữ liệu theo dõi trực quan VOT2018 và UAV123 chứng minh rằng trình theo dõi của chúng tôi hoạt động ở tốc độ 42 khung hình/giây và đạt được hiệu suất hiện đại.

Từ khoá: Theo dõi đối tượng; Cơ chế tự chú ý; Cơ chế không dùng neo.