

RESEARCH ARTICLE

SiamMFF: UAV Object Tracking Algorithm Based on Multi-Scale Feature Fusion

YANLI HOU¹, XILIN GAI¹, XINTAO WANG², AND YONGQIANG ZHANG^{1,3}¹School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang 050018, China²China Mobile Hebei Company Ltd., Zhangjiakou Branch, Beijing 100032, China³Hebei Technology Innovation Center of Intelligent IoT, Shijiazhuang 050018, China

Corresponding author: Xilin Gai (1643056315@qq.com)

This work was supported in part by the Key Research and Development Program of Hebei Province under Grant 21355901D.

ABSTRACT UAVs have entered various fields of life, and object tracking is one of the key technologies for UAV applications. However, there are various challenges in practical applications, such as the scale change of video images, motion blur and too high shooting angle leading to the tracked objects being too small, resulting in poor tracking accuracy. To cope with the problem that small targets are poorly tracked by UAVs due to less effective information output from the deep residual network, a SiamMFF tracking method that introduces an efficient multi-scale feature fusion strategy is proposed. The method aggregates features at different scales, and at the same time, replaces the ordinary convolution with deformable convolution to increase the sense field of convolution operation to enhance the feature extraction capability. The experimental results show that the proposed algorithm improves the success rate and accuracy of small target tracking.

INDEX TERMS Siamese network, object tracking, unmanned aerial vehicle(UAV), deformable convolution, multi-scale feature fusion.


I. INTRODUCTION

In recent years, with the rapid development of the civilian unmanned aerial vehicle (UAV) industry, the application areas of UAV expand from traditional mapping and agriculture to express delivery, security, and other livelihood areas. Its barrier of entry is also decreasing, and many photography enthusiasts also use UAVs to assist in tracking photography.

Currently, the object tracking algorithms [1], [2], [3] can be classified into generative and discriminative algorithms. Discriminative object tracking algorithms are further divided into correlation filtering class algorithms and deep learning class algorithms. The original correlation filter-like object tracking algorithm is MOSSE (Minimum Output Sum of Squared Error), proposed by Valmadre et al. [4]. It learns a stable tracker from the first frame of the video. It multiplies the tracker with subsequent frames in the frequency domain, and then converts the result into a score response map in the time domain. This map helps to discern the object's location based

on the score. Henriques et al. [5] proposed CSK (Circulant Structure Kernels) to introduce kernel functions to the tracker for the first time. Instead of employing the traditional particle sampling method, it utilizes a circular matrix-based sampling approach. Henriques et al. [6] proposed KCF (Kernel Correlation Filter) algorithm. It introduces HOG features to extend the single channel features based on CSK. Li et al. [7] proposed the SAMF (Scale Adaptive with Multiple Features Tracker) algorithm. It integrates scale adaptive and kernel correlation filtering techniques to solve the problem caused by using fixed scales of KCF, which can result in decreased accuracy when the object experiences deformation, occlusion, or other challenging scenarios.

In deep learning class algorithms, the object tracking algorithms based on the Siamese network perform well [7], [8], [9], [10], [11], [12], [13], [14]. In 2016 Li et al. [8] proposed SiamFC (Fully-Convolutional Siamese Networks), an object tracking algorithm based on Siamese network structure, which has two network structures with the same template branch and search branch. After extracting video frame features from these two branches, template features

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson .

are employed as the convolution kernel to correlate with the search features. This process generates a response score map, and the position with the highest score is highly likely to be the center of the object. In 2018, SiamRPN (Siamese Region Proposal Network) [15] integrated the RPN (Region Proposal Network) detection head from the Fast RCNN [13] detection algorithm into SiamFC. It determined whether each anchor box contained the object and conducted position regression when needed. DaSiamRPN [16] improved the training data for SiamRPN to address the problem of imbalanced training samples and allowed the SiamRPN network to adapt to long-term tracking. On the other hand, SiamMask [17] integrated both tracking and segmentation, leading to greater tracking accuracy than DaSiamRPN, albeit with a slightly reduced speed to 55fps. Li et al. [18] summarized the reason why the Siamese network cannot use deep convolutional networks. The lack of translational invariance invited the network to learn a positional bias, causing the network to believe that the object must exist at the center of the image. Based on this, Li et al. proposed a strategy of uniform sampling around the center point of the image, and proposed the SiamRPN++ (Siamese Region Proposal With Very Deep Networks). Guo et al. [19] proposed SiamDW (Deeper And Wider Siamese Networks) using a novel residual structure. GradNet [20] employed gradient descent to adjust the template, addressing the issue of the template's inapplicability when the object experiences significant deformation or severe occlusion. In contrast, the above Siamese network tracking algorithm relies on a fixed anchor frame to handle changes in object scale and aspect ratio. However, configuring the anchor frame's parameters requires manual tuning, which increases the algorithm's complexity. Therefore, Siamese tracking algorithms with anchorless frame mechanisms have gradually become popular recently. Mueller et al. [21] proposed SiamCAR (Siamese-Based Classification And Regression Network), which is a tracking framework that does not require anchor frames. They also introduced central branches, which are characterized by a simple structure and can effectively reduce the number of parameters.

While SiamCAR demonstrates excellent performance, as other Siamese object tracking algorithms [22], it typically conducts basic operations like feature concatenation and dimensionality reduction to combine output features from various network layers. It often does not fully integrate information from each layer, leading to a suboptimal utilization of available data. To address this problem, this paper introduces SiamMFF, featuring an efficient multi-scale feature fusion strategy. SiamMFF employs a novel multi-scale feature fusion method to comprehensively integrate information from various scales. It replaces the standard convolution in the feature extraction network with deformable convolution, thereby expands the receptive field of the convolution process. The proposed algorithm is tested on the UAV object tracking dataset UAV123 and the common datasets OTB2015 and VOT2018, and it is experimentally verified that SiamMFF has a better performance than SiamCAR.

II. SIAMMFF

SiamMFF is based on SiamCAR and consists of three parts: a feature extraction subnetwork with a deformable convolutional network as its backbone, an Efficient Multi-scale Feature Fusion Network (EMSNet), and a region regression subnetwork. The specific structure is shown in Figure 1.

The feature extraction subnetwork aims at matching the object template with the search region to facilitate global inter-correlation and obtain the response within the search region. The EMSNet is responsible for integrating response feature maps from different levels of the search area, which are obtained from the feature extraction subnetwork. This integration is achieved through a cross-scale feature fusion strategy from top to bottom and from left to right, to maximize the relationships between different object features. The region regression subnetwork is responsible for binary classification, distinguishing background and objects, and performing location regression based on the global information feature maps generated by the EMSNet.

A. FEATURE EXTRACTION SUBNET

The input image for the template branch is $127 \times 127 \times 3$ and the input image for the search branch is $255 \times 255 \times 3$. Both branches use a deformable convolutional residual network, which introduces deformable convolutions embedded in ResNet50 instead of normal convolutions so that the region of convolution always covers around the object. During the inter-correlation matching process, the 3rd, 4th, and 5th convolutional modules of SiamCAR are replaced by the 2nd, 3rd, and 4th modules. This modification retains a shallower network that holds more information related to smaller objects, and eliminates the deeper network layer which yields greater semantic information and involves a significant number of parameters. This adjustment is more suitable for UAV object tracking.

1) DEFORMABLE CONVOLUTION

Normal convolution operation [23] is implemented by fixed size convolution kernel, and the perceptual field on the feature map is fixed. When an object deforms due to the limitations of the perceptual field, it is often difficult to fully extract object features, so a method of adjusting the convolutional field is needed.

2) OFFSET LEARNING

The output offset of $H \times W \times 2H$ is obtained by performing normal convolution of the input $H \times W$ feature map with the convolution kernel, where $2N$ denotes the offsets in both X and Y directions.

Set x as a feature map, t_0 represents a point in the feature map, t_n represent the offset of the feature points in the feature map, Δt_n is the learning offset, usually a decimal, and the sampling coordinates are located by means of bilinear difference, and the output is the position $x(n)$ of the point

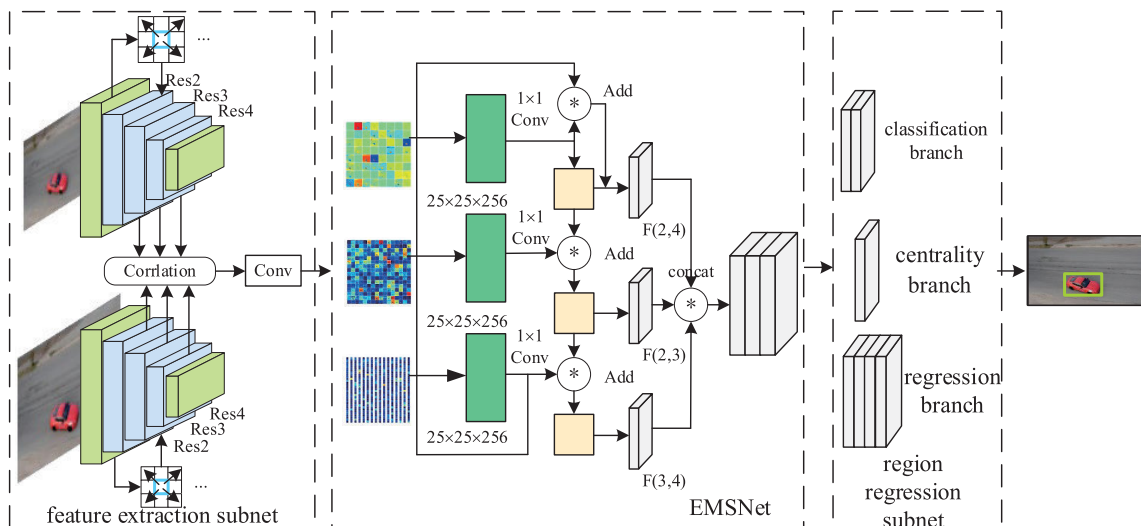


FIGURE 1. SiamMFF Network Architecture.

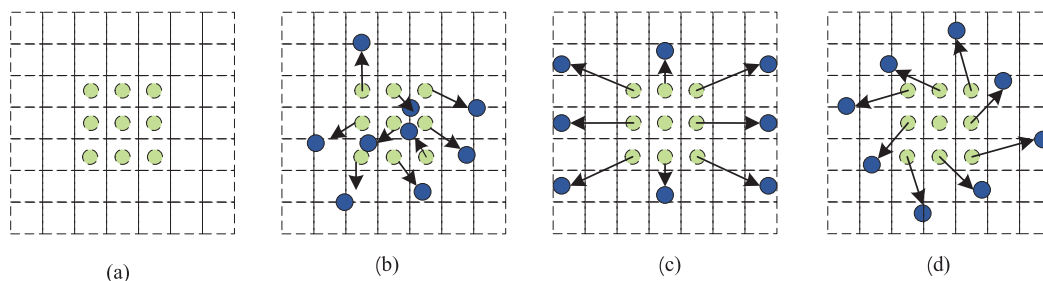


FIGURE 2. Illustration of deformable convolution samplings. Deformable convolution differs from normal convolution in that it can dynamically adjust the convolution field according to the shape of the object. Figure 2 (a) shows the normal convolution, and Figure 2 (b), (c), and (d) shows the deformable convolution learning a perceptual field that is more adapted to the object shape by the offset when the object has deformation, scale change, or rotation.

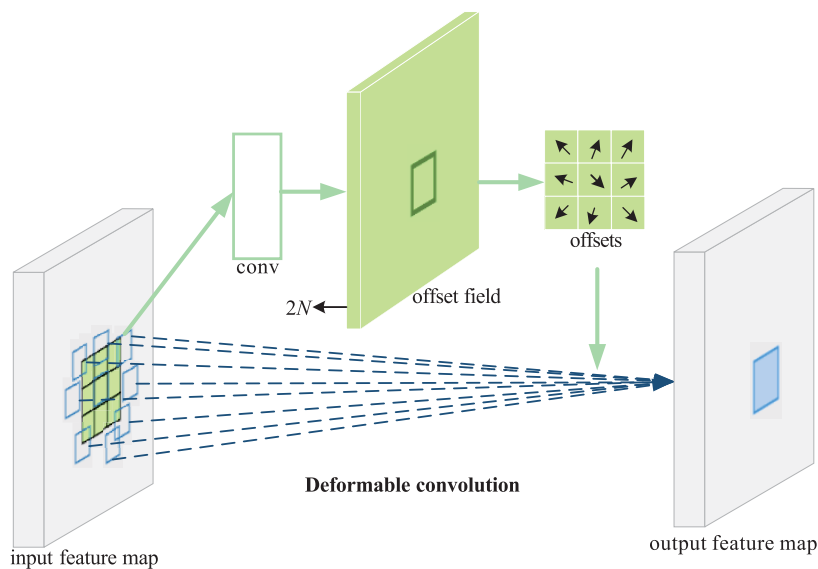


FIGURE 3. Deformable convolution. As an example, feature maps are subjected to 3×3 deformable convolution, which is divided into two parts: offset learning and deformable convolution based on input feature maps and offsets.

within one pixel of the horizontal and vertical coordinates from point t , The details are as follows:

$$x(n) = x(t_0 + t_n + \Delta t_n) \quad (1)$$

$$G(q, t) = g(q_x, t_x) \cdot g(q_y, t_y) \quad (2)$$

$$g(q_i, t_j) = \max(0, 1 - |q_i - t_j|) \quad i \in x, y \quad j \in x, y \quad (3)$$

where $G(q, t)$ is the learned weight, q is the reference point, q_x and q_y denote the x, y coordinates of reference point q , t is the feature point, t_x and t_y denote the x, y coordinates of feature point t .

3) DEFORMABLE CONVOLUTION BASED ON OFFSETS

Based on the feature map with offsets, normal convolution is performed to obtain a deformable convolution based on the original feature map. It can be expressed as:

$$y(t_0) = \sum_{t_n \in R} w(t_0) \cdot x(t_0 + t_n + \Delta t_n) \quad (4)$$

where w is the convolution kernel and R is the sampling space for the convolution operation.

During the training process, the convolution kernel for generating output features and the convolution kernel for generating offsets are learned simultaneously.

B. EMSNET

Lower-level features with higher resolution contain more detailed positional information, but have less semantic and often have more noise due to fewer convolutions. In contrast, higher-level features offer stronger semantic information, but with lower resolution, their ability to perceive fine details is poor. SiamCAR combines mutual correlation feature maps produced by the last three network layers of ResNet50. Initially, it concatenates three mutual correlation feature maps, each with a dimension of 256, to create a single feature map with a dimension of 256×3 . Then, it downscales this combined feature map to 256 to achieve fusion. This fusion process involves continuous weight adjustments through training to optimize the fusion results. However, it's worth noting that this fusion method may impact tracking speed.

To tackle the aforementioned challenges, this paper introduces a straightforward and effective feature fusion network called EMSNet. EMSNet combines learnable weights to consider the significance of various input features, while implementing multi-scale feature fusion strategies from top to bottom and from left to right.

The structure of EMSNet is shown in Figure 4. To enhance the acquisition of shallow information, the top-down features of F2, F3, and F4 are first downscaled by 1×1 convolution and then fused with the next layer of features, and outputs the fused features. Three adaptive feature maps F(2,3), F(2,4), and F(3,4) are obtained through Add, and then stitched together by Concat. The process can be expressed as follows:

$$F(i, j) = R[F(i)] * F(j) \quad i, j \in 2, 3, 4 \quad (5)$$

where $F(i)$ and $F(j)$ denote the three cross-correlation feature maps of F2, F3, and F4, R denotes the ReSize operation used to adjust the feature maps at different levels to one scale, $*$ is Add feature aggregation. The association of information in the close-by feature maps is achieved by two-by-two aggregation of features at different scales, thus retaining more small object information. Further, the three cross-scale aggregated feature maps are fused together in a Concat manner, which can be expressed as:

$$M = \text{Cat}(F(2, 3), F(2, 4), F(3, 4)) \quad (6)$$

where M is the multiscale features output from EMSNet and Cat is the Concat feature fusion operation.

III. TRACKING PROCESS

One frame of the tracking video sequence is used as the template image Z_1 , the manual boxed object box is used as the position l_1 of the first frame, and the subsequent video sequences are used as the search images, and the SiamMFF algorithm process is shown in Table 1.

(1) The model is loaded and the template image $\varphi(Z_1)$ and search images are input into the template branch and the search branch.

(2) The deformable convolutional network extracts template features $\varphi(Z_1)$ and search image features $\varphi(X_1)$ from the output of the 2nd, 3rd and 4th convolutional modules of ResNet50, and outputs the intercorrelation response maps of objects in the search image at different levels by means of Eq.(4) intercorrelation, respectively.

(3) EMSNet fuses the response maps together top-down two-by-two by Eq. (5) and outputs three neighboring feature fusion maps. Aggregates of global features by Eq. (6).

(4) Regional regression subnet is used to classify, calculate centrality, and perform position regression on the global fusion feature map, obtains the classified feature map $A_{w \times h \times 2}^{cls}$, centrality feature map $A_{w \times h \times 4}^{cen}$ and regression feature map $A_{w \times h \times 4}^{reg}$, respectively.

(5) $A_{w \times h \times 2}^{cls}$ locates the object in the search image, $A_{w \times h \times 4}^{reg}$ gets the prediction frame, and $A_{w \times h \times 4}^{cen}$ constrains the position away from the center point.

(6) The object position is optimized by scale penalty, aspect ratio penalty, and cosine window penalty. $A_{w \times h \times 2}^{cls}$ outputs the final object prediction frame.

(7) Repeat steps (2) to (6) until the last frame of tracking video is completed.

IV. EXPERIMENTS

A. EXPERIMENTAL ENVIRONMENT

The experimental environments for training and testing are shown in Table 2.

To improve the convergence speed of the model and the accuracy of the gradient estimation, the proposed model is trained by mini-batch gradient descent on the UAV123 dataset. A total of 60 epochs are trained, with a learning rate of 0.001 for the first 15 epochs, and 0.005 to 0.0005 for the last 45 epochs with exponential decay.

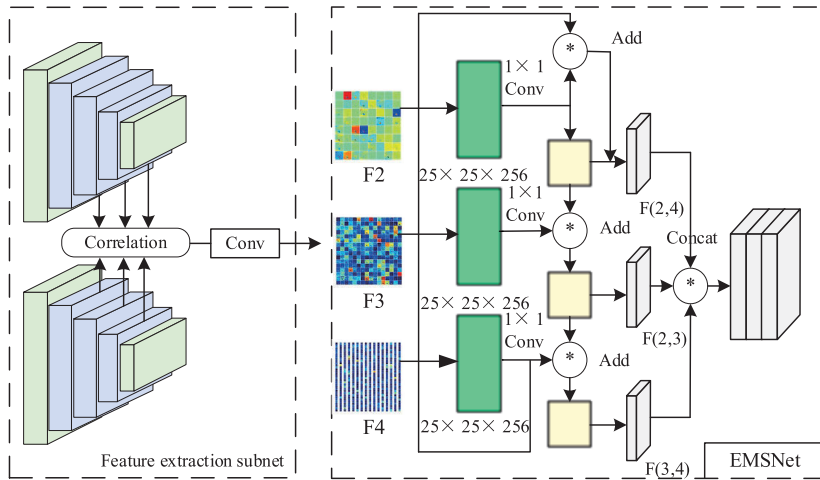


FIGURE 4. EMSNet. F2, F3, and F4 represent the mutual correlation feature maps of Res2, Res3, and Res4, respectively.

TABLE 1. SiamMFF tracking process.

Input: First frame template image Z_1 , Box the object location l_1 , Video Sequence S , Parameters.
Output: t_{th} search frame object position l_t .
1. Load tracker.
2. $\varphi(Z_1) \leftarrow$ Siamese network φ extraction of template frame Z_1 features.
3. For video frame t in S do:
4. $X_1 \leftarrow$ Crop the search area at the object location predicted in the previous search frame.
5. $\varphi(X) \leftarrow$ Output feature map of the current frame X_t .
6. $F1, F2, F3 \leftarrow$ Mutual relations ($\varphi(Z_1), \varphi(X)$).
7. $F(2, 3), F(3, 4), F(2, 4) \leftarrow$ Feature fusion [$Add * (F2, F3), Add * (F3, F5), Add * (F2, F4)$].
8. $F(2, 3, 4) \leftarrow$ Feature fusion [$Concat[F(2, 3), F(3, 4), F(2, 4)]$].
9. Classification feature map $A_{w \times h \times 2}^{cls}$ calculates the object location.
10. The centrality feature map $A_{w \times h \times 4}^{cen}$ calculates the centrality score.
11. Regression feature diagram $A_{w \times h \times 4}^{reg}$ calculating the final prediction box.
12. Smoothing the position shift and scale change of the prediction box by applying window and proportional penalty.
13. Output prediction box of the current frame.
14. Update the size of the prediction box by linearly interpolating the object state in the previous frame.
15. End.

TABLE 2. Experimental environment.

Experimental environment	Configuration
CPU	Intel® XeonW-2245 CPU, 32GB
GPU	NVIDIA RTX 3080
System	Windows10
Software development environment	Python3.7, Pycharm Community 2022
Development framework	Pytorch1.4.0, torchvision0.5.0

B. DATASET

The training dataset uses UAV123. The test datasets include UAV123, VOT2018 [24], and OTB2015 [25].

The UAV123 dataset contains 91 video sequences captured by UAVs. This dataset has many challenging scenarios that ordinary tracking datasets do not have, such as out of field of view, scale changes, and changes in perspective. The UAV123 dataset can be divided into three subsets: The first subset contains 103 video sequences captured by professional UAVs equipped with stable and controllable cameras. These sequences were shot at heights of 5 to 25 meters, with a frame rate of up to 96fps and resolutions ranging from 720p to 4K. All videos have been manually annotated and

provided at a 720p resolution of 30fps. The second subset includes 12 video sequences captured by lower-cost UAVs using unstable cameras. These sequences have lower quality and resolution, as well as reasonable levels of noise. Like the first subset, they are also manually annotated. The third subset includes 8 synthesized video sequences generated using the proposed UAV simulator. In these sequences, objects follow predefined trajectories and are rendered using the Unreal Game Engine. They are automatically annotated at 30fps and include target masks/segmentation information.

The VOT2018 dataset contains 60 video sequences ranging from 2,000 to 20,000 frames each. Its evaluation metrics are EAO (Expect Average Overlaprate), Accuracy, and Robustness which are used to evaluate the stability of the tracker, and the Robustness value is inversely proportional to the stability. EAO categorizes all sequences based on their length and averages the measurement accuracy of sequences with the same length. It reflects the relationship between the length and average accuracy. The OTB2015 dataset contains 11 common challenge attributes in object tracking tasks, such as illumination change, viewpoint variation, camera shaking,

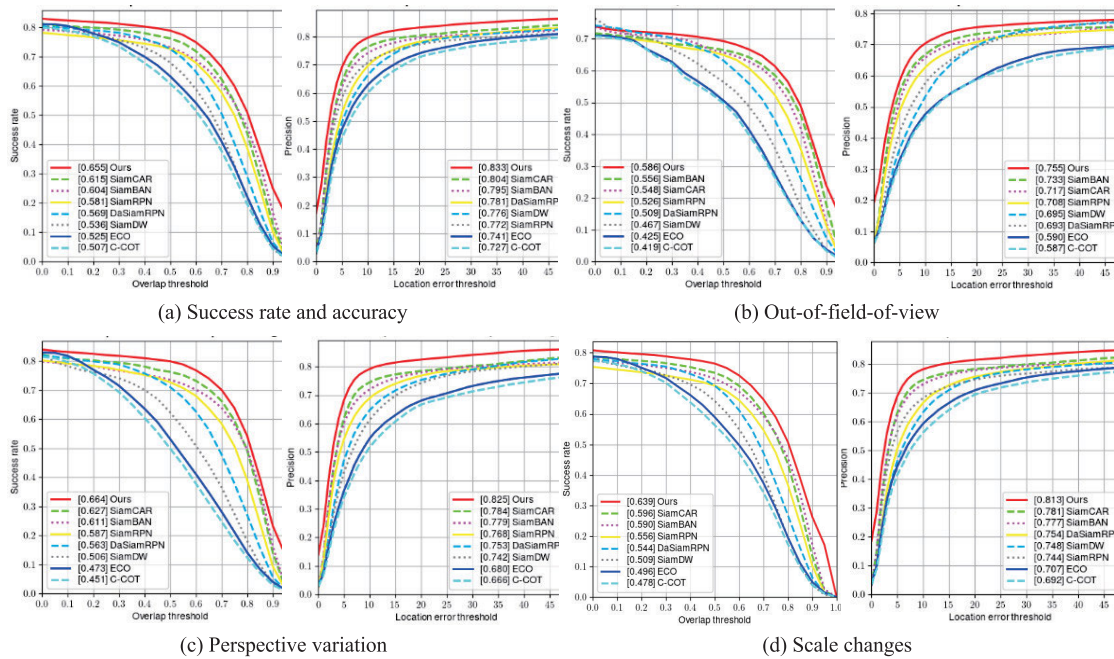


FIGURE 5. Comparison of algorithms for the UAV123 dataset: (a) Shows the success rate and accuracy of SiamMFF on UAV123. (b) Demonstrates the performance of each algorithm when dealing with out-of-field-of-view challenges. Out-of-field-of-view indicates that some or all of the object’s features are not in the image. (c) Contrasts for the challenge of perspective variation. Perspective change is a change in the appearance of the object in the video caused by a change in the posture of the UAV or an adjustment in the angle of the camera. (d) Shows the performance of each algorithm under scale variation.

TABLE 3. Comparison of algorithms for the UAV123 dataset.

Tracking algorithms	Success rate and accuracy		Out-of-field-of-view		Perspective variation		Scale changes	
	Success	Precision	Success	Precision	Success	Precision	Success	Precision
Ours	0.655	0.833	0.586	0.755	0.664	0.825	0.639	0.813
SiamCAR	0.604	0.759	0.556	0.733	0.611	0.779	0.590	0.777
SiamBAN	0.604	0.759	0.556	0.733	0.611	0.779	0.590	0.777
SiamRPN	0.581	0.772	0.526	0.708	0.587	0.768	0.556	0.744
DaSiamRPN	0.569	0.781	0.509	0.693	0.563	0.753	0.544	0.754
SiamDW	0.536	0.776	0.467	0.695	0.506	0.742	0.509	0.748
ECO	0.525	0.741	0.425	0.590	0.473	0.680	0.496	0.707
C-COT	0.507	0.727	0.419	0.587	0.451	0.666	0.478	0.692

and low resolution, and it uses success rate and accuracy as evaluation metrics just like the UAV123 dataset.

C. COMPARISON EXPERIMENT

1) UAV123 DATASET

The SiamMFF algorithm is compared with the seven algorithms on UAV123, namely SiamCAR [21], SiamBAN [26], SiamRPN [9], DaSiamRPN [16], SiamDW [19], ECO [20], and C-COT [27] on UAV123. The experimental results are shown in Figure 5 and Table 3.

Figure 5 (a) shows that the success rate and accuracy of SiamMFF on UAV123 are 0.655 and 0.833, respectively, which are 4% and 2.9% higher than SiamCAR.

Figure 5 (b) shows that compared with SiamCAR, the success rate and accuracy of SiamMFF have been improved by 3% and 2.2% respectively, demonstrating the effectiveness of the proposed multiscale feature aggregation network EMSNet combined with the deformable convolution method when objects exceed the field of view.

Figure 5 (c) shows that compared with the SiamCAR benchmark algorithm, the success rate and accuracy have been improved by 2.5% and 4.1% respectively, ranking first among all comparison algorithms, indicating that the SiamMFF algorithm can cope with the problem of passive deformation of objects caused by UAVs.

Figure 5 (d) shows that SiamMFF can better cope with scale variations. The success rate and accuracy have been improved by 4.3% and 3.2%, respectively, compared with SiamCAR, and are superior to other comparison algorithms.

The results demonstrate the SiamMFF can better deal with challenging out-of-field-of-view, perspective variation and scale changes, which benefit from our EMSNet.

2) OTB2015 DATASET

To further verify the tracking performance of SiamMFF in general scenario, nine comparison algorithms SiamRPN++ [28], SiamCAR [21], MDNet [29], DaSiamRPN [16], SiamRPN [9], GradNet [18], DeepSRDCF [30], SRDCF

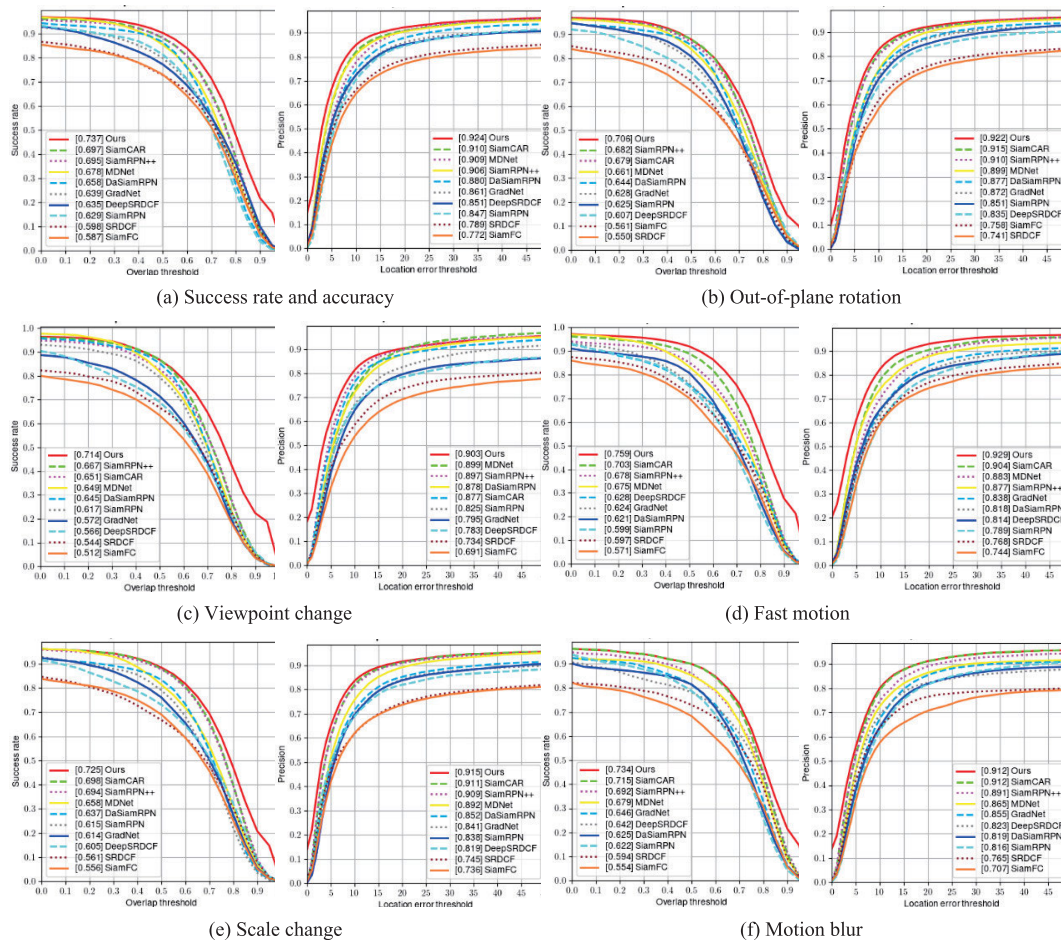


FIGURE 6. Comparison of algorithms for the OTB2015 dataset.

[29], and SiamFC [8] were tested on the OTB2015 dataset. The performance of the ten algorithms was analyzed in terms of two metrics, success rate and accuracy, as well as five different challenge attributes of out-of-plane rotation, viewpoint change, fast motion, scale change, and motion blur. The experimental results are shown in Figure 6 and Table 4. It can be seen from Figure 6 (a) that SiamMFF improves by 4% and 1.4%, respectively, compared with SiamCAR.

From Figure 6 (b), it can be seen that the success rate and accuracy of SiamMFF are improved by 2.7% and 0.7%, respectively, compared with SiamCAR, and it performs the best among all the comparison algorithms. Figure 6 (c) to (f) shows the experimental results under other challenge attributes, and it can be seen that SiamMFF also performs better than the other nine algorithms.

3) VOT2018 DATASET

Based on the fact that VOT2018 has richer evaluation metrics, SiamMFF was quantitatively analyzed with different comparison algorithms on the VOT2018 dataset to verify its performance. Table 5 shows the performance of SiamMFF and four comparison algorithms, including SiamCAR [21], SiamVGG [30], C-COT [27], and SiamFC [8].

From Table 5, it can be seen that compared with the benchmark algorithm SiamCAR, the accuracy improvement is more significant, reaching 17.7%, and it also outperforms the comparison algorithms in terms of robustness and EAO.

The results on OTB2015 and VOT2018 datasets demonstrate that the SiamMFF is also valid to deal with the tracking challenges in the two datasets, which illustrates our proposed SiamMFF has a good generalizability.

D. ANALYSIS OF VISUAL TRACKING

In order to test the tracking performance of SiamMFF more intuitively, it was tested with SiamBAN [26], SiamCAR [21], and SiamVGG [30] on four video sequences of group1_2, boat6, bike3, and car2_s from the UAV123 dataset, and the actual tracking effect of the four algorithms was compared by visual images, as shown in Figure 8.

From Figure 8 (a), it can be seen when the UAV is tracking the crowd, the crowd is in a dispersed state at the beginning and the all used algorithms can track the object accurately. At frame 124, the crowd gathers together and the SiamCAR and SiamVGG algorithms have drifts, mistaking some other features as objects, while SiamMFF still maintains stable tracking. At frames 485 and 515, the crowd disperses again,

TABLE 4. Comparison of algorithms for the OTB2015 dataset.

Tracking algorithms	Success rate and accuracy		Out-of-plane rotation		Viewpoint change		Fast motion		Scale change		Motion blur	
	Success	Precision	Success	Precision	Success	Precision	Success	Precision	Success	Precision	Success	Precision
Ours	0.737	0.924	0.706	0.922	0.714	0.903	0.759	0.929	0.725	0.915	0.734	0.912
SiamCAR	0.697	0.910	0.679	0.915	0.651	0.877	0.703	0.904	0.698	0.911	0.715	0.912
SiamRPN++	0.695	0.906	0.682	0.910	0.667	0.897	0.678	0.877	0.694	0.909	0.692	0.891
MDNet	0.678	0.909	0.661	0.899	0.649	0.899	0.675	0.883	0.658	0.892	0.679	0.865
DaSiamRPN	0.658	0.880	0.644	0.877	0.645	0.878	0.621	0.818	0.637	0.852	0.625	0.819
GradNet	0.639	0.861	0.628	0.872	0.572	0.795	0.624	0.838	0.614	0.841	0.646	0.855
DeepSRDCF	0.635	0.851	0.607	0.835	0.566	0.783	0.628	0.814	0.605	0.819	0.642	0.823
SiamRPN	0.629	0.847	0.625	0.851	0.617	0.825	0.599	0.789	0.615	0.838	0.622	0.816
SRDCF	0.598	0.789	0.550	0.741	0.544	0.734	0.597	0.768	0.561	0.745	0.594	0.765
SiamFC	0.587	0.772	0.561	0.758	0.512	0.691	0.571	0.744	0.556	0.736	0.554	0.707

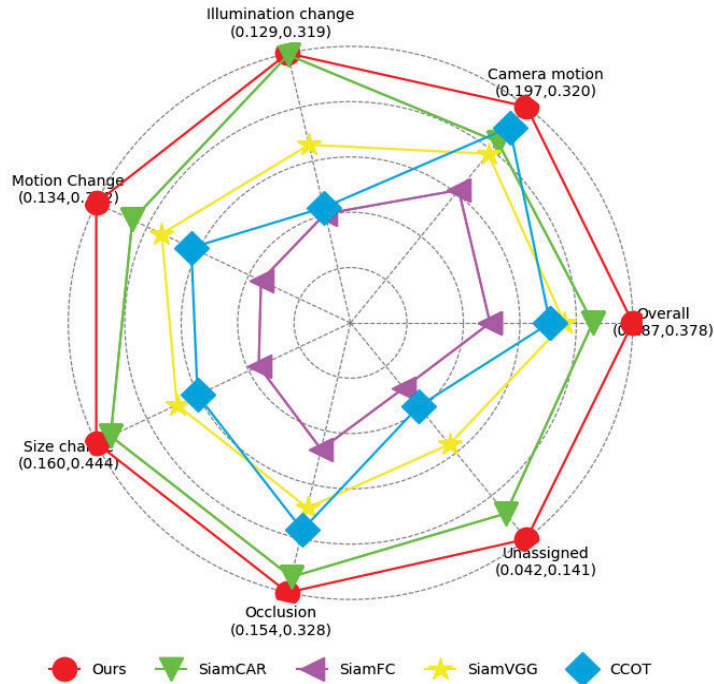


FIGURE 7. Contrast algorithm challenge attribute radar map: The figure shows the EAO values of the five algorithms on the VOT2018 dataset under the five challenging attributes of illumination change, motion change, camera motion, size change and occlusion, the closer to the periphery, the larger the EAO value, the better the performance, and unassigned denotes the video frames without the above five challenging attributes.

TABLE 5. Performance comparison of five algorithms on VOT2018 dataset.

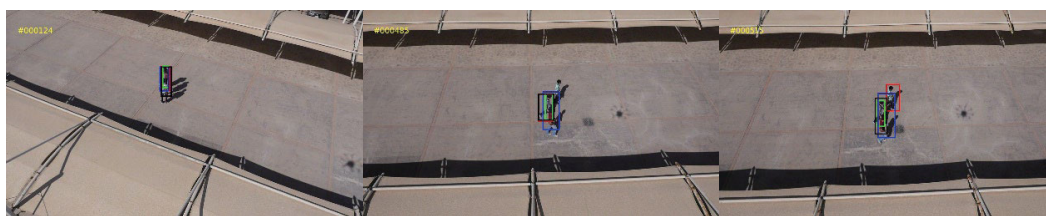
Tracking algorithms	Accuracy	Robustness	EAO
Ours	0.542	0.27	0.378
SiamCAR	0.365	0.033	0.324
SiamVGG	0.531	0.325	0.286
C-COT	0.494	0.318	0.267
SiamFC	0.503	0.585	0.187

and SiamBAN follows the wrong object completely, while SiamMFF still maintains normal tracking.

From Figure 8 (b), it can be seen that during the process of tracking ships at sea, the object scale changes greatly, there are few features of the ship in the image at frame 106, and the ship gradually becomes larger at frames 636 and 761. The prediction frames of SiamCAR and SiamVGG have poor quality and fail to surround the ship well, while SiamMFF using EMSNet can always track the ship more accurately.

From Figure 8 (c) it can be seen that when tracking the crowd, the illumination is dark and the crowd is far away from the UAV. When the object overlaps with others at frame 73, the prediction boxes of the three comparison algorithms drift with different degrees. At frame 429, SiamBAN is disturbed by similarities to treat mistakenly other pedestrians as objects. At frame 497, SiamCAR drifts more severely, while SiamMFF is more stable in tracking objects.

From Figure 8 (d), it can be seen that the UAV is tracking the car, and there is a challenge of dramatic changes in illumination in this scene, from the first frame to frame 323. The illumination changes from bright to dim and SiamCAR is the first to shift until the object is lost, SiamBAN and SiamVGG also lose the tracking object when the car enters a completely dim area. The proposed algorithm SiamMFF maintains tracking the object even in dim light.



(a) Group1_2 video Sequence



(b) Boat6 video Sequence



(c) Group3_4 video Sequence



(d) Car2_s video Sequence

— Ours — SiamBAN — SiamCAR — SiamVGG

FIGURE 8. Comparison of UAV123 dataset visualization.

V. CONCLUSION

In this paper, we introduce a UAV object tracking algorithm called SiamMFF, which utilizes an efficient multi-scale feature fusion strategy and deformable convolution. We present the overall network architecture of the SiamMFF algorithm, which employs deformable convolution to enhance the perceptual field during the convolution process, allowing for global perception of object features in the feature map. This effectively accommodates changes in object scale and captures detailed information, particularly for small objects. Additionally, we integrate an efficient multi-scale feature fusion network EMSNet, which combines deep-level semantic information with shallow-level detailed features.

Our experiments involve both qualitative and quantitative analyses of SiamMFF, and compare it with benchmark algorithm using UAV123, OTB2015, and VOT2018 datasets.

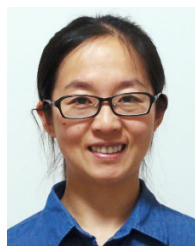
The results demonstrate that compared to SiamCAR and other comparison algorithms, SiamMFF has improved in terms of accuracy, success rate, robustness, and other key metrics. The visualization of tracking results further confirms the effectiveness of SiamMFF.

With the popularisation of UAV technology, the use of UAVs for object tracking will penetrate into various fields and play an important role in production, life and disaster prevention. The research and improvement of UAV target tracking algorithms will further expand its application scope, so the results of this article have strong practical value.

REFERENCES

[1] P. Gao, R. Yuan, F. Wang, L. Xiao, H. Fujita, and Y. Zhang, "Siamese attentional keypoint network for high performance visual tracking," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105448.

- [2] P. Gao, Q. Zhang, F. Wang, L. Xiao, H. Fujita, and Y. Zhang, "Learning reinforced attentional representation for end-to-end visual tracking," *Inf. Sci.*, vol. 517, pp. 52–67, May 2020.
- [3] P. Gao, Y. Ma, K. Song, C. Li, F. Wang, L. Xiao, and Y. Zhang, "High performance visual tracking with circular and structural operators," *Knowl.-Based Syst.*, vol. 161, pp. 240–253, Dec. 2018.
- [4] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5000–5008.
- [5] J. F. Henriques, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, Berlin, Germany: Springer, Oct. 2012, pp. 702–715.
- [6] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.
- [7] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, Cham, Switzerland: Springer, Sep. 2014, pp. 254–265.
- [8] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
- [9] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.
- [10] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFNet: Discriminant correlation filters network for visual tracking," 2017, *arXiv:1704.04057*.
- [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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4854–4863.
- [12] Q. Wang, M. Zhang, J. Xing, J. Gao, W. Hu, and S. Maybank, "Do not lose the details: Reinforced representation learning for high performance visual tracking," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.
- [13] Z. Zhu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 101–117.
- [14] M. Cen and C. Jung, "Fully convolutional Siamese fusion networks for object tracking," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3718–3722.
- [15] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 650–657.
- [16] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.
- [17] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4586–4595.
- [18] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6161–6170.
- [19] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6268–6276.
- [20] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4277–4286.
- [21] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Cham, Switzerland: Springer, Oct. 2016, pp. 445–461.
- [22] M. Kristan, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, Sep. 2018, pp. 1–52.
- [23] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.
- [24] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6667–6676.
- [25] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient convolutional network for online video understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 695–712.
- [26] M. Danelljan, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Cham, Switzerland: Springer, Oct. 2016, pp. 472–488.
- [27] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.
- [28] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 621–629.
- [29] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4310–4318.
- [30] F. J. Chen and W. Xie, "SiamVGG network target tracking algorithm with anti-occlusion mechanism," *J. Signal Process.*, vol. 36, no. 4, pp. 562–571, 2020.



YANLI HOU received the Ph.D. degree in signal and information processing from Harbin Engineering University, Harbin, China, in 2008.

She is currently an Assistant Professor with the School of Information Science and Engineering, Hebei University of Science and Technology, China. Her current research interests include wireless communication technology, electronic countermeasure technology, motion target detection, and image processing.



XILIN GAI received the degree from the School of Information Science and Engineering, Hebei University of Science and Technology, China.

His research interest includes image processing.



XINTAO WANG received the M.D. degree from the Hebei University of Science and Technology, China, in 2023.

He is currently with China Mobile Hebei Company Ltd., Zhangjiakou Branch. His research interests include artificial intelligence and object tracking and detection.



YONGQIANG ZHANG received the M.S. degree in computer application technology from the Anhui University of Technology, China.

He is currently an Assistant Professor with the School of Information Science and Engineering, Hebei University of Science and Technology, China. He is also the Deputy Director of the Hebei Technology Innovation Center of Intelligent IoT. His current research interests include complex networks and artificial intelligence technology.

• • •