Searching Region-Free and Template-Free Siamese Network for Tracking Drones in TIR Videos

Bo Huang, Zeyang Dou, Junjie Chen, Jianan Li, Ning Shen, Ying Wang, and Tingfa Xu

***Abstract*— With the growing threat of unmanned aerial vehicle (UAV) intrusions, the topic of anti-UAV tracking has received widespread attention from the community. Traditional Siamese trackers struggle with small UAV targets and are plagued by model degradation issues. To mitigate this, we propose a novel searching region-free and template-free Siamese network (SiamSRT) to track UAV targets in thermal infrared (TIR) videos. The proposed tracker builds a two-stage Siamese architecture with the former providing detection of the first-frame ground truth by using a cross-correlated region proposal network (C-C RPN) and the latter providing detection of previous-frame predictions via a similarity-learning region convolutional neural network (S-L RCNN). In both stage, global proposals are acquired by region of interest (ROI) alignment operation to break the limitation of searching region. Then, a spatial location consistency function is introduced to suppress background thermal distractors and a temporal memory bank (TMB) is utilized to avoid template update degradation problem. Further, a single-category foreground detector (SCFD) is designed to independently predict the position of the UAV target. SCFD can re-initialize the tracker without the given target in the first frame, which can help to recover the tracking failures. Comprehensive experiments demonstrate that SiamSRT achieves the best performance compared to the most advanced algorithms in the anti-UAV tracking missions.**

***Index Terms*— Anti-unmanned aerial vehicles (UAVs), Siamese network, single-category foreground detector (SCFD), temporal memory bank (TMB), thermal infrared (TIR).**

Manuscript received 27 August 2023; revised 5 October 2023 and 20 November 2023; accepted 7 December 2023. Date of publication 8 December 2023; date of current version 18 December 2023. This work was supported in part by the Youth Program of National Natural Science Foundation of China under Grant 62306049, Grant 62101032, and Grant 62174018, in part by the General Program of Chongqing Natural Science Foundation under Grant CSTB2023NSCQ-MSX0665, in part by the Key Laboratory Foundation under Grant TCGZ2020C004 and Grant 202020429036, in part by the Young Elite Scientist Sponsorship Program of China Association for Science and Technology under Grant YESS20220448, and in part by the Young Elite Scientist Sponsorship Program of Beijing Association for Science and Technology under Grant BYESS2022167. *(Bo Huang and Zeyang Dou contributed equally to this work.) (Corresponding authors: Jianan Li; Tingfa Xu.)*

Bo Huang is with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China, and also with the College of Optoelectronic Engineering and the Key Laboratory of Optoelectronic Technology and Systems of the Education Ministry of China, Chongqing University, Chongqing 400044, China (e-mail: huangbo0326@cqu.edu.cn).

Zeyang Dou is with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China, and also with Zugo Intelligent Technology Company Ltd., Shenzhen 518057, China (e-mail: douzeyang@qq.com).

Junjie Chen, Jianan Li, Ning Shen, and Ying Wang are with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China (e-mail: 3120190516@bit.edu.cn; lijianan@bit.edu.cn; shennbit@163.com; 3120215325@bit.edu.cn).

Tingfa Xu is with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China, also with the Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China, Beijing 100081, China, and also with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing 401135, China (e-mail: ciom_xtf1@bit.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TGRS.2023.3341331, provided by the authors.

Digital Object Identifier 10.1109/TGRS.2023.3341331

I. Introduction

RECENTLY, unmanned aerial vehicles (UAVs) have received a lot of attention due to their flexibility, portability, and a large number of applications, including aerial photography [1], intelligent monitoring [2], reconnaissance, and rescue [3], [4]. As the technical barriers and difficulties in modifying UAVs continue to diminish, UAVs are being used with increasing frequency to carry out illegal missions such as physical attacks (via explosives) and cyber-attacks (hacking a critical infrastructure) [5]. As a result, anti-UAV technologies, notably vision-based, have been extensively promoted to counter the potential threat of drone intrusion.

Visual object tracking, especially in thermal infrared (TIR) mode, as a fundamental step in computer vision, paves a promising path for subsequent research on anti-drone missions. Along with the significant progress of deep learning technology, Siamese networks light up the task of visual object tracking, by providing the strong capacity of learning powerful deep features [6], [7]. SiamFC [8] initially introduces the Siamese network with two shared branches for visual tracking and adopts the correlation layer to learn a decision making-based similarity evaluation. Currently, Siamese trackers have significantly advanced the state-of-the-art tracking performance on multiple well-established benchmarks and competitions by incorporating the regional proposal networks (RPNs) [9], [10], attention mechanisms [11], [12], [13], correlation filters [14], [15], [16], residual structures [17], [18], [19], region convolutional neural networks [20], [21], [22], and transformers [23], [24]. Nevertheless, these trackers are designed for RGB tracking. While the potential of TIR tracking in some special scenarios such as night and fog should not be ignored. As is obvious, TIR tracking technique is better suited to the low-light scenarios, thus catering to all-weather requirements.

When tracking UAVs in TIR videos, targets tend to be smaller in scale than conventional tracked objects. Traditional tracking methods, which search the target in a neighborhood several times the size of the target object, are very prone to tracking failures in the face of small-scale UAVs, due to the fact that the target can easily escape outside the searching region. Thereby the full-image search strategy is urgently

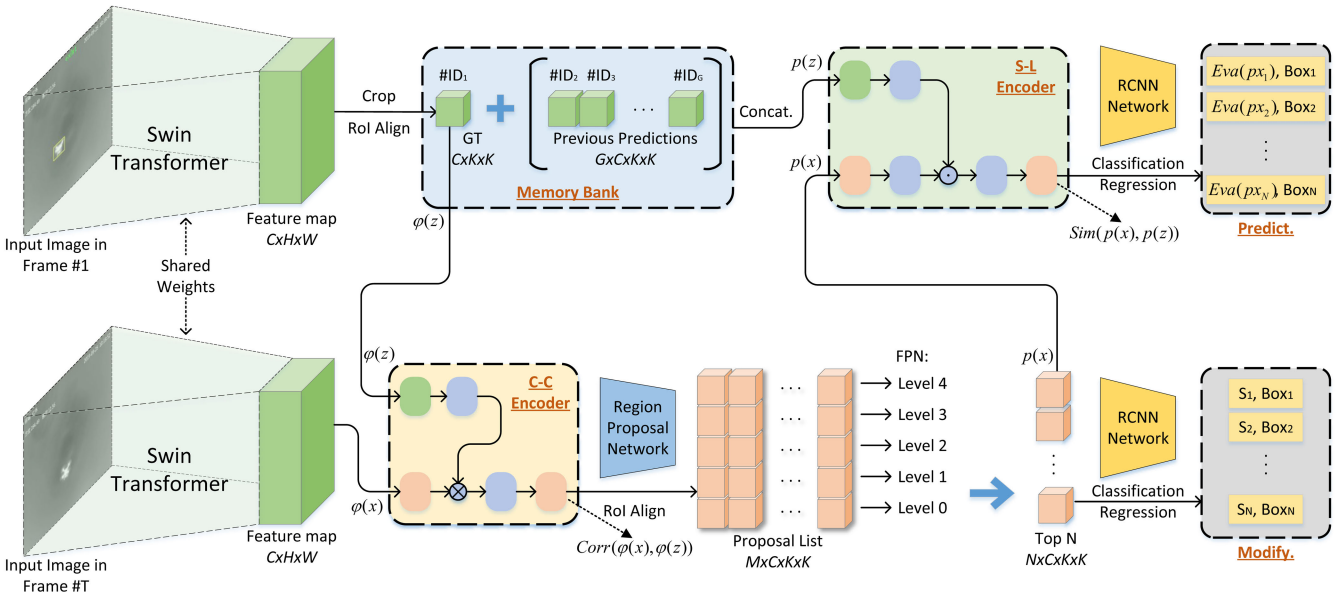


Fig. 1. Overview of the SiamSRT network, which contains a template branch and a testing branch. The proposed network takes the whole images as inputs and extracts the 256-channel Swin transformer features. The feature extractor is then followed by a two-stage global tracking, where the first stage employs the C-C RPN network to perform cross correlation matching on the first-frame ground truth and the second stage utilizes the memory bank and the S-L RCNN matcher to incorporate previous high-confidence results for a comprehensive similarity prediction. In addition, an SCFD is adopted to implement independent UAV detection to correct possible errors. Finally, the optimal bounding box is output by multiple classification predictions and regression predictions.

needed to avoid such local suboptimal solution stemming from the rapid movement of small-scale drone targets. In addition, the issue of updating the Siamese fashion for long-term drone tracking also has to be considered, which is a trade-off problem between model adaption and degradation. Traditional methods employ a fixed learning rate to update the template so as to improve the model's adaptation to target deformations. Such updating mechanism will be more and more unreliable over time and eventually leads to model degradation. How to tackle these problems remain challenging and ill-solved.

In this article, we propose a robust searching region-free and template-free Siamese network (SiamSRT), to track UAV targets in TIR videos. In particular, we apply a searching region-free strategy that takes the whole image as input to achieve global optimal solution, and adopt a template-free strategy that involves all of the historical predictions to prevent model degradation. The proposed network architecture is shown in Fig. 1. Similar to the typical Siamese framework, SiamSRT contains two branches, i.e., a template branch, and a testing branch, which share weights. Among them, the template branch is initialized in the first frame and then remains fixed in subsequent frames. The testing branch inputs the searching image and constantly produces the newest candidate proposals by the cross-correlated region proposal network (C-C RPN), which consists of a cross correlation encoder and an RPN head. Following the first stage C-C RPN network comes a similarity-learning region convolutional neural network (S-L RCNN) matcher, which contains a temporal memory bank (TMB), a similarity-learning (SL) encoder and a RCNN head. Along with the S-L RCNN, there is a single-category foreground detector (SCFD), which enables the tracker independently to independently detect drones.

Specifically, SiamSRT employs a searching region-free strategy that makes the network compatible with arbitrary scale images as input, and then adopts the more powerful Swin transformer [25] as the feature extractor. The S-L RCNN matcher encodes the prediction results of the previous frames into the searching proposals to make optimal decisions in the face of the drastic target appearance deformation. In the matcher, a template-free strategy is utilized to address the limitation of Siamese template updates. In order to minimize the loss of target information in the process of model updating, SiamSRT concatenates high-confidence prediction results of the TMB into feature matrices, called sequential experts. Then, we cross-code the sequential experts with candidate proposals into the RCNN head, that is, each expert contributes to the prediction of the target in the current frame, so as to maximize the retention of target information and effectively solves the updating problem of Siamese templates.

Juxtaposed with the S-L RCNN matcher is the SCFD. One very fortunate attribute in our anti-drone tracking missions is that the targets being tracked are all drone foreground objects. The detector implements the UAV foreground prediction based on original RCNN head, which can not only output the classification score to identify whether the target is a drone or not but also output the regression score to locate the drone target. The detector estimates the target independently for each frame and is not plagued by the target appearance changes, which facilitates the recovery of tracking failures. It is worth noting that SFCD shares the feature extractor and RPN proposal generator with the S-L RCNN matcher, so the whole framework can be trained simultaneously in an end-to-end manner. To cope with similar UAV-like thermal regions in infrared images, we utilize the spatial location consistency

constraints to suppress these background thermal noises in candidate proposals.

In summary, there are three major novelties in this SiamSRT network. First, we build a two-stage searching region-free Siamese tracking framework, which performs coarse localization using C-C RPN network and then fine-tunes the position using S-L RCNN network. Second, we introduce a template-free memory bank into the S-L RCNN network that fully utilizes previous predictions to inspire the current tracking task, solving the model degradation puzzle. Third, we implement an SCFD into the decision-making stage, which delicately exploits the UAV foreground semantics to repair the tracking failures.

II. Related Works

Currently, the mainstream trackers are divided into two main categories: correlation filter-based and Siamese network-based ones. In the following, we first briefly review the development of these two tracking frameworks, and then introduce the most related works for two techniques tackled in this article, i.e. TIR tracking and anti-UAV tracking.

A. Correlation Filters

Bolme et al. [26] first introduce correlation filters into the tracking field by presenting a minimum output sum of squared error (MOSSE) filter that enables tracking with an extremely high speed of 669 frames/s. Inspired by MOSSE, a vast number of followers have improved the performance of correlation tracking from various aspects such as feature representations [27], [28], scale variations [29], boundary effects [30], [31], and kernel tricks [2], [32]. In these algorithms, SRDCF [31] employs a spatial weight function to penalize the filter coefficients to address the unwanted boundary effects, whereas BACF [30] achieves filter learning to real-world samples by multiplying a binary mask matrix. ASRCF [33] combines these two trackers and employs an adaptive weight regularization term to adjust the spatial weight function, achieving decent accuracy improvement. STRCF [34] introduces a temporal regularization term to avoid the filter degradation problem. Yuan et al. [35] propose an adaptive spatial–temporal context-aware (ASTCA) correlation tracker by combining STRCF and ASRCF and apply it to a UAV viewpoint tracking task. Huang et al. [15] introduce an adversarial learning generative network to generate instance-negative samples and encode them into the correlation filter objective function to further improve tracking performance.

B. Siamese Networks

In contrast to correlation filters, Siamese networks can enhance the feature representations of the target through offline training with massive image pairs. Bertinetto et al. [8] propose the first Siamese network tracker, called SiamFC, which utilizes a fully convolutional structure and a cross correlation layer to achieve SL between the template and searching images. SiamRPN [9] and SiamRPN++ [10] introduce the RPNs into Siamese tracking and localize targets through classification and regression prediction. GlobalTrack [20] and SiamRCNN [21] introduce the RCNN structure into the Siamese network, which guarantees the accuracy of the predicted bounding boxes through a two-stage fine-tuning. Yuan et al. [36] propose an effective self-supervised learning-based tracker, which fully utilizes the principle of forward–backward tracking consistency between consecutive frames to improve tracking accuracy under a Siamese correlation tracking framework. SiamATL [37] implements the online updating of the Siamese fashion via an attentional transfer learning strategy to cope with target template degradation under long-time occlusion. The ALT tracker [38] presents an active learning method for deep visual tracking, which can select and annotate the unlabeled samples to train a higher-quality Siamese model.

C. TIR Tracking

TIR tracking can compensates for the degradation issue of RGB tracking when encountering some challenging scenarios, such as foggy days and nights. Therefore, in order to release the power of the infrared tracking, Liu et al. [39] propose a TIR pedestrian tracking dataset for the TIR pedestrian tracker evaluation. What's more, Liu et al. [40] develop another large-scale TIR object tracking dataset, named LSOTB-TIR, to bridge the absence of infrared tracking training datasets. Besides the TIR datasets, the trackers specifically designed for infrared tracking has also received a lot of attention. MCFTS [41] proposes a correlation filter-based ensemble tracker, which utilizes multilayer convolutional features pretrained for the TIR targets. Li et al. [42] propose a hierarchical spatial-aware Siamese network for TIR tracking, which uses hierarchical convolutional features to acquire richer spatial and semantic feature representation for the TIR objects. Yuan et al. [43] present a spatial–temporal memory network to address occlusion and similar target interference in TIR tracking tasks. Liu et al. [44] propose a Siamese TIR tracking framework with a similarity computation structure with multiple levels, where one computes the global semantic similarity and the other computes the local structural similarity for the TIR objects. Zhang et al. [45] adopt image-to-image translation models to transfer the abundantly available labeled RGB images to synthetic TIR ones, which well solves the problem of insufficient infrared training data.

D. Anti-UAV Tracking

With the popularity of drone applications, the potential threat of drone intrusion has gradually increased [5]. Traditional anti-drone defense systems achieve control and defense against illegal intrusion of drones by using spectrum detection, radar detection, radio interference suppression, etc., [46]. These systems tend to be costly and inflexible, making it difficult to cope with high-frequency anti-UAV tracking requirements. Jiang et al. [5] present a large-scale anti-drone dataset that contains more than 300 video pairs with both infrared and visible light modalities. In addition, they have also organized several anti-drone competitions to help develop anti-drone defense systems [47]. Zhao et al. [46] propose a

visible light mode anti-drone dataset, called DUT anti-UAV, which consists of a detection dataset with a total of 10 000 images and a tracking dataset with 20 videos. Huang et al. [22] propose an effective spatio-temporal attention-based Siamese network for anti-drone missions, which achieves robust UAV tracking in TIR scenarios by posing spatial and temporal constraints on searching candidate proposals. Yu et al. [48] propose a unified transformer-based anti-UAV tracker, which designs a multiregion local tracking module to tackle target appearance variation and a global detection module to tackle frequent disappearance.

III. Proposed Method

Fig. 1 presents a detailed flowchart of the SiamSRT framework. The framework starts with Swin transformer feature extraction and then follows the two-stage detection. The first stage provides detections of the ground truth given in the first frame, the second stage implements detections using high-confidence predictions from previous frames and additionally adopts an SCFD to repair tracking failures. In this section, we elaborate in detail on the design of our SiamSRT architecture module by module.

A. Transformer Feature Extraction

Transformer has demonstrated high accuracy and robustness in visual tracking tasks [24], [49], [50], [51]. In Siamese feature extraction subnetwork, we adopt the Swin transformer architecture as the backbone.

1) Self-Attention: The basic block in a standard transformer architecture is the attention mechanism, which applies an attention function for mapping a query and a set of key-value pairs to an output. Denotes the query Q, keys K, and values V as the inputs, where the keys and values are packed together for uniform calculations. In computing self-attention, the similarity between the query and key is computed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}} + B\right)V \tag{1}$$

where d_k is the key dimensionality, and B is the relative position bias.

2) Swin Transformer Block: The single-head self-attention can be extended into multiple-head version, which allows the model to jointly learn various aspects of information from different representation subspaces at different positions. This expansion can be achieved by concatenation and linear projections

$$\text{MultiHead}(Q, K, V) = \text{Concat}(H_1, \ldots, H_h)W^O \tag{2}$$

where $H_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$. $W_i^Q \in R^{d_m \times d_k}$, $W_i^K \in R^{d_m \times d_k}$, $W_i^V \in R^{d_m \times d_v}$, and $W^O \in R^{hd_v \times d_m}$ are parameter matrices for the projections. Take it a step further, Swin transformer is built by replacing the multihead self-attention (MSA) module in a transformer block by a module based on shifted windows, with other layers kept the same. In this work, we keep the default parameters from Swin transformer [25] in the configuration of the feature extraction module.

B. Two-Stage Searching Region-Free Siamese Network

The two-stage Siamese network has presented excellent performance in target tracking [20], [21], [22]. The structure of SiamSRT consists of a cross-correlated RPN (C-C RPN) module and a similarity-learning RCNN (S-L RCNN) module. The C-C RPN module performs the first-stage of detection that encodes the target information into the search branch using the cross correlation layer and generates possible proposals for calculating the classification and regression losses. The S-L RCNN module implements the second stage of detection, which encodes the similarity between the of aligned template and each candidate proposal, and then inputs the proposal list into the RCNN head to compute the final predicted score.

1) Cross-Correlated RPN: Siamese tracking network interacts with the template branch and the search branch through a cross correlation operation [8], which is formulated as

$$\text{Corr}(\varphi(x), \varphi(z)) = \varphi(x) \otimes \varphi(z) + b\mathbf{1} \tag{3}$$

where $\otimes$ denotes cross correlation. $\varphi(z)$ and $\varphi(x)$ denote the template and searching feature maps, respectively. $b\mathbf{1}$ denotes the bias which takes value b in every location. The bias terms are often omitted for the brevity.

Our C-C RPN differs from traditional RPN in that it incorporates a cross correlation encoder before the RPN head. The encoder enables information interaction between the search branch and the template branch, and the encoding process is shown in Fig. 2(a). Denote $\varphi(z) \in R^{C \times K \times K}$ be the template features aligned by ROI layer and $\varphi(x) \in R^{C \times H \times W}$ be the features of search image, respectively, where $K \times K$ denotes the ROI-aligned output size, H and W denote the height and width of the global feature map, respectively, and C denotes the number of feature channels. We encode the target information into the feature map of the search branch as follows:

$$\begin{cases} \phi(z) = \varphi(z) * \text{Filter}(K, K) \\ \text{Corr}(\varphi(x), \varphi(z)) = (\varphi(x) \otimes \phi(z)) * \text{Filter}(1, 1) \end{cases} \tag{4}$$

where $*$ denotes the convolution operation. $\text{Filter}(K, K)$ is a $K \times K$ convolution filter, which convert $\varphi(z)$ to a 1×1 matrix. The target information hosted by $\phi(z)$ is then encoded into the feature map $\varphi(x)$ by a cross correlation layer. We then convert the output back to C channels by a 1×1 convolution layer, i.e., $\text{Corr}(\varphi(x), \varphi(z))$ and $\varphi(x)$ have the same data structure.

The difference between (3) and (4) is that we introduce two convolutional layers. Unlike traditional methods that crop out fixed-size template and search images as input, our SiamSRT takes arbitrary-size images as input to realize the global UAV detection. However for cross correlation operations, we need to scale the correlation kernel (or template) to a fixed size, so we adopt an ROI pooling layer to extract the features of the referenced UAV target. Since the search branch does not perform the same ROI pooling operation, it will lead to a misalignment problem between the template and search features. Therefore, a $K \times K$ convolution layer is introduced to encode the target information. To facilitate the subsequent computation, another 1×1 convolution layer is introduced to convert the number

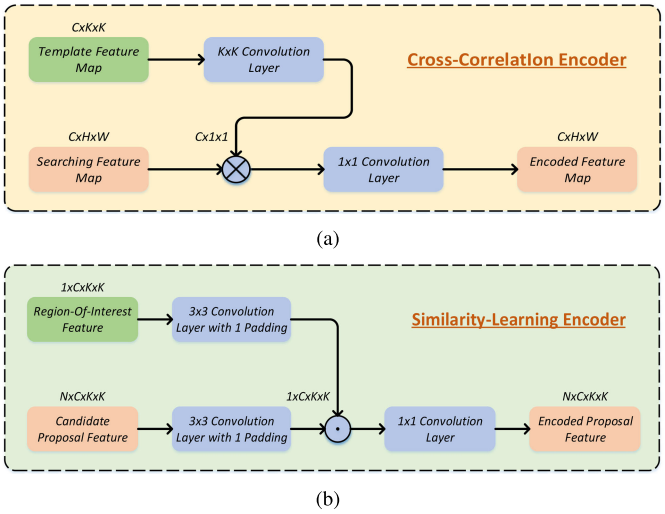


Fig. 2. Illustration of the cross-coding process. (a) Cross correlation encoder, where $\otimes$ represents the cross correlation operation, and the coder encodes the target information of the template branch into the feature matrix of the search branch. (b) SL encoder, in which $\odot$ means the Hadamard production, and the coder encodes the target information into each candidate proposal.

of channels back to the original feature channels. Finally, the cross-coded features $\mathrm{Corr}(\varphi(x), \varphi(z))$ are fed into the RPN head [52] for classification and regression.

2) Similarity-Learning RCNN: Further, we implement the two stage of the S-L RCNN module, which introduces an SL encoder into the traditional detection RCNN head. This encoder will encode the target information into each searching candidate proposal, and the similarity matching is then reached through the collaboration between the encoder and classification/regression. The encoding process is as illustrated in Fig. 2(b). Let $p(z) \in R^{1\times C\times K\times K}$ be the aligned ROI feature of the UAV target z and $p(x) \in R^{M\times C\times K\times K}$ the aligned ROI feature of candidate proposals extracted from the search branch x, where M is the number of candidate proposals. We encode their similarity by the following:

$$\begin{cases} q(z) = p(z) * \mathrm{Filter}(3,3) \\ q(x) = p(x) * \mathrm{Filter}(3,3) \\ \mathrm{Sim}(p(x), p(z)) = (q(x) \odot q(z)) * \mathrm{Filter}(1,1) \end{cases} \tag{5}$$

where $\odot$ indicates the Hadamard production. We first perform a 3×3 convolution with one pixel padding on $p(z)$ and $p(x)$. Since $q(z)$ and $q(x)$ also have the same size, we encode their correlations by the Hadamard production. As with the cross correlation encoder, the number of channels is converted back to c using a 1×1 convolution.

Based on the faster RCNN [53], the S-L RCNN of SiamSRT is computed as follows: first, the network goes through the C-C RPN network to obtain ROIs, and each ROI is pooled into a fixed-size feature map. The SL encoder is then used to encode the UAV template information into these aligned features. Furthermore, the encoded features $\mathrm{Sim}(p(x), p(z))$ are mapped to a feature vector by two shared fully connected layers (FCs). Finally, this feature vector undergoes two independent FCs to realize softmax classification prediction and bounding-box regression prediction, respectively. The calculation process of S-L RCNN is illustrated in Fig. 3(a).

The searching region-free Siamese network is effective in preventing locally optimal solutions caused by targets escaping from the search region. The same knife cuts bread and fingers, global search also has its side effects, for example, there are more thermal distractors in global search due to the introduction of massive background information. These thermal noises make it easy for the traditional trackers to locate the ROI onto similar background regions, and trackers are also plagued with the model degradation problem. To address the downsides associated with searching region-free tracking, our SiamSRT introduces a spatial location consistency function to suppress background thermal distractors and a TMB to avoid template update degradation problem. Next, we will introduce how these two techniques are implemented in the online tracking process.

C. Template-Free Online Tracking

In actual TIR tracking, UAV targets are typically very small, without prominent textures or fixed shapes, which make them extremely difficult to be detected. To mitigate this problem, we explore deeper into the spatio-temporal relationship in drone infrared videos. From the spatial perspective, targets are unlikely to show huge position changes in two consecutive frames. In a follow-up frame of the tracking results with high confidence, we consider it more valuable to detect targets by searching them within a local neighborhood than from a global context where a lot of thermal noise exists. From the temporal perspective, we argue that the target states learned from historical frames should be fully reused in the current tracking task to avoid the degradation problem of a single template.

1) Spatial Location Consistency: Define one high-quality track consists of A continuous non-overlapping sub-track as $L = (l_1, l_2, \ldots, l_A)$. For each sub-track $l_i, \forall i \in \{1, 2, \ldots, A-1\}$, there exists the relation $l_{i,e_i} < l_{i+1,s_{i+1}}$, where s_{i+1} and e_i indicate the start frame and end frame of the sub-track l_{i+1} and l_i, respectively. We calculate the evaluation score of the candidate proposals belonging to track L by the following:

$$\begin{aligned} &\mathrm{Eva}(p(x)) \\ &= w_r \mathrm{sim\_eva}(p(x), gt) \\ &\quad + (1-w_r)\mathrm{sim\_eva}(p(x), l_{A,s_A}) + w_l \mathrm{loc\_eva}(p(x), l_{A,e_A}) \end{aligned} \tag{6}$$

where gt stands for the ground truth given in the first frame and $p(x) = [px_1, px_2, \ldots, px_M]$ denotes the search proposal list of branch x in the testing frame. w_r and w_l are the complementary ratios. sim_eva denotes similarity evaluation, where it feeds the encoded features of $\mathrm{Sim}(p(x), gt)$ and $\mathrm{Sim}(p(x), l_{A,s_A})$ into the RCNN head and returns the detection confidence scores. loc_eva is the location consistency evaluation, where the intersection over union (IoU) is used to impose neighborhood restrictions.

We consider that the target cannot undergo a sudden and dramatic location change in two consecutive frames. When the target is occluded, the neighborhood search tends to be trapped in a local ROI region, and if the target reappears outside this region, the tracker fails completely. Therefore,

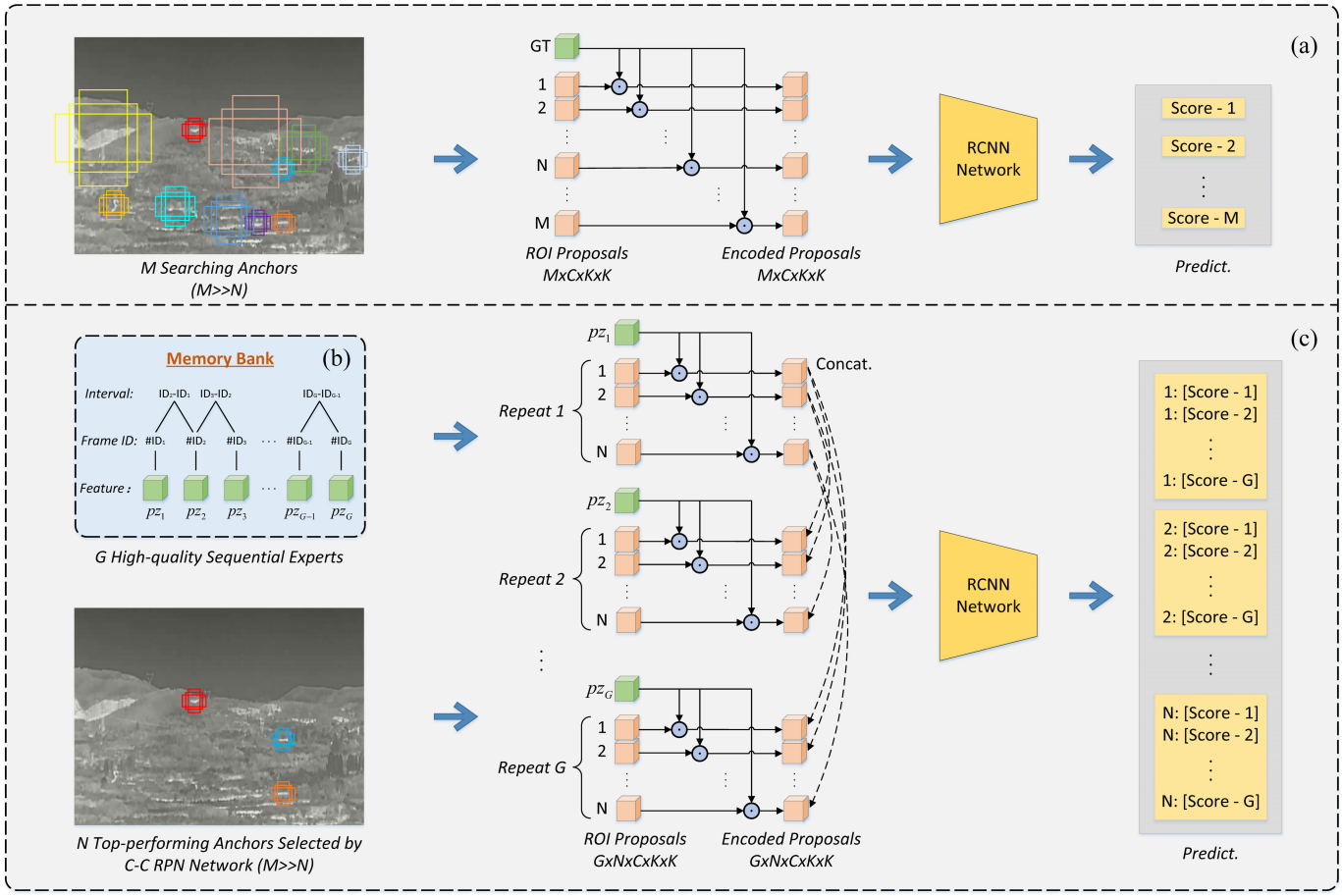


Fig. 3. Comparison of SL for single template and memory bank. (a) Similarity calculation process by S-L RCNN for one target template. (b) Memory bank for storing historical prediction results. (c) Similarity calculation by S-L RCNN for a large number of prediction templates in the memory library.

global search is more effective against target occlusion, but introduces more background noises, while local tracking limits background noise to a greater extent, but it is difficult to capture the target again after the target loss. Our decision strategy wants to combine both of them, which conditionally switches between local tracking and global search to make the optimal decision. In the next frame of the high-confidence prediction result, we first perform a neighborhood search, $\text{loc\_eva}(l_{A,e_A}, p(x)) > 0.2$, to find the promising predictions. When finding a proposal px_i with sim_eva greater than the given threshold, the neighborhood search is regarded as successful and px_i will be added to the sub-track l_A. Otherwise, it means that no suitable target is found in the neighborhood, the sub-track l_A will be terminated. When the neighborhood search fails, we set the output of the spatial consistency evaluation loc_eva to zero, and $\text{Eva}(p(x))$ can still output the computing scores from the global proposals, and we output the global highest scoring proposal as the prediction result. Thanks to the spatial location constraints, extensive proposals with similar objectives are eliminated for decision making, which greatly alleviates the interference of distractors.

2) Template-Free-Based S-L RCNN: To unleash the power of temporally unveiled information, we introduce TMB to fully utilize both the first-frame template and previous-frame predictions for the optimal decision. To this end, we record the aligned features of the target in the historical high-quality predictions, denoted as $p(z) = [pz_1, pz_2, \ldots, pz_G] \in R^{G\times C\times K\times K}$, and the IDs of their corresponding frame numbers, $[\#ID_1, \#ID_2, \ldots, \#ID_G]$, where G indicates the capacity of the memory bank. We also call these estimated targets as sequential decision experts. As shown in Fig. 3(b), to cover as many possible states of the target as possible, the high-quality predictions will be added directly to the memory bank. To prevent an unlimited increase in the number of templates, we make the maximum capacity of the memory bank a manually adjusted parameter. If the capacity of memory bank is fully occupied and a new high-quality prediction comes, the memory bank will be updated by removing the larger of the sequential experts with the minimum frame ID interval. A simple illustration of the update process for a memory bank with a capacity of 5 is shown in Fig. 4. In all experiments, we set the maximum capacity of the memory bank to 50, which is sufficient to cover the possible patterns of the target in a long sequence.

When we repeatedly solve the similarity evaluation of each sequential expert pz_i with respect to the search $p(x)$, then we need G this repeated computation, which is a huge waste of computational resources. For fast computation, we select several top-performing N proposals $p(x) \in R^{N\times C\times K\times K}$ with high confidence scores output by the first-stage C-C RPN detection. Since the SL encoder is a mirroring process, we can treat $p(z)$ as a proposal list as well and then encode the search

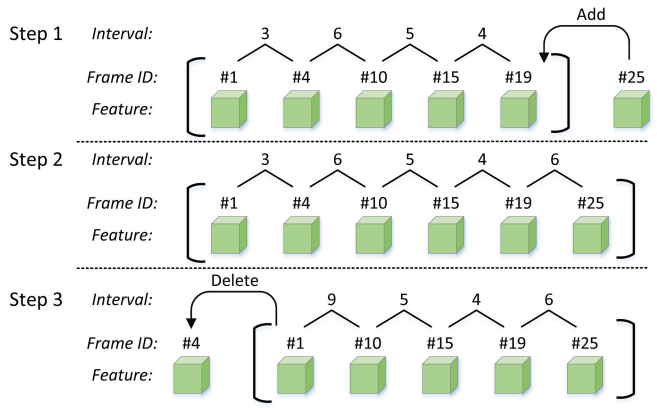


Fig. 4. Illustration of the update process of the memory bank. The capacity size of the memory bank is 5, the frame IDs of the currently stored features are {#1, #4, #10, #15, #19}, and their corresponding intervals are {3, 6, 5, 4}. When the result of the 25th frame is a high quality prediction, we add it to the template library and recalculate the frame interval to {3, 6, 5, 4, 6}. Finally, we remove the larger-frame feature with the minimum frame interval and update the frame interval to {9, 5, 4, 6}.

proposals $p(x)$ into $p(z)$. By SL and concatenating, we obtain an encoded feature matrix $\mathrm{Sim}(p(z), p(x)) \in R^{G\times N\times C\times K\times K}$. Thus, we can calculate the evaluation score for each candidate proposal px_i as

$$\mathrm{Eva}(px_i) = \frac{1}{G}\sum_{j=1}^{G}\alpha_j \mathrm{sim\_eva}(pz_j, px_i) \tag{7}$$

where pz_j denotes the j-th decision-making expert, while α_j denotes a temporal weighting factor. Equation (7) indicates that numerous previous predictions are involved in scoring candidate proposals in the current tracking task, which tackles the template update issue. As illustrated in Fig. 3(c), all these scores are generated directly through one sharing RCNN network, which does not introduce additional computation cost. Considering spatial consistency, the final prediction scores are computed as follows:

$$\begin{aligned}&\mathrm{Eva}(px_i)\\ &= w_r \mathrm{sim\_eva}(gt, px_i) + w_l \mathrm{loc\_eva}(l_{A,e_A}, px_i)\\ &\quad +(1-w_r)\frac{1}{A}\sum_{j=1}^{A}\frac{1}{e_j - s_j + 1}\sum_{t=s_j}^{e_j}\mathrm{sim\_eva}(l_{j,t}, px_i)\end{aligned} \tag{8}$$

where the first term re-detects the first-frame template, the second term imposes the spatial constraints, and the third term performs the detection of historical high-quality prediction results. The confidence scores for all three items range from 0 to 1. It is worth noting that since we consider extensive historical high-quality predictions in L, an error in one sub-track l_i will not cause the model to fail completely.

D. Single-Category Foreground Detector

Although we utilize extensive sequential decision experts to avoid model degradation, heavy occlusion may also pollute the template library causing complete tracking failures. Therefore, when a high-confidence sub-track l_i is broken, we require a global repair strategy to drive the tracker back to the correct trajectory. Therefore, we introduce an independent detection module, an SCFD, into SiamSRT to mitigate the effect of target loss.

First, we assign a binary class label (of being the foregrounds or backgrounds) to each search proposals. The proposal boxes with IoU overlap higher than 0.7 with the ground-truth box are given a positive label; while other background proposals are given a negative label. We construct a two-stage SCFD which takes these labeled proposals as the training set and then predicts the probability of each proposal being a foreground target. In this work, the foregrounds are UAV targets, so training our SCFD can also be considered as training a drone detector. The SCFD shares the same feature extraction and RPN network with the Siamese tracking framework. As a result, SCFD does not introduce much additional computation. Then the proposal list $p(x)$ generated by RPN network is fed into the RCNN head for binary classification, and it will return the probability that the proposal is a drone.

It is worth noting that since the SiamSRT is designed for anti-drone tracking, it does not excel at general target tracking or multicategory tracking missions. For example, the global detection of searching region-free strategy is designed to cope with tiny UAV targets, which may not be useful for tracking large scale targets. In addition, SiamSRT adopts the SCFD to modify tracking failures, which does not necessarily work when tracking semantically diverse generalized targets. SCFD is a binary- or single-category detector, which only gives the foreground target object and the background pixels. In the task of anti-drone tracking, there is only one tracking category, i.e., the drone target is the foreground, whereby the SCFD can be treated as a drone detector. In generalized tracking tasks, targets being tracked vary widely, which makes it particularly difficult for this single-category detector to learn unified semantics, hence SiamSRT struggles a lot when confronted with the generic category of target tracking.

E. Loss Function

In order to learn stronger similarity semantic features, SiamSRT does not include images of the previous frame in the training process. For training the first stage C-C RPN network, similar to [20], we adopt the binary cross-entropy (BCE) and smooth $L1$ as the classification and regression losses to train our model

$$\mathcal{L}_{\mathrm{rpn}} = \frac{1}{N_c}\sum_i \mathcal{L}_{\mathrm{bce}}\left(p_i, p_i^*\right) + \lambda\frac{1}{N_r}\sum_i p_i^*\mathcal{L}_{\mathrm{smooth}\ L1}\left(u_i, u_i^*\right) \tag{9}$$

where p_i denotes the predicted classification values and u_i denotes the predicted bounding-box regression offsets. $p_i^* \in \{0, 1\}$ and u_i^* are the ground-truth label and target box, respectively. $(1/N_c)$ and $(1/N_r)$ are the normalization parameters for classification and regression and λ is the balancing parameter for these two terms.

With the multitask mission in the second phase of RCNN-based decision-making module, we minimize the following objective loss function in SiamSRT:

$$\mathcal{L}_{\mathrm{rcnn}} = \mathcal{L}_{\mathrm{sl}}(B, A) + \mathcal{L}_{\mathrm{scfd}}(B) \tag{10}$$

where the first term represents the SL loss between search branch B and template branch A. The second term indicates the single-category foreground detection (SCFD) loss for UAV object. They adopt the same loss function structure, containing a BCE classification loss and a smooth $L1$ regression loss.

IV. Experiments

In this section, we first illustrate the implementation details and evaluation metrics. Next, we conduct quantitative and qualitative evaluations to demonstrate the effectiveness of our method. Then, we describe the ablation studies of our method. Finally, we verify the generalization ability of our method on RGB datasets, LaSOT [54] and GOT10k [55], and TIR datasets, PTB-TIR [39] and LSOTB-TIR [40].

A. Implementation Details

The proposed approach is implemented on an Ubuntu 16.04 × 64 system with an Intel[1] Xeon[1] CPU (E5-2620 v4 2.10 GHz), an NVIDIA GPU (GTX1080TI), and a 32 GB DDR4 RAM. In the Swin transformer [25] backbone architecture, we set the shifting window size, query dimension, and channel number in the hidden layer to 7, 32, and 96, respectively. The selected layer numbers are {2, 2, 6, 2}. To make the training converge faster, SiamSRT initializes the backbone weights using the Swin transformer for object detection model [25], which is pretrained using the COCO dataset [56]. While the rest parts of SiamSRT are trained from scratch. A total of 12 epochs are trained using the stochastic gradient descent (SGD) [53] optimizer with the learning rate set to 0.01. For the main experiments, we train our model using the training and test-dev splits of the anti-UAV [5] dataset, and then evaluate the model on the validation set, test set, and test-challenge set of the anti-UAV dataset. For the generalization experiments on RGB datasets, we train SiamSRT on the LaSOT [54] and GOT-10k [55] training sets, respectively, and use the corresponding models for evaluation on the corresponding datasets. As for the generalization experiments on the TIR datasets, we train SiamSRT using the LSOTB-TIR training dataset and test on the PTB-TIR and LSOTB-TIR testing datasets.

B. Evaluation Metrics

We use precision plot (PP) and success plot (SP) to evaluate the performance of the tracker through one-pass evaluation (OPE) [57]. PP represents the percentage of frames whose estimated locations are within a given threshold distance from the center of the ground truth. SP measures the percentage of frames for which the IoU ratios of the predicted and ground-truth bounding boxes are greater than a given threshold. In addition, we also employ the state accuracy (SA) metric for performance analysis. The SA metric, defined in anti-UAV [5], additionally introduces a ground-truth visibility flag to calculate the average overlap ratio between the predicted and ground-truth bounding boxes for all sequences. In our experiments, an error threshold of 20 pixels is used to evaluate tracking performance in the PP, and the area under the curve (AUC) for all the thresholds is adopted to rank the trackers in the SP.

[1]Registered trademark.

C. Quantitative Evaluations

In this subsection, we perform quantitative experiments on anti-UAV [5] dataset, which contains 160 videos as the training set, 67 videos as the validation set, and 91 videos as the test set. In order to make a comprehensive evaluation, we compare our SiamSRT method with 36 state-of-the-art trackers, including 11 correlation filter-based trackers, DSST [29], KCF [32], SRDCF [31], Staple [27], CSR-DCF [58], ECO [28], BACF [30], STRCF [34], ASRCF [33], ARCF [59], AutoTrack [60], and 25 deep learning based trackers, MDNet [61], SiamFC [8], CFNet [14], SiamRPN [9], DaSiamRPN [62], TADT [63], SiamRPN++ [10], ATOM [64], DiMP [65], SiamFC++ [66], GlobalTrack [20], PrDiMP [67], SiamCAR [68], SiamBAN [69], Siam RCNN [21], ROAM [70], Super_DiMP [71], KYS [72], TrDiMP [24], TransT [51], STMTrack [50], HiFT [49], Stark [23], KeepTrack [73], and SiamSTA [22]. The results of all compared trackers are produced on our platform with the default parameter settings.

Fig. 5 shows the tracking performance of these mentioned trackers on the anti-UAV test set in terms of the PP [Fig. 5(a)], SP [Fig. 5(b)], and SA [Fig. 5(c)] metric. Since these three metrics follow similar variation trend, with Staple algorithm being the worst and our algorithm the best, we choose the numerical results of the SA metric for the subsequent analysis: 1) ASRCF comes out on top among correlation filter-based algorithms with an average accuracy of 0.431, which is more than 28% worse than our SiamSRT algorithm; 2) SiamRPN++ (0.426) performs better than SiamRPN (0.416) indicating that a deeper network model can slightly enhance tracking performance for UAV tracking tasks; 3) the algorithms with transformer structures, such as Stark (0.591), TrDiMP (0.547), and TransT (0.521), perform well, which means that this new feature is promising for anti-drone tracking; 4) the two-stage trackers, such as GlobalTrack (0.643) and Siam RCNN (0.652), outperform SiamRPN series and DiMP series algorithms owing to the fact that global detection offers advantages over local search when handling TIR similar target interference; and 5) SiamSRT obtains a superb score of 0.716 for using spatio-temporal constraints to ensure robust tracking and a memory bank to prevent the model from degenerating.

D. Qualitative Evaluations

To better demonstrate the effectiveness and robustness of the proposed tracker, in this section, we perform qualitative experiments to analyze the performance against the most common challenging factors in TIR tracking, i.e., LC, tiny scale (TS), occlusion, fast motion (FM), scale variation, and background clutter (BC). For simplicity and clearer presentation, we compare our tracker with six representative trackers, including the correlation filter-based tracker ASRCF [33], the one-stage tracker SiamRPN++ [10], the two-stage tracker Siam RCNN [21], the transformer-based trackers TrDiMP [24] and Stark [23], and the base tracker SiamSTA [22].

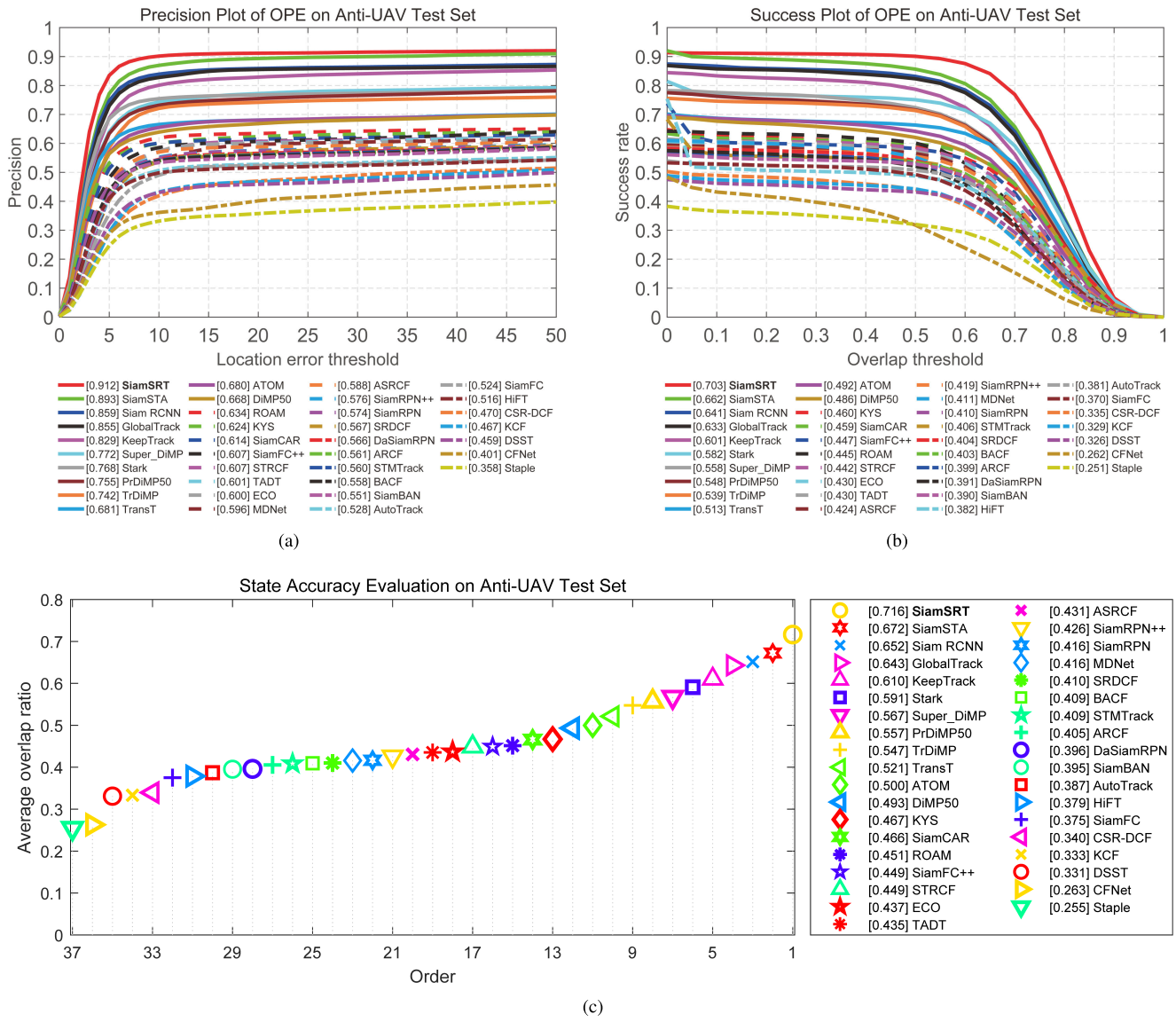


Fig. 5. Results of SiamSRT and the compared algorithms on anti-UAV test set (a) PP, (b) SP, and (c) SA ranking. The numbers in the legend indicate the average precision scores for PP, the AUC scores for SP, and average overlap ratios for SA. SiamSRT obtains best scores of 0.912, 0.703, and 0.716, respectively.

1) Low Contrast: Since there is only 1-D grayscale information in the thermal image, the LC makes it difficult to discern the target from the background with the same heat level. The targets undergo the LC challenge as shown at the top of Fig. 6. In this situation, the UAV target and the building background have the same heat level, so the target is completely submerged in the background. All the trackers except our SiamSRT lose the target, which indicates that our algorithm has stronger discriminative power due to the advanced feature extractor and decision maker.

2) Tiny Scale: Drones are usually very small in size and are basically shot from a long distance, so TS situations are common in UAV tracking missions. Video sequences in the "3700000000002_153934_1" scene are used to test the performance of these trackers to handle TS targets. TS makes it difficult to extract effective semantic features, so the powerful transformer feature-based algorithms Stark and TrDiMP perform poorly, while SiamSRT shows clear superiority over other trackers in handing this tracking issue due to the use of spatio-temporal constraints.

3) Occlusion: The third row of Fig. 6 shows the video sequences in which the targets suffer partial or short-term complete occlusions. Occlusion pollutes the target appearance model with updates and leads to irreversible errors. ASRCF, SiamRPN++, and TrDiMP trackers fail to find the lost target in all three sequences. On the one hand, they lack an effective model update strategy, and on the other hand, they need global detection instead of local search to cope with long-term occlusion. SiamSRT uses memory bank to prevent model degradation, and a combination of neighborhood search and global detection to capture the target again, which results in the superior performance of our algorithm against occlusion.

4) Fast Motion: The movement of the drone or the change of the camera viewpoint will cause FM challenges. FM blurs

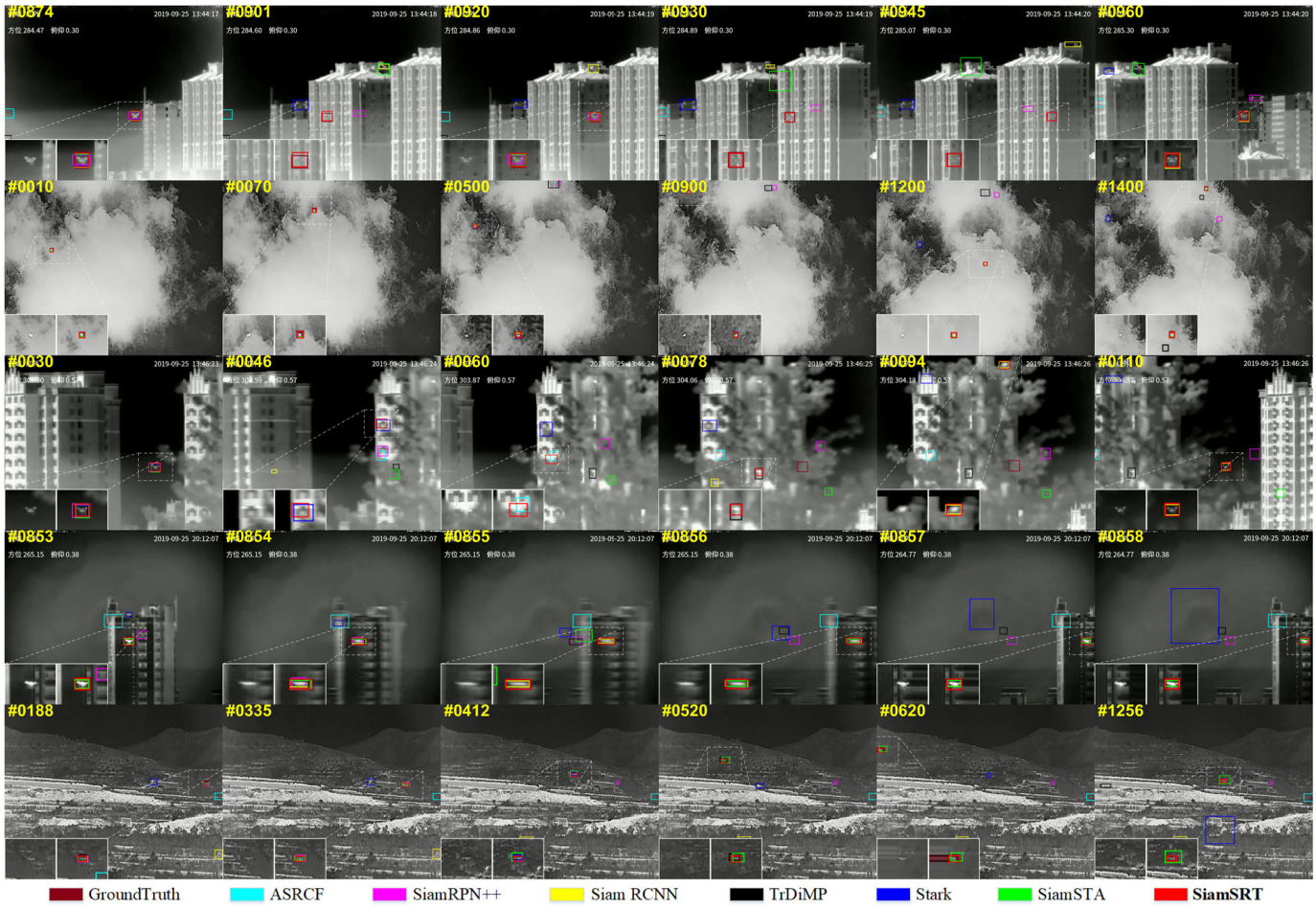


Fig. 6. Visual comparison with six state-of-the-art trackers. Representative frames are shown for five video sequences: "20190925_134301_1_2," "3700000000002_153934_1," "20190925_134301_1_6," "20190925_200805_1_6," and "3700000000002_140908_1" (from top to bottom). The main challenges illustrated in these figures are LC, TS, occlusion (OCC), FM, and BC, respectively.

or even deforms the target, and the tracker requires a wider search range to ensure that it can capture the target again. On the "20190925_200805_1_6" sequence, the rotating camera causes a sharp violent of the target at frame #855, thus resulting in an FM challenge. Only Siam RCNN and SiamSRT can adapt to this change, which illustrates the robustness of our algorithm.

5) Background Clutter: Some wild scenarios are very cluttered with a large number of similar targets. The bottom of Fig. 6 presents the tracking performance in the face of the BC challenge, where the predicted bounding box may drift to the background as it becomes very difficult to recognize the target object from the background by a rather simple model. Our algorithm performs better in this case because the introduction of an additional drone detector in the decision-making phase makes the prediction more accurate.

E. Comparisons With Deep Trackers Re-Trained Using Anti-UAV Dataset

To further demonstrate the superiority of the SiamSRT algorithm, we re-trained several representative deep trackers using the anti-UAV [5] training set for comparison, i.e., ATOM [64], DiMP50 [65], PrDiMP50 [67], SiamBAN [69], SiamCAR [68], KYS [72], KeepTrack [73], and Stark [23]. We keep all default settings exactly when re-training these compared trackers, and the experimental results are shown in Table I. Specifically, the numbers in the brackets indicate the tracking results of the original model without re-training with the anti-UAV data. We can figure out that the comparative trackers achieve more or less improvement by fine-tuning with

TABLE I

COMPARISONS WITH DEEP TRACKERS RE-TRAINED USING ANTI-UAV DATASET. THE NUMBERS SHOWS THE SA (%) SCORES ON ANTI-UAV TEST AND VAL SET

Method	Source	Test w/ (w/o) re-training	Val w/ (w/o) re-training
ATOM [64]	CVPR19	55.17 (49.98)	66.22 (59.82)
DiMP50 [65]	ICCV19	59.29 (49.33)	71.31 (62.48)
PrDiMP50 [67]	CVPR20	57.79 (55.70)	71.90 (62.61)
SiamBAN [69]	CVPR20	51.45 (39.53)	53.40 (42.42)
SiamCAR [68]	CVPR20	50.26 (46.59)	54.85 (54.79)
KYS [72]	ECCV20	46.82 (46.70)	60.70 (60.35)
KeepTrack [73]	ICCV21	61.41 (61.04)	68.86 (67.95)
Stark [23]	ICCV21	62.51 (59.08)	75.22 (69.03)
SiamSRT	**Ours**	**71.64**	**80.04**

TABLE II

COMPARISON OF PERFORMANCE OF DIFFERENT METHODS ON LASOT AND GOT-10K DATASET. RED FONTS INDICATE THE BEST PERFORMANCE, THE BLUE FONTS INDICATE THE SECOND BEST ONES, AND THE GREEN FONTS INDICATE THIRD ONES

Method	Source	LaSOT			GOT-10k		
		AUC	P	P_{Norm}	AO	SR_{50}	SR_{75}
ASRCF [33]	CVPR19	0.344	0.331	0.391	0.312	0.314	0.113
ATOM [64]	CVPR19	0.514	0.505	0.577	0.550	0.626	0.396
SiamRPN++ [10]	CVPR19	0.496	0.491	0.569	-	-	-
DiMP50 [65]	ICCV19	**0.571**	**0.569**	**0.652**	0.611	0.717	0.492
SiamFC++ [66]	AAAI20	0.500	0.474	0.571	0.526	0.625	0.347
GlobalTrack [20]	AAAI20	0.521	0.529	0.599	-	-	-
ROAM [70]	CVPR20	0.390	0.368	0.441	-	-	-
SiamCAR [68]	CVPR20	0.516	0.524	0.610	0.579	0.677	0.437
SiamBAN [69]	CVPR20	0.514	0.521	0.598	-	-	-
Ocean [74]	ECCV20	0.516	0.526	0.607	**0.615**	**0.735**	0.485
SiamGAT [11]	CVPR21	0.539	0.530	0.633	**0.627**	**0.743**	**0.488**
TrDiMP [24]	CVPR21	**0.640**	**0.666**	**0.732**	**0.671**	**0.777**	**0.583**
SiamSRT	**Ours**	**0.556**	**0.573**	**0.648**	0.590	0.678	**0.523**

the anti-UAV training set, where SiamBAN acquires the most substantial boost, yielding more than 10% performance gain on both the test dataset and the Val dataset. Nevertheless, the proposed SiamSRT significantly outperforms these algorithms, surpassing the second-place algorithm, Stark, by more than 9.1% on the test set and 4.8% on the Val set.

F. Generalization Experiments on RGB Datasets

Although our trackers are designed to counter the drone threat, it also has immense potentials in generic target tracking missions. We test the generalization ability of SiamSRT on two popular large-scale RGB datasets, i.e., LaSOT [54] and GOT-10k [55]. According to the protocol evaluation criteria, we use AUC, precision (P), and normalized precision (P_{Norm}) to rank the performance of trackers on LaSOT dataset. On GOT-10k dataset, we employ the default evaluation metrics AO, SR_{50}, SR_{75}, and submit the tracking results to the official evaluation server for evaluation. We compare SiamSRT with 12 competitive methods, ASRCF [33], ATOM [64], SiamRPN++ [10], DiMP50 [65], SiamFC++ [66], GlobalTrack [20], ROAM [70], SiamCAR [68], SiamBAN [69], Ocean [74], SiamGAT [11], and TrDiMP [24].

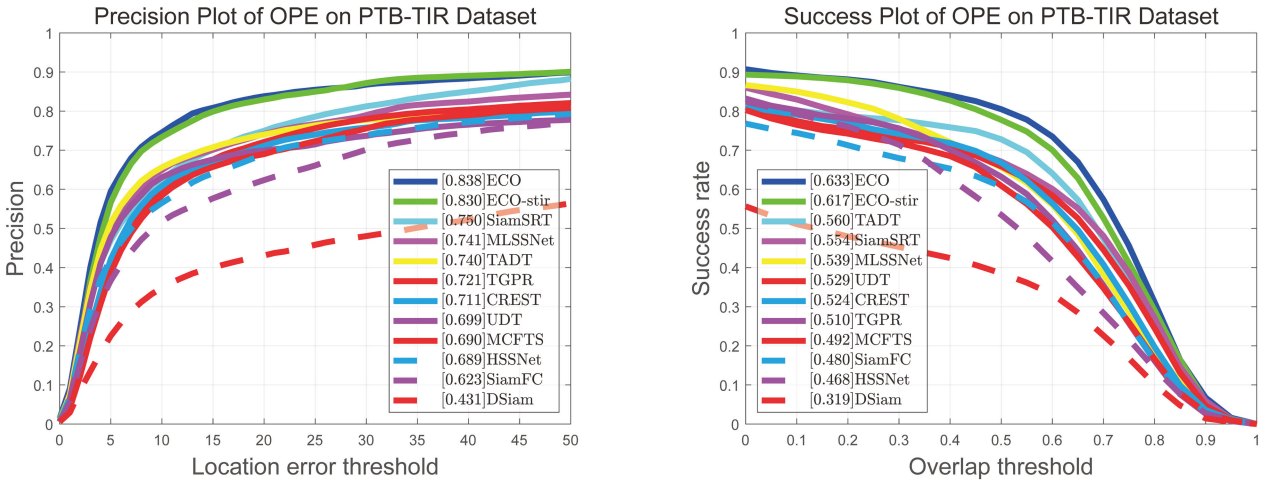
Table II reports comparison scores, and our algorithm achieves the best performance in four out of six metrics. When tracking a generic object, the SCFD in the second stage can be treated as a fine-tuned decision network. Since we have encoded the first-frame template into the search branch in the first stage, the SCFD also outputs the prediction results of similarity matching. Therefore, our algorithm also demonstrates excellent performance for generic target tracking. It is worth noting that SiamSRT vastly outperforms the DiMP and TrDiMP tracker on the anti-UAV dataset, while SiamSRT falls slightly short of these two trackers on the LaSOT and GOT-10k datasets. The reasons are as follows: 1) DiMP and TrDiMP adopt local search centered on the target, which is prone to lose small-scale UAV targets, while SiamSRT provides the ability to globally capture UAV targets by using the entire image as network input; 2) the SCFD employed by SiamSRT will not be significantly useful for generalized target tracking tasks on the LaSOT dataset and the GOT-10k dataset; and 3) DiMP and TrDiMP do not adopt suitable spatio-temporal constraints to limit the background noise in TIR videos, while the thermal noise suppression strategy utilized by SiamSRT is not necessarily effective in visible light datasets.

G. Generalization Experiments on TIR Datasets

In this section, we further conduct generalization experiments on two TIR datasets, i.e., PTB-TIR [39] and LSOTB-TIR [40]. PTB-TIR is a pedestrian tracking dataset containing 60 infrared video sequences for testing, and no training set is provided. LSOTB-TIR is a large-scale generic tracking dataset, which consists of a training subset and a test subset with a total of 1400 TIR sequences (120 sequences for testing) and more than 600 K frames. In this generalization experiment, we utilize the LSOTB-TIR training set to re-train our SiamSRT tracker and then evaluate it on the PTB-TIR and LSOTB-TIR datasets. The compared algorithms include TIR trackers ECO-tir [45], MLSSNet [44], HSSNet [42], MCFTS [41], RGB trackers ECO [28], TADT [63], TGPR [75], CREST [18], UDT [76], SiamFC [8], and DSiam [77]. The experimental results on these two datasets are shown in Figs. 7 and 8, respectively. On the PTB-TIR dataset, SiamSRT obtains scores of 0.750 and 0.554 in the precision and SPs, respectively, which falls short of the ECO-stir algorithm and far exceeds the typical SiamFC algorithm. On the LSOTB-TIR dataset, SiamSRT obtains a precision score of 0.664 and an AUC score of 0.559, achieving fourth place. There are two possible reasons why SiamSRT does not perform outstandingly: one is that the strategy of no search region does not have a significant effect on large-scale targets in the TIR dataset; the other is that the foreground detector does not work well for multiple categories of targets either. Nevertheless, the experimental results demonstrate the potential of SiamSRT for applications in other infrared target tracking tasks in addition to anti-UAV tracking.

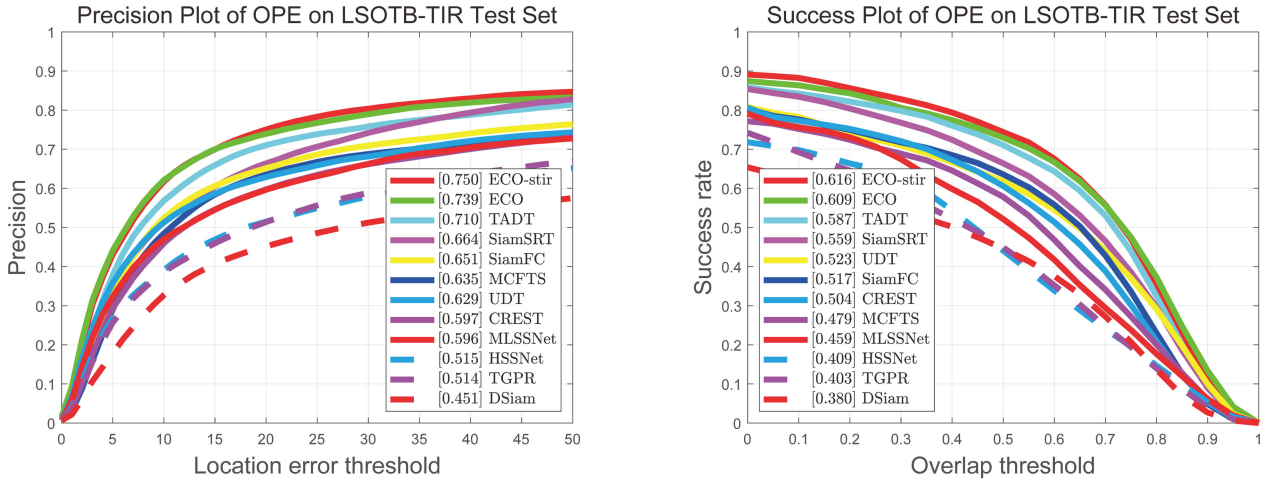
H. Comparison With Different Backbone Architectures

The choice of feature extractor is crucial, as it directly determines the number of parameters and the type of layers, which consequently affects the memory, speed, and performance of the tracker. We compare the tracking effect of



(a) (b)

Fig. 7. Results of SiamSRT and the compared algorithms on PTB-TIR dataset (a) PP and (b) SP. The numbers in the legend indicate the average precision scores for PP and the AUC scores for SP. SiamSRT obtains scores of 0.750 and 0.554, respectively.



(a) (b)

Fig. 8. Results of SiamSRT and the compared algorithms on LSOTB-TIR test set (a) PP and (b) SP. The numbers in the legend indicate the average precision scores for PP and the AUC scores for SP. SiamSRT obtains scores of 0.664 and 0.559, respectively.

using different network architectures as the backbone part, including DarkNet53 [78], ResNet50 [79], ResNet101 [79], ResNeXt50 [80], ResNeXt101 [80], and Swin-T [25]. Table III reports the performance by SA metric on the anti-UAV dataset. We observe that 1) residual series networks obviously outperform dark networks; 2) our SiamSRT cannot benefit from being equipped with a deeper ResNet101/ResNeXt101 network; and 3) ResNet, ResNeXt, and Swin-T networks have similar tracking scores. We finally select Swin-T as our backbone.

TABLE III
COMPARISON WITH DIFFERENT BACKBONE ARCHITECTURES. THE SA (%) SCORES ON TEST SET AND VAL SET OF ANTI-UAV DATASET ARE REPORTED

Backbones	Test	Val	Backbones	Test	Val
DarkNet53	63.13	77.15	ResNeXt50	69.92	79.95
ResNet50	70.00	80.03	ResNeXt101	70.79	79.98
ResNet101	69.94	80.04	Swin-T	71.64	80.04

I. Componentwise Analysis of Different Modules in the SiamSRT

To verify the effectiveness of different components of our SiamSRT, we perform componentwise experiments on the anti-UAV dataset to explore the effect of each incremental module, i.e., C-C RPN, S-L RCNN, TMB, and SCFD. The SA results for the test set and Val set are presented in Table IV, and subsequent statements are described with numerical results from the test set. As shown in the top row, the one-stage tracker with only C-C RPN obtains a score of 55.62, exceeding the traditional one-stage tracker SiamRPN by 14%, which indicates that our cross correlation coding strategy is better than that employed by SiamRPN. Method #2 is the simplest two-stage tracker with C-C RPN as the first-stage prediction

TABLE IV
EFFECTIVENESS OF IMPORTANT COMPONENTS IN THE SIAMSRT, I.E., C-C RPN, S-L RCNN, TMB, AND SCFD. THE EVALUATION CRITERIA IS THE SA (%)

#	C-C RPN	S-L RCNN	TMB	SCFD	Test	Val
1	✓				55.62	64.26
2	✓	✓			65.93	76.12
3	✓	✓	✓		67.74	77.73
4	✓			✓	65.61	75.16
5	✓	✓	✓	✓	71.64	80.04

and S-L RCNN as the second-stage prediction, which only takes the target in the first frame as template and never updates. Method #2 greatly exceeds Method #1 by more than 10%, which also confirms the effectiveness and necessity of second-stage detection. Method #3 further enhances the performance by employing the TMB, in which the high-confidence previous prediction results are stored, thus solving the problem of balancing between model update and degradation. Method #4 does not perform cross-coding of the two branches in the second stage, i.e., a separate RCNN is used in the second stage for independent UAV foreground detection. It is very interesting to note that the output predicted by Method #4 has both the semantics of cross correlation in the first stage and the semantics of the UAV foreground detection in the second stage. Lastly, Method #5 represents our final approach.

V. CONCLUSION AND DISCUSSION

We have presented an effective SiamSRT tracking method with outstanding performance in long-term anti-drone tracking. We build a two-stage network to detect the target using the first-frame template and previous predictions. The spatio-temporal knowledge is fully utilized in our network through a spatial location consistency constraint and a temporal template memory bank. The extensive experiments on the anti-UAV, LaSOT, GOT-10k, PTB-TIR, and LSOTB-TIR datasets demonstrate the effectiveness and robustness of our method, compared with the state-of-the-art trackers.

Traditional trackers tend to overlook the fact that the targets tracked are often foreground objects. In addition to the template information in the first frame, the probability estimation of the foreground object can also assist the tracker in making judgments or correcting the tracker's wrong predictions, which is especially applicable in this anti-drone tracking task. So, we introduce a single-category foreground UAV detector to reduce the potential errors. Furthermore, trackers of the Siamese fashion are always plagued by the online updating issue, which is a trade-off between model adaption and degradation. SiamSRT constructs a new type of two-stage detection network to solve this problem. We do not update the template in the first stage of C-C RPN detection, which means that the target information of the first frame will be preserved permanently and the tracker will not degrade easily. For the second-stage detection, we extremely incorporate a large number of high-quality tracking results into the decision of similarity evaluation, which ensures that our tracker does not miss any of the possible states of the target to accommodate the target appearance changes. Therefore, our tracker has triple guarantees: 1) matching with the target in the first frame; 2) SL with prediction results from previous frames; and 3) UAV-like appearance, these give rise to a highly accurate decision-making outcome.

REFERENCES

[1] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 445–461.

[2] B. Huang, T. Xu, S. Jiang, Y. Chen, and Y. Bai, "Robust visual tracking via constrained multi-kernel correlation filters," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2820–2832, Nov. 2020.

[3] N. T. Jafferis, E. F. Helbling, M. Karpelson, and R. J. Wood, "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle," *Nature*, vol. 570, no. 7762, pp. 491–495, Jun. 2019.

[4] O. M. Cliff, D. L. Saunders, and R. Fitch, "Robotic ecology: Tracking small dynamic animals with an autonomous aerial vehicle," *Sci. Robot.*, vol. 3, no. 23, Oct. 2018, Art. no. eaat8409.

[5] N. Jiang et al., "Anti-UAV: A large-scale benchmark for vision-based UAV tracking," *IEEE Trans. Multimedia*, vol. 25, pp. 486–500, 2023.

[6] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Onboard real-time aerial tracking with efficient Siamese anchor proposal network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022, Art. no. 5606913.

[7] J. Yang, Z. Pan, Z. Wang, B. Lei, and Y. Hu, "SiamMDM: An adaptive fusion network with dynamic template for real-time satellite video single object tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–19, 2023, Art. no. 5608619.

[8] 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.

[9] 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.

[10] 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.

[11] D. Guo, Y. Shao, Y. Cui, Z. Wang, L. Zhang, and C. Shen, "Graph attention tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 9538–9547.

[12] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6727–6736.

[13] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial–temporal attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 548–557.

[14] 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.

[15] B. Huang, T. Xu, J. Li, F. Luo, Q. Qin, and J. Chen, "Learning context restrained correlation tracking filters via adversarial negative instance generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 9, pp. 6132–6145, Sep. 2023.

[16] S. Xuan, S. Li, M. Han, X. Wan, and G.-S. Xia, "Object tracking in satellite videos by improved correlation filters with motion estimations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 2, pp. 1074–1086, Feb. 2020.

[17] 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.

[18] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M.-H. Yang, "CREST: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2574–2583.

[19] F. Wu et al., "Deep Siamese cross-residual learning for robust visual tracking," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15216–15227, Oct. 2021.

[20] L. Huang, X. Zhao, and K. Huang, "GlobalTrack: A simple and strong baseline for long-term tracking," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11037–11044.

[21] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6577–6587.

[22] B. Huang et al., "SiamSTA: Spatio-temporal attention based Siamese tracker for tracking UAVs," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 1204–1212.

[23] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10428–10437.

[24] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1571–1580.

[25] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[26] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[27] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.

[28] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939.

[29] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scaleestimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Nottingham, U.K., Sep. 2014, pp. 65.1–65.11.

[30] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1144–1152.

[31] 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.

[32] 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.

[33] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4665–4674.

[34] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial–temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[35] D. Yuan, X. Chang, Z. Li, and Z. He, "Learning adaptive spatial–temporal context-aware correlation filters for UAV tracking," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 3, pp. 1–18, Aug. 2022.

[36] D. Yuan, X. Chang, P.-Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Trans. Image Process.*, vol. 30, pp. 976–985, 2021.

[37] B. Huang, T. Xu, Z. Shen, S. Jiang, B. Zhao, and Z. Bian, "SiamATL: Online update of Siamese tracking network via attentional transfer learning," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7527–7540, Aug. 2022.

[38] D. Yuan et al., "Active learning for deep visual tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, doi: 10.1109/TNNLS.2023.3266837.

[39] Q. Liu, Z. He, X. Li, and Y. Zheng, "PTB-TIR: A thermal infrared pedestrian tracking benchmark," *IEEE Trans. Multimedia*, vol. 22, no. 3, pp. 666–675, Mar. 2020.

[40] Q. Liu et al., "LSOTB-TIR: A large-scale high-diversity thermal infrared object tracking benchmark," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 3847–3856.

[41] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen, "Deep convolutional neural networks for thermal infrared object tracking," *Knowl.-Based Syst.*, vol. 134, pp. 189–198, Oct. 2017.

[42] X. Li, Q. Liu, N. Fan, Z. He, and H. Wang, "Hierarchical spatial-aware Siamese network for thermal infrared object tracking," *Knowl.-Based Syst.*, vol. 166, pp. 71–81, Feb. 2019.

[43] D. Yuan, X. Shu, Q. Liu, and Z. He, "Aligned spatial–temporal memory network for thermal infrared target tracking," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 70, no. 3, pp. 1224–1228, Mar. 2023.

[44] Q. Liu, X. Li, Z. He, N. Fan, D. Yuan, and H. Wang, "Learning deep multi-level similarity for thermal infrared object tracking," *IEEE Trans. Multimedia*, vol. 23, pp. 2114–2126, 2021.

[45] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, "Synthetic data generation for end-to-end thermal infrared tracking," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1837–1850, Apr. 2019.

[46] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-UAV detection and tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25323–25334, Dec. 2022.

[47] J. Zhao et al., "The 3rd anti-UAV workshop & challenge: Methods and results," 2023, *arXiv:2305.07290*.

[48] Q. Yu, Y. Ma, J. He, D. Yang, and T. Zhang, "A unified transformer-based tracker for anti-UAV tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 3035–3045.

[49] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "HiFT: Hierarchical feature transformer for aerial tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15437–15446.

[50] Z. Fu, Q. Liu, Z. Fu, and Y. Wang, "STMTrack: Template-free visual tracking with space-time memory networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13769–13778.

[51] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8122–8131.

[52] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[54] H. Fan et al., "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378.

[55] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[56] T.-Y. Li et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[57] 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.

[58] A. Lukežic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4847–4856.

[59] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2891–2900.

[60] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11920–11929.

[61] 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.

[62] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–117.

[63] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.

[64] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4655–4664.

[65] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6181–6190.

[66] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, Apr. 2020, pp. 12549–12556.

[67] M. Danelljan, L. Van Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7181–7190.

[68] 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.

[69] 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.
[70] T. Yang, P. Xu, R. Hu, H. Chai, and A. B. Chan, "ROAM: Recurrently optimizing tracking model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6717–6726.
[71] M. Danelljan and G. Bhat. *PyTracking: Visual Tracking Library Based on PyTorch*. Accessed: Aug. 1, 2020. [Online]. Available: https://github.com/visionml/pytracking
[72] G. Bhat, M. Danelljan, L. V. Gool, and R. Timofte, "Know your surroundings: Exploiting scene information for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 205–221.
[73] C. Mayer, M. Danelljan, D. Pani Paudel, and L. Van Gool, "Learning target candidate association to keep track of what not to track," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13424–13434.
[74] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 771–787.
[75] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with Gaussian processes regression," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[76] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.
[77] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic Siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1781–1789.
[78] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[79] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[80] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

Bo Huang received the B.S. and Ph.D. degrees from the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China, in 2016 and 2022, respectively.

From July 2020 to June 2021, he was an Algorithm Engineer with the Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. He is currently an Associate Professor with Chongqing University, Chongqing. His research interests include computer vision, artificial intelligence, and machine learning.

Zeyang Dou received the B.E. degree in mathematics from Baoding University, Baoding, China, in 2012, the M.S. degree in computational mathematics from the Communication University of China, Beijing, China, in 2016, and the Ph.D. degree from the Beijing Institute of Technology, Beijing, in 2020.

His research interests include image processing, machine learning, and deep learning.

Junjie Chen is currently pursuing the M.S. degree with the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China.

His research interests include object tracking and related computer vision problems.

Jianan Li received the B.S. and Ph.D. degrees from the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China, in 2013 and 2019, respectively.

From July 2015 to July 2017, he worked as a joint training Ph.D. Student with the National University of Singapore, Singapore. From October 2017 to April 2018, he was an Intern with Adobe Research, San Jose, CA, USA. He is currently an Assistant Professor with the School of Optics and Photonics, Beijing Institute of Technology. His research interests include computer vision and real-time image/video processing.

Ning Shen is currently pursuing the Ph.D. degree with the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China.

His research interests include medical optical imaging and related computer vision problems.

Ying Wang received the B.E. degree in optical engineering from the Beijing Institute of Technology, Beijing, China, in 2019, where she is currently pursuing the Ph.D. degree.

Her research interests include single object tracking and object detection.

Tingfa Xu received the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Changchun, China, in 2004.

He is currently a Professor with the School of Optics and Photonics, Beijing Institute of Technology, Beijing, China. He is also the Director of the Big Data and Artificial Intelligence Laboratory, Beijing Institute of Technology Chongqing Innovation Center, Chongqing, China. His research interests include optoelectronic imaging and detection and hyper-spectral remote sensing image processing.