frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 01 December 2022
DOI 10.3389/fnbot.2022.1082346

OPEN ACCESS

EDITED BY
Xin Ning,
Institute of Semiconductors
(CAS), China

REVIEWED BY
Sahraoui Dhelim,
University College Dublin, Ireland
Nadeem Javaid,
COMSATS University, Islamabad
Campus, Pakistan
Achyut Shankar,
Amity University, India

*CORRESPONDENCE
Yuchun Chang
cyc@dlut.edu.cn

RECEIVED 28 October 2022
ACCEPTED 14 November 2022
PUBLISHED 01 December 2022

CITATION
Dai J, Fu Y, Wang S and Chang Y (2022) Siamese hierarchical feature fusion transformer for efficient tracking. *Front. Neurorobot.* 16:1082346. doi: 10.3389/fnbot.2022.1082346

Siamese hierarchical feature fusion transformer for efficient tracking

Jiahai Dai[1], Yunhao Fu[1], Songxin Wang[2] and Yuchun Chang[1,3]*

[1]Department of Electronic Information Engineering, College of Electronic Science and Engineering, Jilin University, Changchun, China, [2]Department of Computer Science and Technology, College Science and Technology, Shanghai University of Finance and Economics, Shanghai, China, [3]Department of Electronic Science and Technology, School of Microelectronics, Dalian University of Technology, Dalian, China

Object tracking is a fundamental task in computer vision. Recent years, most of the tracking algorithms are based on deep networks. Trackers with deeper backbones are computationally expensive and can hardly meet the real-time requirements on edge platforms. Lightweight networks are widely used to tackle this issue, but the features extracted by a lightweight backbone are inadequate for discriminating the object from the background in complex scenarios, especially for small objects tracking task. In this paper, we adopted a lightweight backbone and extracted features from multiple levels. A hierarchical feature fusion transformer (HFFT) was designed to mine the interdependencies of multi-level features in a novel model—SiamHFFT. Therefore, our tracker can exploit comprehensive feature representations in an end-to-end manner, and the proposed model is capable of handling small target tracking in complex scenarios on a CPU at a rate of 29 FPS. Comprehensive experimental results on UAV123, UAV123@10fps, LaSOT, VOT2020, and GOT-10k benchmarks with multiple trackers demonstrate the effectiveness and efficiency of SiamHFFT. In particular, our SiamHFFT achieves good performance both in accuracy and speed, which has practical implications in terms of improving small object tracking performance in the real world.

KEYWORDS

visual tracking, hierarchical feature, transformer, lightweight backbone, real-time

Introduction

Visual tracking is an important task in computer vision that provides underlying technical support for more complex tasks; and is an essential procedure for advanced computer vision applications. Additionally, visual tracking has been widely used in various fields such as unmanned aerial vehicles (UAVs) (Cao et al., 2021), autonomous driving (Zhang and Processing, 2021), and video surveillance (Zhang G. et al., 2021). However, several challenges remain that hamper tracking performance, including edge computing devices and difficult external environments with occlusion, illumination variation, and background clutter.

Over the past few years, visual object tracking has made significant advancements based on the development of convolutional neural networks due to the breakthroughs that have been made to generate more powerful backbones, such as deeper networks (He et al., 2016; Chen B. et al., 2022), efficient network structure (Howard et al., 2017), attention mechanism (Hu et al., 2018). Inspired by the way of the human brain process the overload information (Wolfe and Horowitz, 2004), the attention mechanism is utilized to enhance the vital features and surpass the unnecessary information of the input feature. Due to the powerful feature representation ability, the attention mechanism becomes an important means to enhance the input features, such as channel attention (Hu et al., 2018), spatial attention (Wang F. et al., 2017; Wang N. et al., 2018), temporal attention (Hou et al., 2020), global attention (Zhang et al., 2020a), and self-attention mechanism (Wang et al., 2018). Among them, the self-attention based models, the transformer was initially designed for natural language processing (NLP) (Vaswani et al., 2017) task, where the attention mechanism is utilized to perform the machine translation tasks and achieved great improvements. Later, the pre-training model BERT (Devlin et al., 2018) achieve breakthrough progress in NLP tasks, further advance the development of the Transformer model. Since then, both academia and industry have set off a boom in the research and application of pre-trained models based on Transformer, and gradually extended from NLP to CV. For example, Vision Transformer (ViT) (Dosovitskiy et al., 2020), DETR (Carion et al., 2020), have surpassed previous SOTA in the fields of image classification, inspection, and video, respectively. Various variant models based on Transformer structure have been proposed, multi-task indicators in various fields have been continuously refreshed, and the deep learning community has entered a new era. Meanwhile, muti-level features fusion can effectively alleviate the deficiency of the transformer in handling the tracking of small objects.

Although transformer models provide enhancements in feature representation and result in promotion in terms of accuracy and robustness, trackers based on transformers have high computational costs that hinder them from meeting the real-time demands of tracking tasks on edge hardware devices, providing a disadvantage for the landing of the application. Therefore, how to balance the efficiency and efficacy of object trackers remains a significant challenge. Generally, discriminative feature representation is essential for tracking. Therefore, deeper backbones and online updaters are utilized in tracking frameworks, however these methods are computationally expensive leading to increased run time and budget. Typically, the lightweight backbone is also limited as it typically provides inadequate feature extraction, rendering the tracking model less robust for small objects or complex scenarios.

In this study, we employed a lightweight backbone network to avoid the efficiency loss caused by the computations of deep networks. To address the insufficient feature representations extracted by shallow networks, we extracted features from multiple levels of the backbone to enrich the feature representations. Furthermore, to leverage the advantages of transformers in global relationship modeling, we designed a hierarchical feature fusion module to integrate multi-level features comprehensively using multi-head attention mechanisms. The proposed Siamese hierarchical feature fusion transformer (SiamHFFT) tracker achieved robust performance in complex scenarios while maintaining real-time tracking speed on a CPU and it can be deployed on consumer CPUs. The main contributions of this study can be summarized as follows:

(1) We proposed a novel type of tracking network based on a Siamese architecture, which consisting of feature extraction, reshape module, Transformer-like feature fusion module, and head prediction modules.

(2) We designed a feature fusion transformer to exploit the hierarchical features in the Siamese tracking framework in an end-to-end manner, which is capable of advancing discriminability for small object tracking task.

(3) Comprehensive evaluations on five challenging benchmarks demonstrate the proposed tracker achieved promising results among state-of-the-art trackers. Besides, our tracker can run at a real-time speed. This efficient method can be deployed on resource-limited platforms.

The remainder of this paper is organized as follows. Section Related work describes related work on tracking networks and transformers. Section Method introduces the methodology used for implementing the proposed HFFT and network model. Section Experiments presents the results of experiments conducted to verify the proposed model. Finally, Section Conclusion contains our concluding remarks.

Related work

Siamese tracking

In recent years, Siamese-based networks have become a ubiquitous framework in the visual tracking field (Javed et al., 2021). Tracking an arbitrary object can be considered as learning similarity measure function learning problems. SiamFC (Bertinetto et al., 2016) introduced a correlation layer as a fusion tensor into the tracking framework for the first time, which pioneered the Siamese tracking procedure. Instead of directly estimating the target position according to the response map, SiamRPN (Li B. et al., 2018) attaches a region proposal extraction subnetwork (RPN) to the Siamese network and formulates the tracking as a one-shot detection task. Based on the results of classification and regression branches, SiamRPN achieves enhanced tracking accuracy. DaSiamRPN (Zhu et al.,

2018) uses a distractor-aware module to solve the problem of inaccurate tracking caused by the imbalance of positive and negative samples of the training set. C-RPN (Fan and Ling, 2019) and Cract (Fan and Ling, 2020) incorporate multiple stages into the Siamese tracking architecture to improve tracking accuracy. To address unreliable predicted fixed-ratio bounding boxes when a tracker drifts rapidly, an anchor-free mechanism was also introduced into the tracking task. To rectify the inaccurate bounding box estimation strategy of the anchor-based mechanism, Ocean (Zhang et al., 2020b) directly regresses the location of each point located in the ground truth. SiamBAN (Chen et al., 2020) adopts box adaptive heads to handle the classification and regression problem parallelly. SiamFC++ (Xu et al., 2020) and SiamCAR (Guo et al., 2020) draw on the FCOS architecture and add a branch to measure the accuracy of the classification results. Compared with anchor-based trackers, anchor-free-based trackers utilize fewer parameters and do not need prior information for the bounding box, these anchor-free-based trackers can achieve a real-time speed.

As feature representation plays a vital role in the tracking process (Marvasti-Zadeh et al., 2021), several works delicate to obtain discriminative features from different perspectives, such as adopting deeper or wider backbones, and using attention mechanisms to advance the feature representation. In the recent 3 years, the Transformer is capable of using global context information and preserving more semantic information. The introduction of the Transformer model in the tracking community boots the tracking accuracy to a great extent (Chen X. et al., 2021; Lin et al., 2021; Liu et al., 2021; Chen et al., 2022b; Mayer et al., 2022). However, the promotion of the accuracy of these trackers' increasingly complex models relies heavily on powerful GPUs, leading to the inability to deploy such models on edge devices, which hinders the further practical application of the models.

In this study, to optimize the trade-off between tracking accuracy and speed, we designed an efficient algorithm that employs a concise model consisting of a lightweight backbone network, a feature reshaping model, a feature fusion module, and a prediction head. Our model is capable of handling complex scenarios, and the proposed tracker can also achieve real-time speed on a CPU.

Transformer in vision tasks

As a new type of neural network, transformer shows superior performance in the field of AI applications (Han et al., 2022). Unlike the structure of CNNs and RNNs, Transformer adopts the self-attention mechanism, which has been proved to have strong feature representation ability and better parallel computing capability, making it more advantageous in several tasks.

The transformer model was first proposed by Vaswani et al. (2017) for application to natural language processing (NLP) tasks. In contrast to convolutional neural networks (CNNs) and recurrent neural networks (RNNs), self-attention facilitates both parallel computation and short maximum path lengths. Unlike earlier self-attention models based on RNNs for input representations (Lin Z. et al., 2017; Paulus et al., 2017), the attention mechanisms in transformer model are implemented with attention-based encoders and decoders instead of convolutional or recurrent layers.

Because transformers were originally designed for sequence-to-sequence learning on textual data and have exhibited good performance, their ability to integrate global information has been gradually unveiled and transformers have been extended to other modern deep learning applications such as image classification (Liu et al., 2020; Chen C. -F. R. et al., 2021; He et al., 2021), reinforcement learning (Parisotto et al., 2020; Chen L. et al., 2021), face alignment (Ning et al., 2020), object detection (Beal et al., 2020; Carion et al., 2020), image recognition (Dosovitskiy et al., 2020) and object tracking (Yan et al., 2019, 2021a; Cao et al., 2021; Lin et al., 2021; Zhang J. et al., 2021; Chen B. et al., 2022; Chen et al., 2022b; Mayer et al., 2022). Based on CNNs and transformers, the DERT (Carion et al., 2020) applies a transformer to object detection tasks. To improve upon previous CNN models, DERT eliminates post-processing steps that rely on manual priors such as non-maximum suppression (NMS) and anchor generators; and constructs a complete end-to-end detection framework. ViT (Dosovitskiy et al., 2020) mainly converts images into serialized data through token processing and introduces the concept of patches, where input images are divided into smaller patches and each patch is converted into a bidirectional encoder representation from transformers-like structure. Similar to the concept of patches in ViT, Swin Transformer (Liu et al., 2021) uses the concept of windows, but the calculations of different windows do not interfere with each other, hence, the computational complexity of the Swin Transformer is significantly reduced.

In the tracking community, transformers have achieved remarkable performance. STARK (Yan et al., 2021a) utilizes an end-to-end transformer tracking architecture based on spatiotemporal information. SwinTrack (Lin et al., 2021) incorporates a general position-encoding solution for feature extraction and feature fusion, enabling full interaction between the target object and search region during tracking process. TrTr (Zhao et al., 2021) used the transformer architecture to perform target classification and bounding box regression and designed a plug-in online update module for classification to further improve tracking performance. DTT (Yu et al., 2021) also feed these architectures to predict the location and the bounding box of the target. Cao et al. (2021) proposed an efficient and effective hierarchical feature transformer (HiFT) for aerial tracking. HCAT (Chen et al., 2022b) utilizes a novel feature sparsification module to reduce computational complexity and a hierarchical cross-attention transformer that employs a full cross-attention structure to improve efficiency and enhance representation ability. The hierarchical-based methods, both HiFT and HCAT show good tracking performance. However, transformer-based trackers lack robustness in small objects. In this paper, we propose a novel hierarchical feature fusion module based on a transformer to enable a tracker to achieve real-time speed while maintains good accuracy.

Feature aggregation network

Feature aggregation plays a vital role in the multi-level feature process, and is used to improve cross-scale feature interaction and multi-scale feature fusion, thereby enhancing the representation of features and enhancing network performance. Zhang G. et al. (2021) proposed a hierarchical aggregation transformer (HAT) framework consisting of transformer-based feature calibration (TFC) and deeply supervised aggregation (DSA) modules. The TFC module can merge and preserve semantic and detail information at multiple levels, and the DSA module aggregates the hierarchical features of the backbone with multi-granularity supervision. Feature pyramid networks (FPN) (Lin T.-Y. et al., 2017) introduce cross-scale feature interactions and achieve good results through the fusion of multiple layers. Qingyun et al. (2021) introduced a cross-modality fusion transformer, that makes full use of the complementarity between different modalities to improve the performance of features. However, the main challenge of a simple feature fusion strategy is how to fuse high-level semantic information and low-level detailed features. To address these issues, we propose an aggregation structure based on hierarchical transformers, which can fully mine the coherence among multi-level features at different scales, and achieve discriminative feature representation ability.

Method

Overview

In this section, we describe the proposed SiamHFFT model. As can be seen in Figure 1, our model follows a Siamese tracking framework. There are four key components in our model, namely the feature extraction module, reshape module, feature fusion module, and prediction head. During tracking, the feature extraction module extracts feature from the template and search region. The features of the two branches from the last three layers of the backbone are correlated separately, and the outputs are denoted as M_2, M_3, and M_4 in order. We then feed the correlated features into the reshaping module, which can transform the channel dimensions of the backbone features and flatten features in the spatial dimension. The feature fusion module is implemented by fusing features using our hierarchical feature fusion transformer (HFFT) and a self-attention module. Finally, we used the prediction head module to perform bounding box regression and binary classification on the enhanced features to generate tracking results.

Feature extraction and reshaping

Similar to most Siamese tracking networks, the proposed method uses template frame patch ($Z \in \mathbb{R}^{3\times80\times80}$) and search frame patch ($X \in \mathbb{R}^{3\times320\times320}$) as inputs. For the backbone, our method can use an arbitrary deep CNN such as ResNet, MobileNet (Sandler et al., 2018), AlexNet, or ShuffleNet V2 (Ma et al., 2018). In this study, because a deeper network is unsuitable for deployment with limited computing resources, we adopted ShuffleNetV2 as a backbone network. This network is utilized for both template and search branch feature extraction.

To obtain robust and discriminative feature representations, we incorporate detailed structural information into our visual representations by extracting hierarchical features with different scales and semantic information in stage two, three and four of feature extraction. We denote feature tokens from the template branch as $F_i(Z)$ and those from the search branch as $F_i(X)$, where i represents the stage number of feature extraction and $i \in \{2, 3, 4\}$.

Next, a convolution operation is performed on the feature maps from the multi stages correlation, which is defined as:

$$M_i = F_i(Z) * F_i(X), i = 2, 3, 4, \tag{1}$$

where $M_i \in \mathbb{R}^{C_i\times H_i\times W_i}$, and C, H, and W denote the channel, width, and height of the feature map respectively. Additionally, $C_i \in \{116, 232, 464\}$ and $*$ denotes the cross-correlation operator. Next, we use the reshaping module which consists of 1×1 convolutional kernels, to change the channel dimensions of the features from Equation (1). We then flatten the features in the spatial dimension because a unified channel can not only effectively reduce computing resource requirements, but is also an essential component for improving the performance of feature fusion. After these operations, we can obtain a reshaped feature map $M_i{}' \in \mathbb{R}^{W_iH_i\times C}$, where $C = 192$.

Feature fusion and prediction head

As illustrated in Figure 1, following the convolution and flattening operations in the reshaping module, the correlation features from different stages are unified in the channel dimension. To explore the interdependencies among multi-level features fully, we designed the HFFT, which is detailed in this section.

Multi-Head Attention (Vaswani et al., 2017): Generally, transformers have been successfully applied to enhance feature

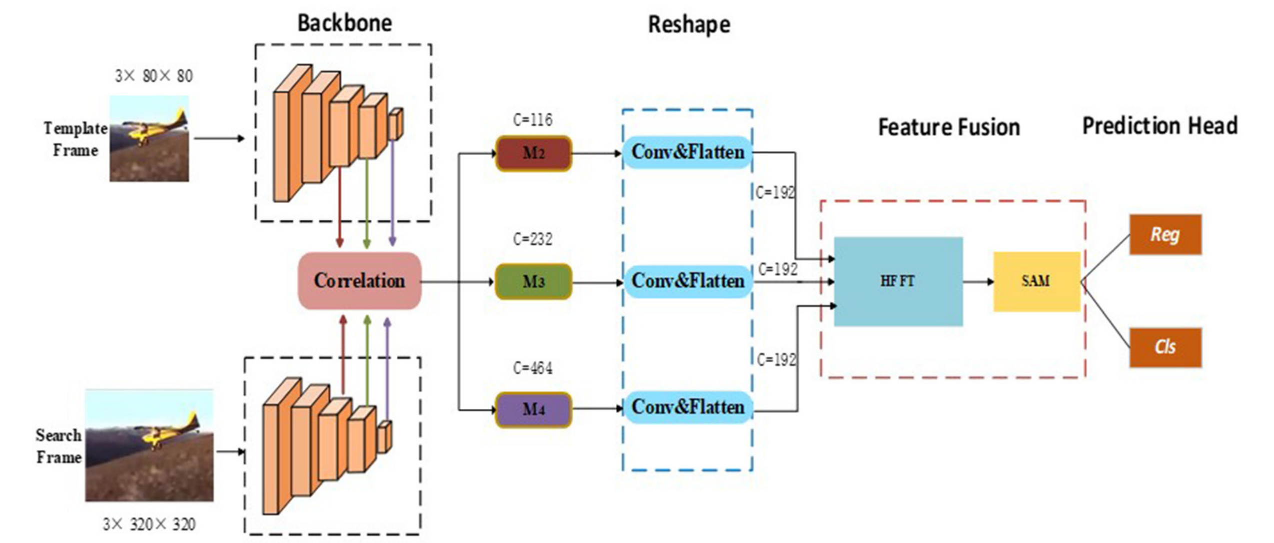

FIGURE 1
Architecture of the proposed SiamHFFT tracking framework. This framework contains four fundamental components: a feature extraction network, reshaping module, feature fusion module, and prediction head. The backbone network is used to extract hierarchical features. The reshaping module is designed to perform convolution operations and flatten features. The feature fusion transformer consists of the proposed HFFT module and a self-attention module (SAM). Finally, bounding boxes are estimated based on the regression and classification results.

representations in various bi-modal vision tasks. In the proposed feature fusion module, the attention mechanism is also a fundamental component. It is implemented using an attention function and operated on queries Q, keys K and values V using the scale dot-production method, which is defined as:

$$Attention(Q, K, V) = softmax(\frac{QK^{\top}}{\sqrt{C}})V \tag{2}$$

where C is the key dimensionality for normalizing the attention, and$\sqrt{C}$ is a scaling factor to avoid gradient vanishing in the loss function. Specifically, $Q = [q_1, \ldots, q_N]^T \in \mathbb{R}^{N\times C}$ is the q input in Figure 2B, which denotes a collection of N features; similarly, K and V are the k and v inputs, respectively, which represent a collection of Mfeatures (i.e., $K, V \in \mathbb{R}^{M\times C}$). Notably, Q, K, V represent the mathematical implementation of the attention function and do not have practical meaning.

According to Vaswani et al. (2017), extending the attention function in Equation (2) to multiple heads is beneficial for enabling the mechanism to learn various attention distributions and enhancing its feature representation ability. This extension can be formulated as follows:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots head_h)W^o \tag{3}$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V), i = 1, \ldots h \tag{4}$$

where W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{C\times d_h}$, and $W^o \in \mathbb{R}^{C\times C}$. Here, h is the number of attention heads, which is defined as $d_h = \frac{C}{h}$. In this study, we adopted and $h = 6$ as default values.

Application to Dual-Input Tasks: The structure of a dual-input task is presented in Figure 2A, where Q, K, and V for normal NLP/vision tasks (Nguyen et al., 2020) share the same modality. In recent years, this mechanism has been extended to dual-inputs and applied to vision tasks (Chen X. et al., 2021; Chen et al., 2022a,b). However, the original attention mechanism cannot distinguish between the position information of different input feature sequences. The original mechanism only considers the absolute position and adds absolute positional encodings to inputs. It considers the attention from a source feature ϕ to a target feature θ as:

$$A_{\phi}(\theta) = MultiHead(\theta + P_{\theta}, \phi + P_{\phi}, \phi) \tag{5}$$

where P_{θ} and P_{ϕ} are the spatial positional encodings of features θ and ϕ, respectively. Spatial positional encoding is generated using a sine function. Equation (5) can be used not only as a single-direction attention enhancement, but also as a co-attention mechanism in which both directions are considered. Furthermore, self-attention from a feature to itself is also defined as a special case:

$$A_{\theta}(\theta) = MultiHead(\theta + P_{\theta}, \theta + P_{\theta}, \theta) \tag{6}$$

As shown in Figure 2A, following Equations (5) and (6), the designed transformer blocks are processed independently.

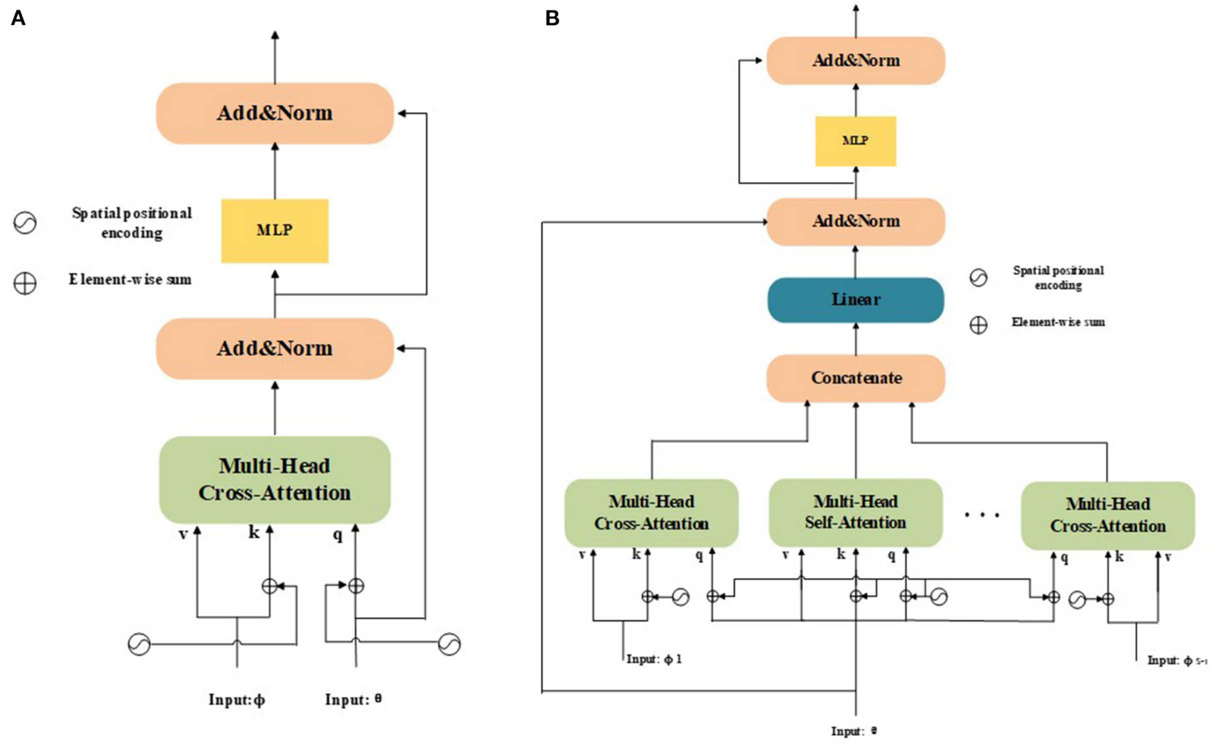
A
Add&Norm
MLP
Spatial positional encoding
Element-wise sum
Add&Norm
Multi-Head Cross-Attention
v
k
q
Input: ϕ
Input: θ
B
Add&Norm
MLP
Add&Norm
Linear
Concatenate
Multi-Head Cross-Attention
Multi-Head Self-Attention
Multi-Head Cross-Attention
Input: ϕ 1
Input: ϕ S-1
Input: θ

FIGURE 2
(A) Structure of a dual-input tasks; **(B)** Structure of a multi-input tasks. Unlike the original dual-input tasks, multi-input tasks can be used to learn the interdependencies of multi-level features and enhance the feature representation of the model in an end-to-end manner.

Therefore, the two modules can be used sequentially or in parallel. Additionally, a multilayer perceptron (MLP) module is used to enhance the fitting ability of the model. The MLP module is a fully connected network consisting of two linear projections with a Gaussian error linear unit (GELU) activation function between them, which can be denoted as:

$$MLP(\theta') = FC_2(GELU(FC_1(\theta'))) \tag{7}$$

Application to Multi-Input Tasks: To extend the attention mechanism to multiple inputs that are capable of handling multimodal vision tasks, pyramid structures, etc., we denote the total input number as S. The structure of a multi-input task is presented in Figure 2B. If we consider each possibility, there are a total of $S(S-1)$ source-target cases and S self-attention cases. Now, we denote the multiple inputs as $\{\theta, \phi_1, \ldots, \phi_{S-1}\}$, where the target $\theta \in \mathbb{R}^{N\times C}$ and source $\phi_i \in \mathbb{R}^{M\times C}$. Notably, θ and ϕ_i must have the same size as C. We then compute all the source-target cases as $\{A_{\phi_1}(\theta), \ldots, A_{\phi_{S-1}}(\theta)\}$. Next, we concatenate all source-to-target attention cases with self-attention $A_\theta(\theta)$, which can be formulated as:

$$\theta_{concat} = [A_\theta(\theta), A_{\phi_1}(\theta), \ldots, A_{\phi_{S-1}}(\theta)] \tag{8}$$

where $\theta_{concat} \in \mathbb{R}^{N\times SC}$. After concatenation, the dimensions of the enhanced features in the channel change to match the size SC of the original feature. To accelerate these calculations further, we apply a fully connected layer to reduce the channel dimensions to:

$$\theta_{concat}{}' = Linear[\theta_{concat}] \tag{9}$$

where $\theta_{concat}{}' \in \mathbb{R}^{N\times C}$. Through this process, we can obtain more discriminative features efficiently by aggregating features from different attention mechanisms.

HFFT: As is shown in Figure 2B, in our model, we make full use of the hierarchical features $M_i{}' \in \mathbb{R}^{W_iH_i\times C}$ $(i \in \{2,3,4\})$ and generate tracking-tailored features. To integrate low-level spatial information with high-level semantic information, we feed the reshaped features from the output of Equation (1), namely $M_2{}'$, $M_3{}'$, and $M_4{}'$, into the HFFT module, where $M_3{}'$ is used for target feature, $M_2{}'$ and $M_4{}'$ represent source features. The importance of different aspects feature information is assigned by applying the cross-attention operator to $M_2{}'$ and $M_4{}'$, which is beneficial for obtaining more discriminative features. We apply self-attention to $M_3{}'$, which can preserve the details of target information during tracking. Furthermore, positional information is encoded during the calculation process to enhance spatial information during the tracking process. The attention mechanisms are implemented using the operation of K, Q, V. Then, comprehensive features can be obtained by concatenating the outputs. Due to the complexity of a model increases with its input size, a fully connected layer is utilized to resize outputs. We also adopt residual connections around each sub-layer. Additionally, we use an MLP module to enhance the fitting ability of the model, and layer normalization (LN) is performed before the MLP and final output steps. The entire process of the HFFT can be expressed as:

$$M_{concat} = [A_{M_3'}(M_3'), A_{M_2'}(M_3'), A_{M_4'}(M_3')],$$

$$M_{concat}' = Linear[M_{concat}],$$

$$M_{out} = LN(M_{concat}' + M_3'),$$

$$X_{out} = LN(M_{out} + MLP(M_{out})) \tag{10}$$

SAM: The SAM is a feature enhancement module. The structure of the SAM is presented in Figure 3. The SAM adaptively integrates information from different feature maps using multi-head self-attention in the residual form. In the proposed model, the SAM take the out of Equation (10) X_{out} as input. The mathematical process of the SAM can be summarized as:

$$X_{out2} = LN(MultiHead(X_{out} + P_X, X_{out} + P_X, X_{out}) + X_{out}),$$

$$X_{SAM} = LN(MLP(X_{out2}) + X_{out2}) \tag{11}$$

Prediction Head: The enhanced features are reshaped back to the original feature size before being fed into the prediction head. The head network consists of two branches: a classification branch and bounding box regression branch. Each branch consists of a three-layer perceptron. The former is utilized to distinguish the target from the background, and the latter is used for estimating the location of the target by using a bounding box. Overall, the model is trained using a combination loss function formulated as:

$$L = \lambda_{cls} L_{cls} + \lambda_{giou} L_{giou} + \lambda_{loc} L_{loc} \tag{12}$$

where L_{cls}, L_{giou}, and L_{loc} represent the binary cross-entropy, GIOU loss, and L1-norm loss, respectively. λ_{cls}, λ_{giou}, and λ_{loc} are coefficients that balance the contributions of each type of losses.

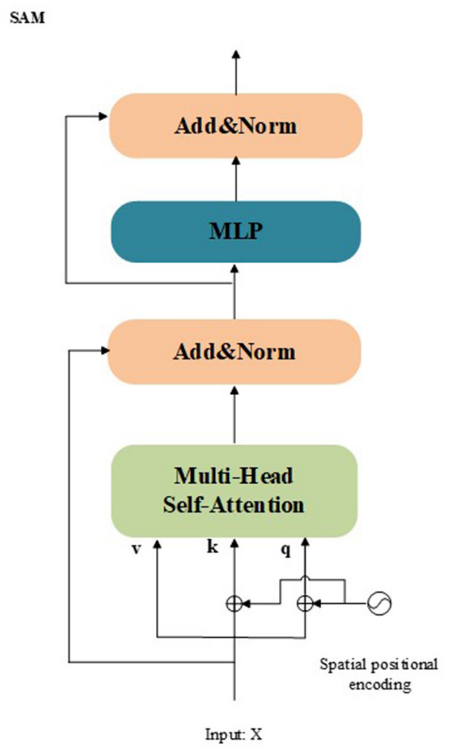

FIGURE 3
Architecture of the proposed SAM.

Experiments

This section presents the details of the experimental implementation of the proposed model. To validate the performance of the proposed tracker, we compared our method to state-of-the-art methods on four popular benchmarks. Additionally, ablation studies were conducted to analyse the effectiveness of key modules.

Implementation details

The tracking algorithm was implemented in Python based on PyTorch. The proposed model was trained on a PC with an Intel i7-11700k, 3.6 GHz CPU, 64 GB of RAM, and an NVIDIA 3080Ti RTX GPUs. The training splits of LaSOT (Fan et al., 2019), GOT-10k (Huang et al., 2019), COCO (Lin et al., 2014), and TrackingNet (Muller et al., 2018) were used to train the model. We randomly selected two image pairs from the same video sequences with a maximum gap of 100 frames to generate the search patches and template patches. The sizes of search patches were set to 320 × 320 × 3 and template patches were resized to sizes of 80 × 80 × 3. The parameters for the backbone network were initialized using ShuffleNetV2, which was pretrained on ImageNet. All models were trained for 150 epochs with a batch size of 32. Each epoch contained 60,000 sampling pairs. The coefficient parameters in Equation (12) were set to $\lambda_{cls} = 2$, $\lambda_{giou} = 2$, and $\lambda_{loc} = 5$. In the offline training phrase, the parameters of the model are optimized by ADAMW optimizer. The learning rates of the backbone network were set to le-5, and le-4 for the remaining parts.

Comparisions to state-of-the-art methods

We compared SiamHFFT to state-of-the-art trackers on four benchmarks: LaSOT, UAV123 (Mueller et al., 2016), UAV123@10fps, and VOT2020 (Kristan et al., 2020). The evaluation results are presented in the following paragraphs. It is worthy note that the performance (accuracy and success scores) of the comparision methods on these compared benchmarks are obtained by the public tracking results files, which are released by their authors.

Evaluation on LaSOT: LaSOT is a large-scale long-term tracking benchmark consisting of 1,400 sequences. We used test splits and the one pass evaluation (OPE) to evaluate the performances of the compared trackers. That is, initialize the tracking algorithm according to the target position given in the first frame of the video sequence, and then run the prediction of the target position and size in the whole video to obtain the tracking accuracy or success rate.

Figures 4, 5 report the plots of the precision and success scores of the comparision trackers, respectively. The precision score measures the center location error (CLE), which calculates the average Euclidean distance between the estimated bounding box and the ground truth. The CLE is calculated as follows:

$$CLE = \sqrt{(x_a - x_b)^2 + (x_a - x_b)^2} \tag{13}$$

As the CLEs of frame are obtained, the precision plots (Figure 4) show the percentage of frames in which the estimated CLE is lower than a certain threshold (usually set to 20 pixels) in the total frames of the video sequence.

The Success curve (Figure 5) refers to the percentage of the number of frames whose predicted overlap rate between the estimated bounding box and the ground truth is higher than the given threshold (usually set to 0.5) to the total number of frames in the video sequence. The overlap rate is calculated as follows:

$$S = \frac{|b_t \cap b_g|}{|b_t \cup b_t|} \tag{14}$$

where b_t denotes the estimated bounding box, b_g represents the ground truth bounding box, $\cap$ refers to intersection operator, $\cup$ stands for union operator, and || denotes the number of pixels in the resulted region.

The curves of the proposed SiamHFFT are depicted in green. Overall, our tracker ranks the third in precision, and achieves the second-best score in success, with 61% at the precision score and 62% success score. Compared with the trackers with deeper backbones, such as SiamCAR, SiamBAN, and SiamRPN++ (Li B. et al., 2019), our tracker exhibits competitive performance with a lighter structure. The DiMP achieves the best performance both in precision and success. Our SiamHFFT tracker outperforms other Siamese-based trackers, even with deeper backbones and dedicated-designed structures.

Evaluation on UAV123: UAV123 is an aerial tracking benchmark consisting of 123 videos containing small objects, target occlusions, out of view, and distractors. To validate the performance of our tracker, we evaluated the performances of our trackers and other state-of-the-art trackers, including SiamFC, ECO (Danelljan et al., 2017), ATOM (Danelljan et al., 2019), SiamAttn (Yu et al., 2020), SiamRPN++, SiamCAR, DiMP (Bhat et al., 2019), SiamBAN, and HiFT. Table 1 lists the results in terms of success, precision, and speed on GPU. Additionally, the backbones of the trackers are also reported for an intuitive comparision. The best performance for each criterion is indicated in red.

Among the trackers, those with deeper backbones, such as DiMP, ATOM, and SiamBAN, achieve better performance in term of both precision and success rate. SiamFC, HiFT, and the proposed SiamHFFT utilize lightweight backbone. SiamFC achieves the best performance in speed, but this naive network structure does not achieve satisfactory results in terms of precision and success rate. HiFT adopts a feature transformer to enhance feature representations. Compared to HiFT, our tracker exhibits a clear advantage in term of precision (82.8 vs. 78.7%) and success rate (62.5 vs. 58.9%), which validates the effectiveness of the proposed tracker. According to the last row in Table 1, all compared trackers can run in real-time on a GPU at an average speed of 68 FPS, proving that SiamHFFT maintains a suitable balance between performance and efficiency.

Figure 6 depicts the qualitative results by multiple algorithms on a subset of sequences in UAV123 benchmarks. We choose three sets of the challenging video sequences: Car18_1, Person21_1, and Group3_4_1. All of the three video sequences are shot by the camera of the UAV, the video frames undergo multiple challenges, for example scale variation, changes of different viewpoint, and so on. Generally, the given target appears in small size during the tracking process. The bounding boxes estimated by the trackers are marked in different colors to give an intuitive contrast. The bounding box of our SiamHFFT is shown in red, and it is obvious that our tracker can handle these complex scenarios well, especially for the small object tracking task.

UAV123@10fps: UAV123@10fps is a subset of UAV123 obtained by down-sampling the original videos with an image

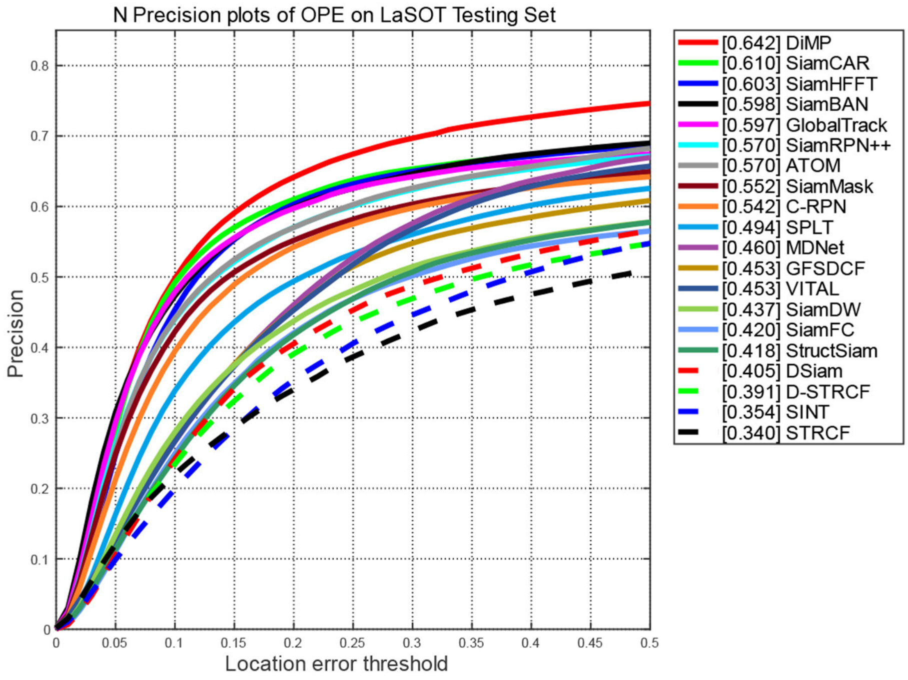

FIGURE 4
Precision scores of compared trackers on LaSOT.

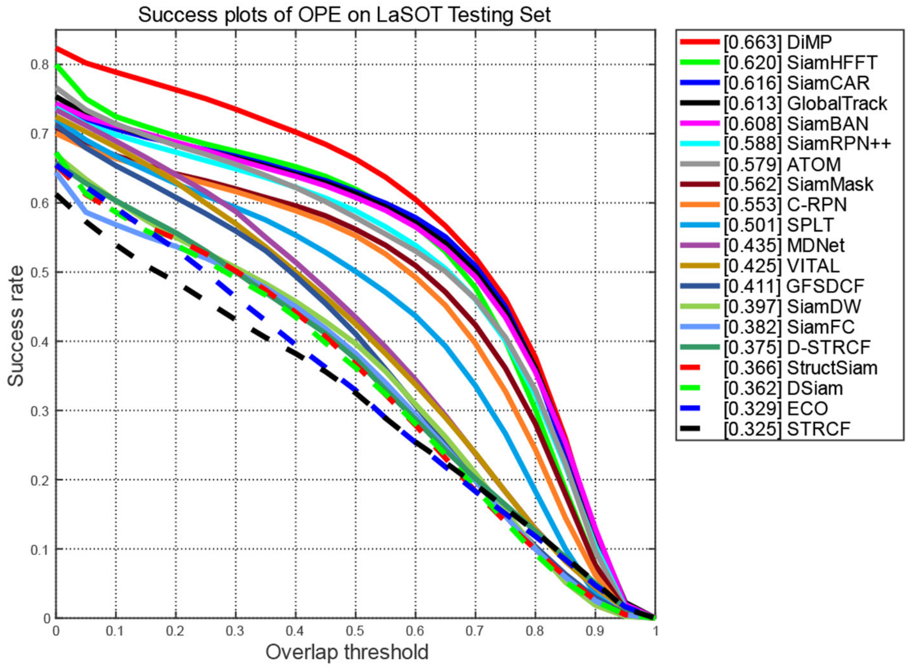

FIGURE 5
Success scores of compared trackers on LaSOT.

TABLE 1 Quantitative evaluation on UAV123 in term of precision (Prec.), success (Succ.) and GPU speed (FPS).

	SiamFC	ECO	ATOM	SiamAttn	SiamRPN++	SiamCAR	DiMP	SiamBAN	HiFT	SiamHFFT
Feat.	Alex	VGG	R18	R50	R50	R50	R50	R50	Alex	ShuffleNet
Prec.	72.5	75.2	83.7	84.5	76.9	76	84.9	83.3	78.7	82.9
Succ.	49.4	52.8	64.2	65	57.9	61.4	65.4	63.1	58.9	62.6
FPS	130	45	46	45	35	52	45	40	/	68

The best performance are shown in red.

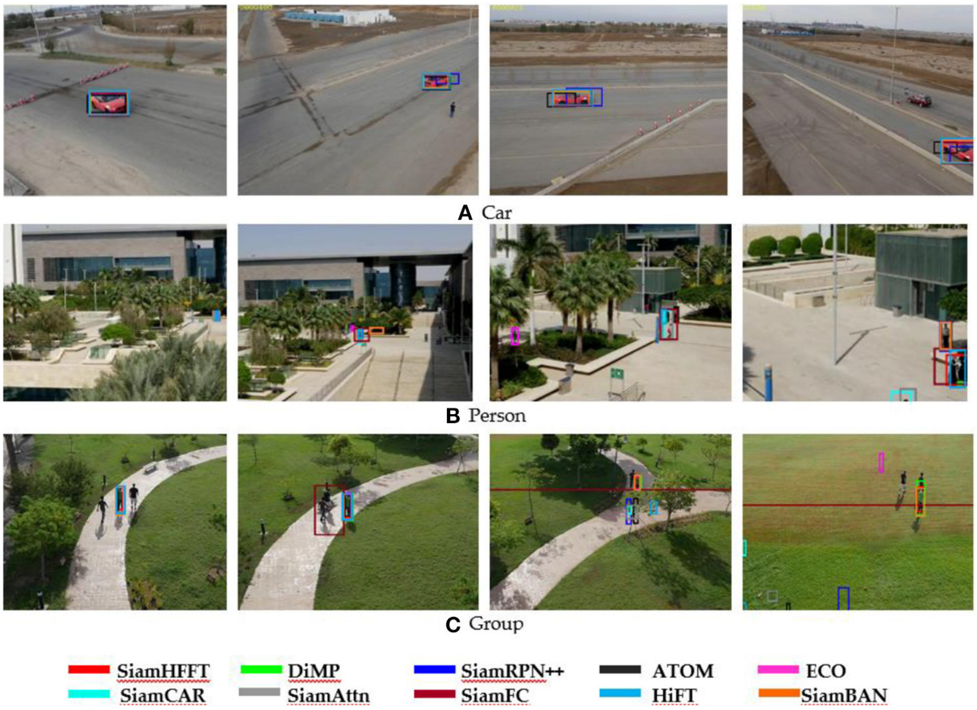

FIGURE 6
Qualitative experimental results in several challenging sequences on UAV123 dataset. **(A)** Video sequences of the Car, **(B)** video sequences of the Person, and **(C)** video sequences of the Group.

rate of 10 FPS. We use SiamFC, AutoTrack (Li et al., 2020), TADT (Li X. et al., 2019), MCCT (Wang et al., 2018), SiamRPN++, DeepSTRCF (Li F. et al., 2018), CCOT (Danelljan et al., 2016), ECO, and HIFT as comparisions. Among these trackers, AutoTrack, TADT, MCCT, CCOT, ECO and DeepSTRCF are correlation filter based trackers, which has a lightweight structure and less parameters than deep learning based trackers, and the model can be deployed on limited source device. Compared with UAV 123 benchmark, challenge in UAV123@10fps dataset are more abrupt and severe. The experimental results are listed in Table 2. Compared with the correlation filter based trackers, the deep trackers, HiFT and SiamRPN++ achieve higher precision and success scores, the performance of SiamFC is closer to these correlation based trackers, SiamFC utilize the AlexNet as the backbone, but the model does not further enhance the feature representation. Our SiamHFFT model yields the best precision (76.5%) and success rate (59.5%), which has an advantage over HiFT by 1.1, 2.1%, demonstrating the effectiveness of the HFFT module, and superior robustness capacity compared to other prevalent trackers.

Evaluation on VOT2020: We also test SiamHFFT on the VOT2020 benchmark against HCAT, LightTrack (Yan et al., 2021b), ATOM and DiMP. VOT2020 consists of 60 videos with mask annotations and adopts the expected average overlap (EAO) as the metric for evaluating the performance of the trackers, which is calculated by:

$$\bar{\phi}N_S = \frac{1}{N_S}\sum\nolimits_{i=1}^{N_S}\phi N_S \tag{15}$$

TABLE 2 Overall evaluation on UAV123@10fps.

	SiamFC	AutoTrack	TADT	MCCT	SiamRPN++	DeepSTRCF	CCOT	ECO	HiFT	SiamHFFT
Prec.	67.8	67.6	68.4	68.1	74.0	68.0	70.4	70.9	75.4	76.5
Succ.	47.2	48.1	50.7	49.2	55.5	49.9	50.2	51.9	57.4	59.6

The best performance are shown in red.

TABLE 3 Evaluation on VOT2020.

	HCAT	LightTrack	ATOM	DiMP	SiamHFFT
EAO	0.276	0.242	0.271	0.274	0.231
Accuracy	0.455	0.422	0.462	0.457	0.459
Robustness	0.747	0.689	0.734	0.740	0.646

The best performance are shown in red.

where N_S denotes the length of the video sequences, ϕN_S denotes the average accuracy of a video sequence whose length is N_S. Finally, the EAO value can be obtained by calculating the average value of the video sequences of N_S length.

The experimental results are presented in Table 3. Our tracker achieves an EAO value of 0.231, robustness of 0.646, and accuracy of 0.459. The performance of SiamHFFT is comparable to that of the state-of-the-art models for each criterion.

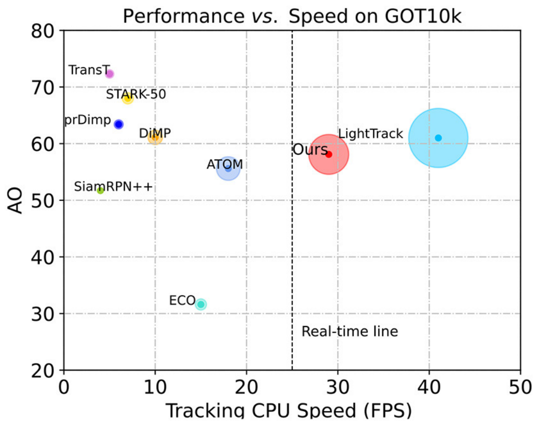

FIGURE 7
Speed and performance comparisions on GOT-10k. The horizontal axis represents model speed on a CPU and the vertical axis represents the AO score.

Speed, FLOPs and params

To verify the efficiency of our tracker, we conducted a set of experiments on the GOT-10k benchmark, which is a large-scale tracking dataset consisting of more than 10,000 videos, covering a wide range of 560 types of common moving objects. Following the test protocols of GOT-10k, all of the evaluated trackers are trained with the same training data, and are tested with the same test data. We evaluated the performance of SiamHFFT against TransT, STARK, DiMP, SiamRPN++, ECO, ATOM, and LightTrack. Our SiamHFFT is conducted on PC while the data of other trackers on GOT-10k is obtained from Chen et al. (2022b). Both average overlap (AO) and speed were considered to evaluate the performance of the trackers. We visualize the AO performance with respect to the frames-per-seconds (FPS) tracking speed. The comparision results are presented in Figure 7. Each tracker is represented by a circle, and the radius of the circle r is calculated as follows:

$$r = k\frac{speed/Average(speed)}{AO} \tag{16}$$

where k denotes a scale factor, we set k=10. The higher value of r indicates the better performance. All trackers were tested on CPU platform, and real-time line (26 fps) performance is represented by a dotted line to measure the real-time capacity of the trackers, trackers locate on the right side of the line are considered to achieve the real-time performance. According to Figure 7, only SiamHFFT and LightTrack can meet the real-time requirement on the CPU. Among these comparision trackers, TransT utilized a modified ResNet50 as backbone and a transformer-based network to obtain discriminative features, and achieve the highest AO score, but it sacrifices the speed which runs a low speed on CPU. Similarly, STARK, DiMP, prDiMP, SiamRPN++ can only obtain satisfactory AO scores at the expense of speed. The correlation filter-based tracker, ECO, also adopts the deep features which does not achieve a satisfactory speed on CPU. Our tracker exhibits an average speed of 28 FPS on the CPU, not only reach the real-time requirement, but the area of the circle representing our method is the second large of all the trackers.

To validate the lightness of our model, we compared the floating-point operations (FLOPs) and Params of the model with STARK-S50 and SiamRPN++. FLOPs represent the theoretical

TABLE 4 Comparision about the FLOPs and params.

Trackers	FLOPs (G)	Params (M)
STARK-S50	10.5	23.3
SiamRPN++	48.9	54
SiamHFFT	0.6	4.4

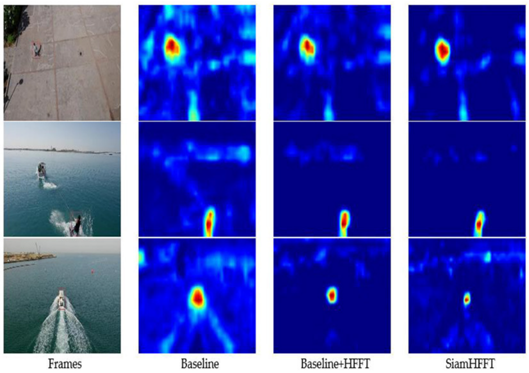

FIGURE 8
Visualization of the confidence maps of three trackers on several sequences from the UAV123 dataset. The response visualization results are an intuitive reflection of tracker performance.

calculation volume of the model, which means the number of calculations required for the inference. Params refer to the amount of the parameters in the model, which directly determines the size of the model and also directly affects the memory consumption when a model making inferences. The comparison results are illustrated in Table 4. It is worth note that our SiamHFFT tracker achieve a promising result over other trackers. The FLOPs and Parameters are 16× and 5× less than those of STARK-S50. This shows that our method can use fewer parameters and lower memory consumption, making it possible for deployments in the edge hardware environments.

Ablation studies

This section presents ablation studies conducted to verify the effectiveness of our framework. We selected several challenging frames from the UAV123 dataset and visualized the tracking results using heatmaps, as shown in Figure 8. The first column presents the given target which is highlighted with a red box, and the remaining columns present the visualized results of the predicted target prior to the current frame.

The second column presents the visualization results of the baseline, which only adopts ShuffleNetV2 as backbone with the reshaping module and the prediction head. The response area of the baseline is much larger than the original target size and has obscure edges affected by distractors in the frames.

TABLE 5 Experimental results on UAV 123 benchmark with different backbones.

	Baseline	Baseline+HFFT	SiamHFFT
AlexNet	73.6	77.2	78.9
ShuffleNetV2	74.1	81.6	82.8

The third column presents the visualization results of the baseline with the HFFT module. Compared with the baseline alone, the response area is smaller and clearer because the HFFT module enhances the critical semantic and spatial features of the target, enabling the model to generate more discriminative response maps. With the HFFT module, our tracker achieves significant improvement in tracking accuracy, which validates the effectiveness of the HFFT module for handling small objects.

The last column presents the response map generated by the proposed SiamHFFT, which adopts the entire operation module, backbone, reshaping module, HFFT module and the SAM, where the classification and regression head are utilized to estimate the location of a target. According to the visualization results of the response maps, our SiamHFFT model has clear advantages over other modified versions. The response areas are more precise and discriminative relative to the distractors.

We also test the performance on UAV123 benchmark with different backbones, we use the accuracy score to measure the performance variation. Experimental result is shown in Table 5, we choose two lightweight networks, AlexNet and ShuffleNetV2, to make a comparision. Similar to Figure 8, the effectiveness of the HFFT module is measured in a quantitative manner. The model adopts ShuffleNetV2 as backbone has better performance on all of the three criteria. The experiment results of Table 4 also demonstrate the effectiveness of the HFFT module.

Conclusion

In this paper, an HFFT tracking method based on a Siamese network was proposed. To integrate and optimize multi-level features, we designed a novel feature fusion transformer that can reinforce semantic information and spatial details during the tracking process. Additionally, based on our lightweight backbone, excessive computation for feature extraction is avoided, which accelerates object tracking speed. To validate the effectiveness of our trackers, extensive experiments were conducted on five benchmarks. Our method achieves excellent results on small target datasets such as UVA123 and UAV123@10fps, and also shows good performance on genetic public visual tracking datasets, such as LaSOT, VOT2020, and GOT-10k. Our method can potentially inspire further research on small object tracking, particularly for UAV tracking.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author/s.

Author contributions

Conceptualization, methodology, software, validation, formal analysis, data curation, and writing—original draft preparation: JD. Investigation: SW. Resources, writing—review and editing, supervision, and funding acquisition: YC. Visualization: YF. All authors have read and agreed to the published version of the manuscript.

Funding

YC is the leader of the funding for the research of National Science Foundation of China (No. 11975066), he received his Ph.D. degree from Jilin University, in 2002. He is a Professor with Jilin University, and also with Dalian University of Technology. His research interests include CMOS image sensor and digital signal processing of images.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.1082346/full#supplementary-material

References

Beal, J., Kim, E., Tzeng, E., Park, D. H., Zhai, A., and Kislyuk, D. J. (2020). Toward transformer-based object detection. *arXiv [Preprint]*. doi: 10.48550/arXiv.2012.09958

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016). "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 850–865. doi: 10.1007/978-3-319-48881-3_56

Bhat, G., Danelljan, M., Gool, L. V., and Timofte, R. (2019). "Learning discriminative model prediction for tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 6182–6191. doi: 10.1109/ICCV.2019.00628

Cao, Z., Fu, C., Ye, J., Li, B., and Li, Y. (2021). "HiFT: hierarchical feature transformer for aerial tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 15457–15466. doi: 10.1109/ICCV48922.2021.01517

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). "End-to-end object detection with transformers," in *European Conference on Computer Vision* (New York, NY: Springer), 213–229. doi: 10.1007/978-3-030-58452-8_13

Chen, B., Li, P., Bai, L., Qiao, L., Shen, Q., Li, B., et al. (2022). Backbone is all your need: a simplified architecture for visual object tracking. *arXiv preprint arXiv:2203.05328*. doi: 10.1007/978-3-031-20047-2_22

Chen, C. -F. R., Fan, Q., and Panda, R. (2021). "Crossvit: cross-attention multi-scale vision transformer for image classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 357–366. doi: 10.1109/ICCV48922.2021.00041

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., et al. (2021). Decision transformer: reinforcement learning *via* sequence modeling. *Adv. Neural Inform. Process. Syst.* 34, 15084–15097.

Chen, X., Wang, D., Li, D., and Lu, H. J. (2022b). Efficient visual tracking via hierarchical cross-attention transformer. *arXiv [Preprint]*. doi: 10.48550/arXiv.2203.13537

Chen, X., Yan, B., Zhu, J., Wang, D., and Lu, H. J. (2022a). High-performance transformer tracking. *arXiv preprint arXiv:2203.13533* (New Orleans, LA: IEEE). doi: 10.1109/CVPR46437.2021.0080

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. (2021). "Transformer tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE), 8126–8135. doi: 10.1109/CVPR46437.2021.00803

Chen, Z., Zhong, B., Li, G., Zhang, S., and Ji, R. (2020). "Siamese box adaptive network for visual tracking", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6668–6677. doi: 10.1109/CVPR42600.2020.00670

Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M. (2019). "Atom: accurate tracking by overlap maximization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4660–4669. doi: 10.1109/CVPR.2019.00479

Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. (2017). "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 6638–6646. doi: 10.1109/CVPR.2017.733

Danelljan, M., Robinson, A., Shahbaz Khan, F., and Felsberg, M. (2016). "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 472–488. doi: 10.1007/978-3-319-46454-1_29

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. J. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv [Preprint]*. doi: 10.48550/arXiv.1810.04805

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16 × 16 words: transformers for image recognition at scale. *arXiv [Preprint]*. doi: 10.48550/arXiv.2010.11929

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., et al. (2019). "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 5374–5383. doi: 10.1109/CVPR.2019.00552

Fan, H., and Ling, H. (2019). "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition*, 7952–7961. doi: 10.1109/CVPR.2019.00814

Fan, H., and Ling, H. J. (2020). Cract: cascaded regression-align-classification for robust visual tracking. *arXiv preprint arXiv:2011.12483*. doi: 10.1109/IROS51168.2021.9636803

Guo, D., Wang, J., Cui, Y., Wang, Z., and Chen, S. (2020). "SiamCAR: siamese fully convolutional classification and regression for visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 6269–6277. doi: 10.1109/CVPR42600.2020.00630

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2022). A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* 1. doi: 10.1109/TPAMI.2022.3152247

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV: IEEE), 770–778. doi: 10.1109/CVPR.2016.90

He, X., Chen, Y., and Lin, Z. J. R. S. (2021). Spatial-spectral transformer for hyperspectral image classification. *Remote Sens*. 13, 498. doi: 10.3390/rs13030498

Hou, R., Ma, B., Chang, H., Gu, X., Shan, S., Chen, X. J. I. T., et al. (2020). IAUnet: global context-aware feature learning for person reidentification. *IEEE Trans Neural Netw Learn Syst*. 32, 4460–4474. doi: 10.1109/TNNLS.2020.3017939

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv [Preprint]*. doi: 10.48550/arXiv.1704.04861

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7132–7141. doi: 10.1109/CVPR.2018.00745

Huang, L., Zhao, X., Huang, K. J., and Intelligence, M. (2019). Got-10k: a large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 1562–1577. doi: 10.1109/TPAMI.2019.2957464

Javed, S., Danelljan, M., Khan, F. S., Khan, M. H., Felsberg, M., and Matas, J. (2021). Visual object tracking with discriminative filters and siamese networks: a survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* 1–20. doi: 10.1109/TPAMI.2022.3212594

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kämäräinen, J.-K., et al. (2020). "The eighth visual object tracking VOT2020 challenge results," in *European Conference on Computer Vision* (New York, NY: Springer), 547–601.

Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J. (2019). "Siamrpn++: evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 4282–4291. doi: 10.1109/CVPR.2019.00441

Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018). "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 8971–8980. doi: 10.1109/CVPR.2018.00935

Li, F., Tian, C., Zuo, W., Zhang, L., and Yang, M.-H. (2018). "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4904–4913. doi: 10.1109/CVPR.2018.00515

Li, X., Ma, C., Wu, B., He, Z., and Yang, M.-H. (2019). "Target-aware deep tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA: IEEE), 1369–1378. doi: 10.1109/CVPR.2019.00146

Li, Y., Fu, C., Ding, F., Huang, Z., and Lu, G. (2020). "AutoTrack: towards high-performance visual tracking for UAV with automatic spatio-temporal regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 11923–11932. doi: 10.1109/CVPR42600.2020.01194

Lin, L., Fan, H., Xu, Y., and Ling, H. J. (2021). Swintrack: a simple and strong baseline for transformer tracking. *arXiv [Preprint]*. doi: 10.48550/arXiv.2112.00995

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 2117–2125. doi: 10.1109/CVPR.2017.106

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: common objects in context," in *European Conference on Computer Vision* (New York, NY: Springer), 740–755. doi: 10.1007/978-3-319-10602-1_48

Lin, Z., Feng, M., Santos, C. N., Yu, M., Xiang, B., Zhou, B., et al. (2017). A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, L., Hamilton, W., Long, G., Jiang, J., and Larochelle, H. J. (2020). A universal representation transformer layer for few-shot image classification. *arXiv preprint arXiv:2006.11702*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). "Swin transformer: hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 10012–10022. doi: 10.1109/ICCV48922.2021.00986

Ma, N., Zhang, X., Zheng, H.-T., and Sun, J. (2018). "Shufflenet v2: practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 116–131. doi: 10.1007/978-3-030-01264-9_8

Marvasti-Zadeh, S. M., Cheng, L., Ghanei-Yakhdan, H., and Kasaei, S. (2021). Deep learning for visual tracking: a comprehensive survey. *IEEE Trans. Intell. Transp. Syst.* 23, 3943–3968. doi: 10.1109/TITS.2020.3046478

Mayer, C., Danelljan, M., Bhat, G., Paul, M., Paudel, D. P., Yu, F., et al. (2022). "Transforming model prediction for tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (New Orleans, LA: IEEE), 8731–8740. doi: 10.1109/CVPR52688.2022.00853

Mueller, M., Smith, N., and Ghanem, B. (2016). "A benchmark and simulator for uav tracking," in *European Conference on Computer Vision* New York, NY: Springer), 445–461. doi: 10.1007/978-3-319-46448-0_27

Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B. (2018). "Trackingnet: a large-scale dataset and benchmark for object tracking in the wild," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 300–317.

Nguyen, V.-Q., Suganuma, M., and Okatani, T. (2020). "Efficient attention mechanism for visual dialog that can handle all the interactions between multiple inputs," in *European Conference on Computer Vision* (New York, NY: Springer), 223–240. doi: 10.1007/978-3-030-58586-0_14

Ning, X., Duan, P., Li, W., and Zhang, S. J. I. S. P. L. (2020). Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer. *IEEE Signal Process. Lett.* 27, 1944–1948. doi: 10.1109/LSP.2020.3032277

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., et al. (2020). "Stabilizing transformers for reinforcement learning," in *International Conference on Machine Learning*: *PMLR* (Vienna: ACM), 7487–7498.

Paulus, R., Xiong, C., and Socher, R. J. (2017). A deep reinforced model for abstractive summarization. *arXiv [Preprint]*. doi: 10.48550/arXiv.1705.04304

Qingyun, F., Dapeng, H., and Zhaokui, W. J. (2021). Cross-modality fusion transformer for multispectral object detection. *arXiv [Preprint]*. doi: 10.48550/arXiv.2111.00273

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4510–4520. doi: 10.1109/CVPR.2018.00474

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Adv. Neural Inform. Process. Syst*. 30, 6000–6010.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI: IEEE), 3156–3164. doi: 10.1109/CVPR.2017.683

Wang, N., Zhou, W., Tian, Q., Hong, R., Wang, M., and Li, H. (2018). "Multi-cue correlation filters for robust visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 4844–4853. doi: 10.1109/CVPR.2018.00509

Wang, X., Girshick, R., Gupta, A., and He, K. (2018). "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT: IEEE), 7794–7803. doi: 10.1109/CVPR.2018.00813

Wolfe, J. M., and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nat. Rev. Neurosci.* 5, 495–501. doi: 10.1038/nrn1411

Xu, Y., Wang, Z., Li, Z., Yuan, Y., and Yu, G. (2020). "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proceedings of the AAAI Conference on Artificial Intelligence* (New York, NY: AAAI), 12549–12556. doi: 10.1609/aaai.v34i07.6944

Yan, B., Peng, H., Fu, J., Wang, D., and Lu, H. (2021a). "Learning spatio-temporal transformer for visual tracking," in *Proceedings of the IEEE/CVF*

International Conference on Computer Vision (Montreal, QC: IEEE), 10448–10457. doi: 10.1109/ICCV48922.2021.01028

Yan, B., Peng, H., Wu, K., Wang, D., Fu, J., and Lu, H. (2021b). "LightTrack: finding lightweight neural networks for object tracking *via* one-shot architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE), 15180–15189. doi: 10.1109/CVPR46437.2021.01493

Yan, B., Zhao, H., Wang, D., Lu, H., and Yang, X. (2019). "'Skimming-perusal'tracking: a framework for real-time and robust long-term tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Seoul: IEEE), 2385–2393. doi: 10.1109/ICCV.2019.00247

Yu, B., Tang, M., Zheng, L., Zhu, G., Wang, J., Feng, H., et al. (2021). "High-performance discriminative tracking with transformers," in: *Proceedings of the IEEE/CVF International Conference on Computer Vision* (Montreal, QC: IEEE), 9856–9865. doi: 10.1109/ICCV48922.2021.00971

Yu, Y., Xiong, Y., Huang, W., and Scott, M. R. (2020). "Deformable siamese attention networks for visual object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 6728–6737. doi: 10.1109/CVPR42600.2020.00676

Zhang, G., Zhang, P., Qi, J., and Lu, H. (2021). "Hat: hierarchical aggregation transformers for person re-identification," in *Proceedings of the 29th ACM International Conference on Multimedia*, 516–525. doi: 10.1145/3474085.3475202

Zhang, J., Huang, B., Ye, Z., Kuang, L.-D., and Ning, X. J. S. R. (2021). Siamese anchor-free object tracking with multiscale spatial attentions. *Sci. Rep*. 11, 1–14. doi: 10.1038/s41598-021-02095-4

Zhang, W. J. M. S., and Processing, S. (2021). A robust lateral tracking control strategy for autonomous driving vehicles. *Mech. Syst. Signal Process*. 150, 107238. doi: 10.1016/j.ymssp.2020.107238

Zhang, Z., Lan, C., Zeng, W., Jin, X., and Chen, Z. (2020a). "Relation-aware global attention for person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Seattle, WA: IEEE), 3186–3195. doi: 10.1109/CVPR42600.2020.00325

Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. (2020b). "Ocean: object-aware anchor-free tracking," in *European Conference on Computer Vision* (New York, NY: Springer), 771–787. doi: 10.1007/978-3-030-58589-1_46

Zhao, M., Okada, K., and Inaba, M. J. (2021). Trtr: visual tracking with transformer. *arXiv [Preprint]*. doi: 10.48550/arXiv.2105.03817

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., and Hu, W. (2018). "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich: Springer), 101–117. doi: 10.1007/978-3-030-01240-3_7