Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.039489

ARTICLE

Siamese Dense Pixel-Level Fusion Network for Real-Time UAV Tracking

Zhenyu Huang[1,2], Gun Li[2], Xudong Sun[1], Yong Chen[1], Jie Sun[1], Zhangsong Ni[1,*] and Yang Yang[1,*]

[1]Chengdu Fluid Dynamics Innovation Center, Chengdu, 610031, China

[2]School of Aeronautics and Astronautics, University of Electronic Science and Technology of China, Chengdu, 611731, China

*Corresponding Authors: Zhangsong Ni. Email: nzscczx@163.com; Yang Yang. Email: yy_doc@163.com

Received: 01 February 2023 Accepted: 29 May 2023 Published: 08 October 2023

ABSTRACT

Onboard visual object tracking in unmanned aerial vehicles (UAVs) has attracted much interest due to its versatility. Meanwhile, due to high precision, Siamese networks are becoming hot spots in visual object tracking. However, most Siamese trackers fail to balance the tracking accuracy and time within onboard limited computational resources of UAVs. To meet the tracking precision and real-time requirements, this paper proposes a Siamese dense pixel-level network for UAV object tracking named SiamDPL. Specifically, the Siamese network extracts features of the search region and the template region through a parameter-shared backbone network, then performs correlation matching to obtain the candidate region with high similarity. To improve the matching effect of template and search features, this paper designs a dense pixel-level feature fusion module to enhance the matching ability by pixel-wise correlation and enrich the feature diversity by dense connection. An attention module composed of self-attention and channel attention is introduced to learn global context information and selectively emphasize the target feature region in the spatial and channel dimensions. In addition, a target localization module is designed to improve target location accuracy. Compared with other advanced trackers, experiments on two public benchmarks, which are UAV123@10fps and UAV20L from the unmanned air vehicle123 (UAV123) dataset, show that SiamDPL can achieve superior performance and low complexity with a running speed of 100.1 fps on NVIDIA TITAN RTX.

KEYWORDS

Siamese network; UAV object tracking; dense pixel-level feature fusion; attention module; target localization

1 Introduction

UAVs can be effectively used to construct object-tracking applications because of their versatility, small volume, and straightforward operation [1]. Generally speaking, object tracking in UAVs has long been a great interest in computer vision [2]. Given an initial target region in the first frame, the object tracker aims to continuously predict the target location and generate a bounding box to fit the target in subsequent frames. Despite the considerable achievements in visual object tracking, various realistic challenges still exist in UAV tracking.

- Due to the limited onboard computational power of a UAV, the deployed models need to be carefully designed to limit their computing consumption.
- Complex views in aerial photography make distinguishing the foreground and background challenging.

Excellent tracking algorithms have increased in recent years, which are primarily divided into two groups: discriminative correlation filter (DCF)-based methods [3] and deep learning (DL)-based methods [4]. The DCF-based trackers were frequently used for aerial object tracking due to their practical computation [1]. However, a key issue is that DCF-based trackers cannot satisfy the demands of UAV tracking in complex dynamic environments due to their hand-crafted features and sophisticated optimization strategies. The DL-based trackers have demonstrated outstanding performance in UAV tracking with the representation of deep features. However, these trackers require numerous calculations. As a result, the main challenge faced by many researchers is to keep a balance between accuracy and efficiency.

With the gradual development of deep learning, Siamese-based trackers have performed exceptionally well in visual object tracking, such as Fully-Convolutional Siamese (SiamFC) [5], Siamese Region Proposal Network (Siamese-RPN) [6] and SiamRPN++ [7]. The parameter-sharing Siamese backbone is responsible for extracting deep features from both the template and search regions. Subsequently, the cross-correlation operation obtains the response map containing similarities between the two areas. Almost all Siamese trackers implement similarity matching with a simple convolution operation. The main weakness of this operation is that the matching area is larger than the target area, leading to massive noises from the background [8], especially in small object scenarios of UAV tracking. Moreover, the matching area containing interference will blur the object boundary and produce an inaccurate bounding box.

Siamese trackers aim to encode the local information in a sliding window without global context information. In recent years, the attention mechanism has achieved significant improvements in DL [9,10], allowing models to filter out meaningful features from the global context of the whole image. By enhancing the representation power, the attention mechanism also helps improve tracking accuracy in UAV tracking challenges, such as fast motion and small objects.

This paper proposes a dense pixel-level feature fusion module by adopting pixel-wise correlation to enhance the matching ability and then resist background interference. The attention module, including self-attention and channel attention, is introduced to enrich the target representation with solid robustness to distractors and complex backgrounds. Moreover, referring to the structure of LightTrack [11], the target localization module is carefully designed to strengthen the target discriminability.

The main contributions of this work are summarized as follows:

- A dense pixel-level feature fusion module is designed to improve the correlation matching between template features and search features, which helps alleviate background interference.
- An attention module is introduced to aggregate and recalibrate the single feature in spatial and channel dimensions. It refines features effectively and boosts the representation power.
- A target localization module composed of classification and regression branches is designed to produce a more precise position.
- The proposed SiamDPL tracker has been evaluated on two challenging UAV tracking benchmarks, demonstrating its effectiveness and efficiency in precision and time consumption. Moreover, SiamDPL performs well in partial occlusion, viewpoint variation, and fast motion.

The rest of this article is organized as follows. Section 2 briefly introduces the related work on the DCF-based tracker, Siamese tracker, and UAV object tracking. In Section 3, this paper describes the designed modules, including the dense pixel-level feature fusion network, the attention module, and the target localization module. The performance analysis is carried out in Section 4, and the conclusion is presented in Section 5.

2 Related Work

This section briefly reviews the DCF-based trackers and details the Siamese trackers and the object tracking algorithms based on UAVs.

2.1 DCF-Based Tracker

The DCF-based trackers, such as Kernelized Correlation Filter (KCF) [12], Efficient Convolution Operators (ECO) [13], and Aberrance Repressed Correlation Filter (ARCF) [14], aim to classify and score the search samples by minimizing the loss between the labels and the cyclic correlation between instances and filters [15]. Most DCF-based trackers have been deployed widely on UAVs due to their expansibility and efficiency [16]. However, the hand-crafted features limit the representation ability of such trackers, while introducing the deep learning network reduces the speed of inference.

2.2 Siamese Tracker

By training offline, Siamese trackers are more prominent and stable in performance. SiamFC [5] used a fully-convolutional Siamese architecture to calculate the similarities between the template and search regions. Applying Region Proposal Network (RPN), SiamRPN (short for Siamese-RPN) [6] formulated the tracking problem as a local one-shot detection challenge, furthermore enriching the structure of trackers for UAV tracking. Distractor-aware Siamese Region Proposal Networks (DaSiamRPN) [17] enlarged the training dataset and customized multiple data augmentations to introduce negative semantic pairings. Furthermore, it designed a distractor-aware module to counter semantic distractors. However, the tracking algorithms used shallow networks to extract features from limited semantic information. SiamDW [18] developed four strategies to design deeper backbone networks to obtain richer semantic information. Apart from SiamDW, SiamRPN++ [7] proposed a spatial aware sampling strategy to relieve the restriction of translation invariance. In addition, the depth-wise correlation was employed to decrease computational costs for a stable training process. SiamMask [19] combined tracking and segmentation to locate the target with a rotative mask. Some trackers adopted the anchor-free methods to avoid false positive samples in the anchor-based methods [20–23]. Fully Convolutional Siamese tracker++ (SiamFC++) [21] directly predicted the confidence score of target existence without predefined anchor boxes. At the same time, SiamFC++ applied the quality evaluation branch independently in classification. Siamese Box Adaptive Network (SiamBAN) [22] adopted the box adaptive head for classification and regression with more minor output variables. Object-aware Anchor-free Networks (Ocean) [23] proposed the anchor-free method and the feature alignment module to correct the inaccurate bounding box. Although most of the above trackers are highly robust, they must be simplified to meet the real-time demand in UAV tracking. LightTrack [11] adopted neural architecture search (NAS) to design a lighter yet more efficient tracker considering limited computational resources. Compared with hand-crafted architectures, the network structure trained by the NAS method was superior, with a unique network design. Therefore, this paper designed a target localization module based on the configuration of LightTrack to get a more vital discriminative ability.

A large and growing body of literature has investigated the attention mechanism, which dynamically conducts recalibration by allocating each input a separate weight. Residual Attentional Siamese Network (RASNet) [24] adopted general attention, residual attention, and channel attention to recalibrate the features of the template branch, while it was a restricted sample strategy. Deformable Siamese Attention Networks (SiamAttn) [25] employed a deformable Siamese attention module (DSA) to enhance the spatial and channel information of the template and search features and implicitly update the template features. Despite the excellent performance, SiamAttn used the attention module for feature extraction, which increased the computational consumption and reduced the speed. This paper used the attention module for the single feature after feature fusion, improving efficiency.

2.3 UAV Object Tracking

Many previous types of research on UAV tracking have focused on DCF-based trackers. AutoTrack [26] proposed an automatic and adaptive learning method to adjust the spatiotemporal regularization online, which was robust to complex and varied UAV scenarios. Bidirectional Incongruity-aware Correlation Filter (BiCF) [27] effectively learned object appearance variation by integrating the bidirectional inconsistency error during the UAV tracking process.

Real-time performance is an essential requirement for UAV object tracking. Recent papers have discussed the impact of real-time performance on systems and algorithms [28–31]. With the emergence of lightweight onboard Graphic Processing Units (GPU), such as NVIDIA Jetson AGX Xavier, it is becoming increasingly more work to ignore the existence of Siamese trackers. Siamese Anchor Proposal Network (SiamAPN) [32] proposed an anchor proposal network to display excellent performance with high speed, satisfying the real-time needs of UAV tracking, avoiding numerous predefined anchors, and acquiring better accuracy through refinement. Based on the attentional aggregation network (AAN), SiamAPN++ [33] utilized self-AAN and cross-AAN to enrich feature representation. In addition, the anchor proposal network based on dual features (APN-DF) was introduced to increase the robustness of proposing anchors [1]. However, the correlation matching used by these trackers introduced much noise. This work realized the dense pixel-level feature fusion through pixel-wise correlation to achieve precise matching.

The Transformer algorithms have also developed rapidly in recent years and have been used in UAV object tracking [34]. Hierarchical Feature Transformer (HiFT) [35] used a feature transformer to aggregate multi-layer information to raise the global contextual information. It captured the space information of the object with an encoder and the semantic information with a decoder. Siamese Transformer Pyramid Network (SiamTPN) [36] used a transformer pyramid network (TPN) to integrate multi-layer features. In addition, a pooling attention (PA) layer was used to reduce memory and time complexity while improving robustness. Although they achieved excellent performance, they required a large amount of data for training and had many parameters, which affected the speed. This paper designed the dense pixel-level feature fusion module and combined it with the attention module and the target localization module to perform a high-speed and excellent tracker in UAV tracking with low computational complexity and parameters.

3 Proposed Method

The structure of the proposed Siamese dense pixel-level fusion network is shown in Fig. 1. The template and search images are inputted, and the Siamese architecture is adopted to extract features from both inputs with the same backbone. The dense pixel-level feature fusion module performs cross-correlation between template features and search features to obtain the response map. After that,

the attention module is embedded to emphasize key features of the response map in both spatial and channel dimensions, enhancing the self-semantic interdependencies of the response map. The target localization module learned from LightTrack [11] is designed to locate the object's position and determine the predicted boundary, including classification and regression branches.

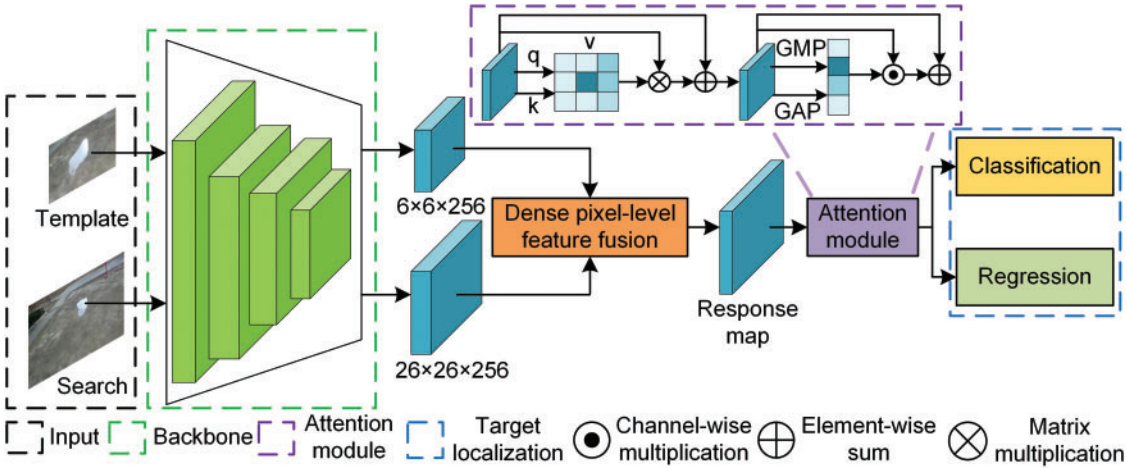

Figure 1: The structure of the proposed tracker. The tracker is composed of the feature extraction network (backbone), the dense pixel-level feature fusion, the attention module, and the target localization

3.1 Dense Pixel-Level Feature Fusion

The Siamese network treats visual tracking as a similarity-matching problem. As shown in Fig. 2a, several original Siamese trackers adopt naive correlation [5] for aggregation. The Siamese backbone (denoted as $f(\cdot)$) extracts the template image (marked as x) provided by the initial frame to obtain the template features $f(x) \in \mathbb{R}^{C\times H_x\times W_x}$. The search image (drawn as z) cropped from the current frame is extracted as the search features $f(z) \in \mathbb{R}^{C\times H_z\times W_z}$. The naive correlation uses the template features as a sliding window to perform cross-correlation calculation with the search features, as shown in Eq. (1):

$$f(x,z) = f(x) \star f(z) \tag{1}$$

where $\star$ refers to the cross-correlation operation. The response map $f(x,z) \in \mathbb{R}^{1\times H_o\times W_o}$ has the height $H_o = H_z - H_x + 1$ and the width $W_o = W_z - W_x + 1$. Considering the requirement for multi-dimensional output channels, the number of template feature channels needs to be increased. This operation is called up-channel cross-correlation (UP-Xcorr), used by SiamRPN [6], which brings the difficulty of training optimization and parameter imbalance between the template and search features.

As shown in Fig. 2b, SiamRPN++ [7] adopts a lightweight cross-correlation layer called depth-wise correlation. The template and search features perform the cross-correlation operation channel by channel, which is shown in Eq. (2):

$$f_i(x,z) = f_i(x) \star f_i(z),\ i \in [1,2,\ldots,C] \tag{2}$$

where $f_i(x,z) \in \mathbb{R}^{1\times H_o\times W_o}$, $f_i(x) \in \mathbb{R}^{1\times H_x\times W_x}$ and $f_i(z) \in \mathbb{R}^{1\times H_z\times W_z}$. The final response map $f(x,z) \in \mathbb{R}^{C\times H_o\times W_o}$ is obtained. The computational cost and memory consumption are significantly reduced by using depth-wise correlation. However, features in each channel are independent, resulting in the background noise.

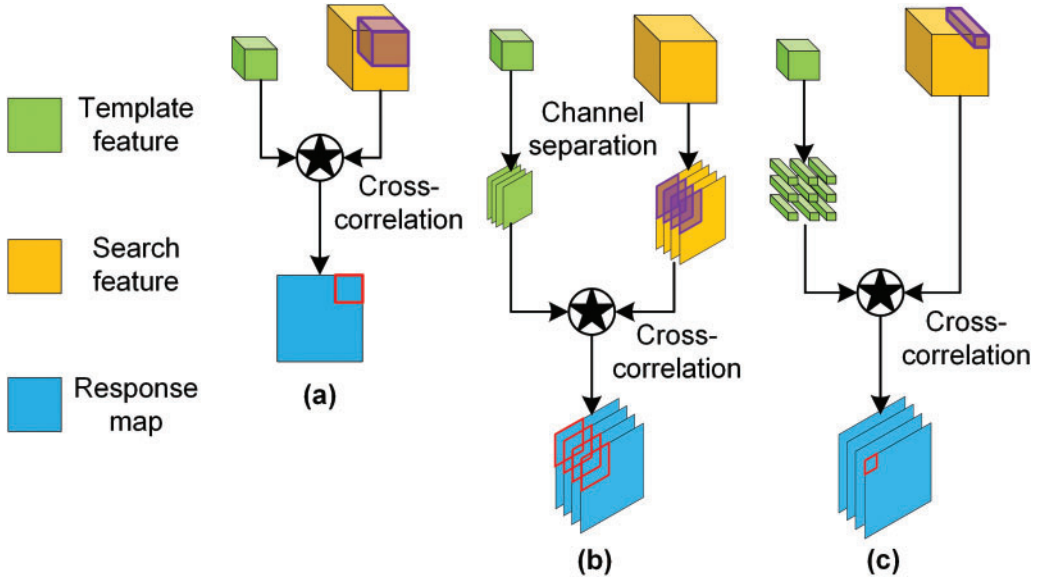

Figure 2: Three different cross-correlation methods. (a) naive correlation, (b) depth-wise correlation, (c) pixel-wise correlation

Pixel-wise correlation is introduced from Alpha-Refine [37] method to preserve more boundary information. As shown in Fig. 2c, $f(x) \in \mathbb{R}^{C\times H_x\times W_x}$ is decomposed into several small kernels $f_j(x) \in \mathbb{R}^{C\times 1\times 1}, j \in [1, 2, \ldots, H_x W_x]$. The cross-correlation between $f_j(x) \in \mathbb{R}^{C\times 1\times 1}$ and $f(z) \in \mathbb{R}^{C\times H_z\times W_z}$ is performed in Eq. (3), and the response map $f(x, z) \in \mathbb{R}^{H_x W_x\times H_z\times W_z}$ is obtained.

$$f_j(x, z) = f_j(x) \star f(z), j \in [1, 2, \ldots, H_x W_x] \tag{3}$$

The template features can be directly mapped to the search region to obtain an ideal matching area. Still, most trackers need to obtain the response map through cross-correlation and then map it to the search image to get the corresponding matching area [8]. Both naive and depth-wise correlations use the template features as a sliding window to calculate with the search features, which causes the receptive field to expand. It blurs spatial information and forms a larger corresponding matching area than the ideal one. In contrast, pixel-wise correlation divides the template features spatially into 1×1 small kernels. The response map is calculated by cross-correlation between kernels and the search features, which encodes the local region information to avoid a large correlation window from blurring the feature. Compared with naive correlation and depth-wise correlation, the response map of pixel-wise correlation has a larger size. A feature point in this response map corresponds to a smaller receptive field of the search region so that the corresponding matching area will be closer to the ideal matching area, resulting in less background information and avoiding spatial distortion.

In Fig. 3, the response maps of the three correlation methods are visualized. It can be seen from Fig. 3c that the naive correlation roughly represents the center location of the object without distinct shapes and scales, while the response map only has a single channel. In Fig. 3d, the depth-wise correlation encodes erroneous matching information in some channels of the response map. The pixel-wise correlation response map in Fig. 3e shows more boundary information of the object with better correlation matching ability.

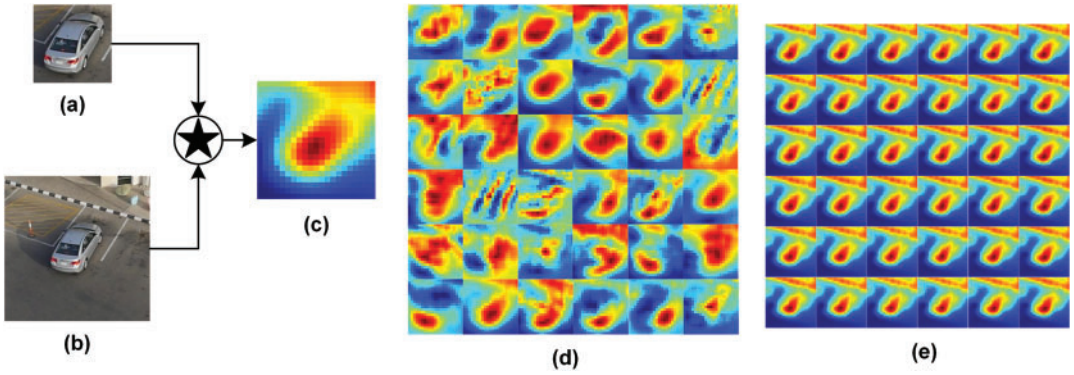

Figure 3: The visualization of response maps for naive correlation (c), depth-wise correlation (d), and pixel-wise correlation (e). Two inputs are 127 × 127 template image (a), and 287 × 287 search image (b)

Given the small size of the template features, the pixel-wise correlation response map needs to be expanded in the channel dimension. Therefore, as shown in Fig. 4, a dense pixel-level feature fusion module is designed by referring to Dense Convolutional Network (DenseNet) [38]. This module strengthens feature propagation and encourages feature reuse by applying the dense connectivity pattern. The pixel-wise correlation between the template and search features obtains the first response map. After concatenating with the search features, it is aggregated through a 1 × 1 convolution kernel to get the feature map, which must have the same channel number as the template features. The first feature map is then calculated with the template features by pixel-wise correlation to obtain the new response map with richer semantic information. The latest response map gets the last feature map through another 1 × 1 convolution kernel. Compared with the first feature map, the last feature map has fewer channels to reduce the computational cost. Only two dense connections are applied in this module to avoid excessive computational consumption caused by the dense connectivity pattern. With pixel-wise correlation, the fused feature map has the same size and receptive field as the search features. Its matching area is closer to the ideal matching area, avoiding spatial distortion.

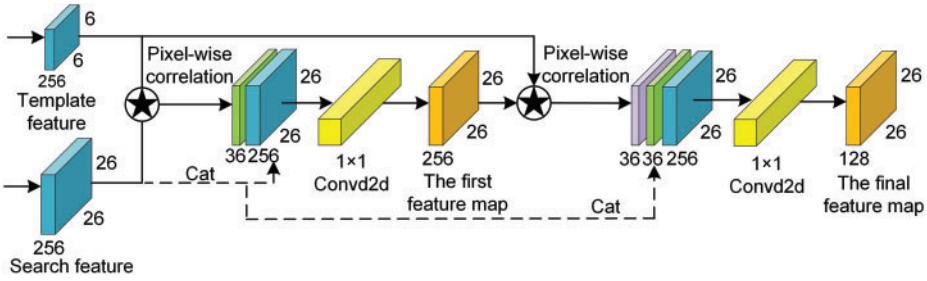

Figure 4: The dense pixel-level feature fusion module

3.2 Attention Module

Prior studies [24,25] have noted the importance of attention modules. The introduced attention module aims to strengthen the target features' representation power and enhance the feature map's self-semantic information. Compared with SiamAttn [25], the attention module employed for a single feature map consumes fewer computational resources.

Self-attention: Inspired by Dual Attention Network (DANet) [39], the self-attention module attends to spatial encoding. Limited by its intrinsic narrow receptive fields, a feature point can only be mapped to a small patch with the local context. Therefore, learning the global semantic connections from the entire feature map makes sense.

As shown in Fig. 5a, the feature map $F \in \mathbb{R}^{C\times H\times W}$ is utilized as the input to generate query features Q, key features K, and value features V respectively through three different 1×1 convolution kernels. Channels of $Q, K \in \mathbb{R}^{C/4\times H\times W}$ are compressed to decrease computational complexity, while $V \in \mathbb{R}^{C\times H\times W}$ preserves the number of channels. The three features are reshaped as $\hat{Q}^T \in \mathbb{R}^{HW\times C/4}$, $\hat{K} \in \mathbb{R}^{C/4\times HW}$, $\hat{V} \in \mathbb{R}^{C\times HW}$. Via matrix multiplication between $\hat{Q}^T$ and $\hat{K}$, the spatial self-attention map $\hat{M}_s$ is generated in Eq. (4) through a softmax operation on each row, which builds the relationship between pixel points and global context. $\hat{V}$ are multiplied with $\hat{M}_s$ to obtain the refined feature map $\hat{O}_p$, as follows in Eq. (5):

$$\hat{M}_s = Softmax\left(\hat{Q}^T\hat{K}\right) \in \mathbb{R}^{HW\times HW} \tag{4}$$

$$\hat{O}_p = \hat{V} \otimes \hat{M}_s \in \mathbb{R}^{C\times HW} \tag{5}$$

where $\otimes$ represents matrix multiplication. After reshaping $\hat{O}_p$ back to $O_p \in \mathbb{R}^{C\times H\times W}$, O_p is weighted by a 1×1 convolution kernel with a residual connection in Eq. (6):

$$O_n = \alpha O_p + F \in \mathbb{R}^{C\times H\times W} \tag{6}$$

where α is the weight factor given by the 1×1 convolution kernel. O_n is the output.

Channel attention: Unlike the detection or classification task, visual object tracking is independent of category recognition. The object class remains unchanged throughout the track. Each channel of features typically represents a specific object class, which can be adaptively enhanced. By introducing the channel attention inspired by Convolutional Block Attention Module (CBAM) [40], the interconnection can be applied between channels to improve the expression ability of specific semantics.

In Fig. 5b, the output $O_n \in \mathbb{R}^{C\times H\times W}$ of the self-attention module is taken as the input. To aggregate spatial information, global max pooling (GMP) and global average pooling (GAP) are employed to obtain two different spatial context descriptors, $GMP(O_n) \in \mathbb{R}^{C\times 1\times 1}$ and $GAP(O_n) \in \mathbb{R}^{C\times 1\times 1}$. The channel attention map M_c is produced by forwarding both descriptors to a shared network N, which consists of a 1×1 convolution kernel $W^{C/4}$ for adjusting the channel number to $C/4$, the activation function ReLU, and a 1×1 convolution kernel W^C for restoring the channel number to C, as shown in Eq. (7):

$$N = W^C(\mathrm{Re}LU(W^{C/4}(GMP(O_n); GAP(O_n)))) \tag{7}$$

After applying the shared network to the descriptors, M_c is merged in Eq. (8) using element-wise summation between $N_{Max} \in \mathbb{R}^{C\times 1\times 1}$ and $N_{Avg} \in \mathbb{R}^{C\times 1\times 1}$, and passing through a sigmoid function:

$$M_c = Sigmoid(N_{Max} + N_{Avg}) \in \mathbb{R}^{C\times 1\times 1} \tag{8}$$

M_c is channel-wise multiplicated with O_n, and adopts a residual connection, as follows in Eq. (9):

$$X = \beta \cdot O_n \odot M_c + O_n \in \mathbb{R}^{C\times H\times W} \tag{9}$$

where $\odot$ represents channel-wise multiplication. β is a scalar parameter, and X is the final refined output.

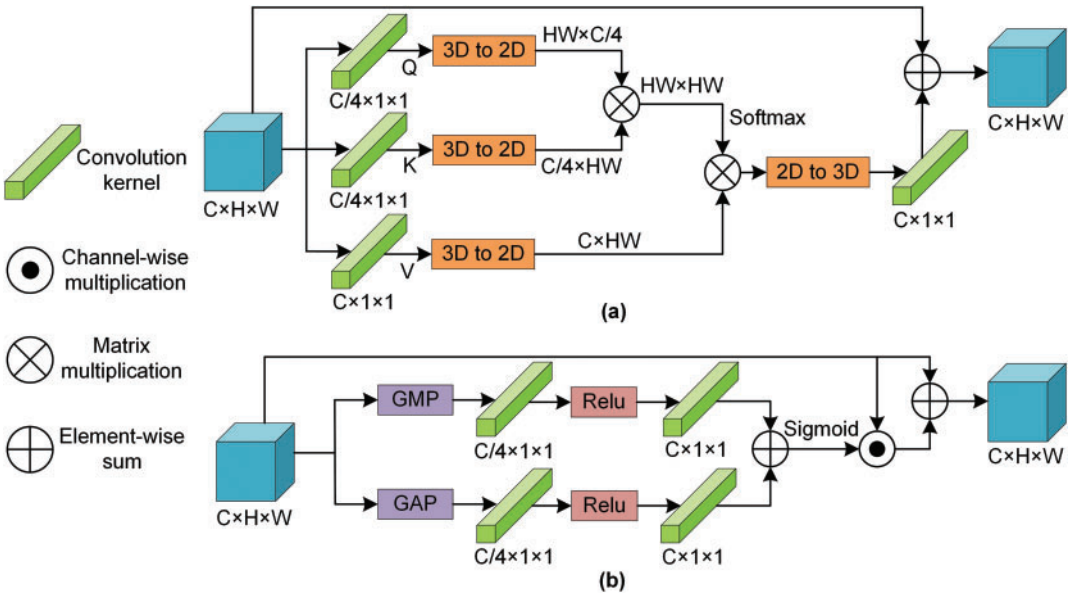

Figure 5: Attention modules include self-attention (a) and channel attention (b)

3.3 Target Localization

Divided into classification and regression branches, the RPN-style box head is still adopted in the target localization module. To design a sophisticated structure, the head of LightTrack [11] is referred to in the method.

From the LightTrack architecture searched by the one-shot NAS method, there are fewer layers in the classification branch compared with the layers in the regression branch. A possible explanation might be that the task of coarse target localization for the classification branch is more accessible than the task of precise bounding box prediction for the regression branch. Besides, the number of output channels for each layer in the classification branch is more significant than that in the regression branch because the classification of foreground and background requires more semantic information in the channel dimension. Following the same spirit, the target localization module is designed as shown in Fig. 6. Considering the speed of depthwise separable convolutions [41] employed in LightTrack on the GPU, regular convolutions are finally adopted. 1×1 convolution kernels adjust the channels at the end of the two branches, where k is the number of anchors. The classification branch outputs the confidence score of foreground and background. Thereby the number of output channels is $2k$. The regression branch outputs the distances dx, dy, dw and dh to refine the location and scale of each anchor, so the number of output channels is 4k.

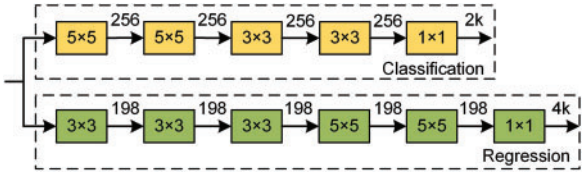

Figure 6: The target localization module is divided into the classification branch and the regression branch

4 Experiments

SiamDPL has been comprehensively validated through extensive experiments on two authoritative UAV tracking benchmarks, UAV20L [42] and UAV123@10fps [42]. In addition, 12 well-known trackers, including 6 DCF-based trackers (KCF [12], DSST [43], AutoTrack [26], Background-Aware Correlation Filters (BACF) [44], ECO_HC [13], and ARCF_H [14]) and 6 Siamese trackers (DaSiamRPN [17], Ocean [23], UpdateNet [45], SiamMask [19], SiamRPN++ [7], and SiamFC++ [21]), are also evaluated.

4.1 Experimental Details

The first five convolutional layers of AlexNet, which were pre-trained on ImageNet [46], are applied as the backbone. The entire tracker is fine-tuned on the training sets of COCO [47] and Youtube-Bounding Boxes [48] using several data augmentation strategies, such as translation, scale variations, illumination, and motion blur. The two datasets contain diverse categories of positive pairs to promote generalization and semantic negative pairs to improve discriminative ability [17]. The stochastic gradient descent (SGD) is applied with a momentum of 0.9 and a minibatch of 128. The size of the template image is 127×127, and the size of the search image is 287×287 in both the training and testing phases.

Following SiamRPN++ [7], a warm-up learning rate of 0.01 is used for the first five epochs, and the learning rate is increased to 0.03 in the 6th epoch. After that, the learning rate decays from 0.03 to 0.0005 in a proportional sequence. The backbone parameters are frozen for the first ten epochs, while the last three backbone layers are unfrozen and trained after ten epochs. The entire network is trained end-to-end, and each epoch has 100,000 sample pairs. This paper achieves optimal test performance in the 55th epoch after training for 60 epochs. Following SiamRPN [6], anchors are set with five aspect ratios, [0.33, 0.5, 1, 2, 3], and the anchor stride is 8. During the inference phase, cosine window penalty, aspect ratio penalty, and scale change penalty are applied in SiamDPL.

The method is implemented using Pytorch and SiamDPL, which is trained and tested on a personal computer (PC) with an Intel i9-9920X Central Processing Unit (CPU), 32 GB Random Access Memory (RAM), and NVIDIA TITAN RTX GPU.

4.2 Datasets and Evaluation Metrics

UAV20L: UAV20L is a sub-dataset of UAV123 [42] containing 20 long-term sequences taken by low-altitude UAVs, with a maximum of 5527 frames and an average of 2934 frames per sequence. Therefore, it serves as a verification of UAV long-term tracking scenes.

UAV123@10fps: UAV123@10fps is also a sub-dataset of UAV123 [42], consisting of 123 short-term sequences with a frame interval of 10 frames. Video sequences with a gap of more significant than 30 frames present challenging situations, such as fast movement and drastic object variation. The tracker evaluated on UAV123@10fps can be simulated in the UAV tracking scene with a low frame rate and extreme variation, which measures the effect of tracking speed on performance. Therefore, UAV123@10fps is adopted as a UAV low-speed tracking scene verification.

Evaluation metrics: The one-pass evaluation (OPE) metrics are adopted to evaluate the success rate and precision, using overlap score (OS) and center location error (CLE). OS is the intersection over union (IOU) score between the predicted bounding box and the ground-truth box. The success rate is measured by the percentage of frames whose OS surpasses a certain threshold. The success plot reflects the change in success rate with different thresholds, ranking all trackers by calculating the area

under the curve (AUC). CLE is the Euclidean distance between the center point of the ground-truth box and that of the predicted bounding box, while the precision plot shows the percentage of frames whose CLE is smaller than the threshold. Note that the precision at a threshold of 20 pixels is utilized for ranking in the relevant experiments.

4.3 Comparisons with Advanced Trackers

Compared with other advanced trackers on two benchmarks, SiamDPL has achieved good results. Although SiamRPN++ [7], SiamMask [19], and SiamFC++ [21] have higher precision and AUC scores, the enormous costs of computational resources and running speed are regrettable, which is verified in Section 4.5. Achieving accuracy and efficiency is only possible by carefully considering the real UAV tracking scene.

On UAV123@10fps: Fig. 7 shows the quantitative results of all trackers on the UAV123@10fps dataset. Specifically, SiamDPL achieves a precision of 0.697 and an AUC score of 0.497, surpassing DaSiamRPN [17] in both metrics. Although gaps still exist with the scores of SiamRPN++ [7], SiamFC++ [21], and SiamMask [19], they adopt deeper backbone networks such as ResNet50 and GoogleNet. Although these trackers have achieved a desirable level of tracking accuracy, they still need to be more computationally expensive. In Section 4.5, the speed and complexity of these Siamese trackers are compared with each other. SiamDPL has achieved the fastest speed and the lowest computational complexity, making it more suitable for UAV tracking.

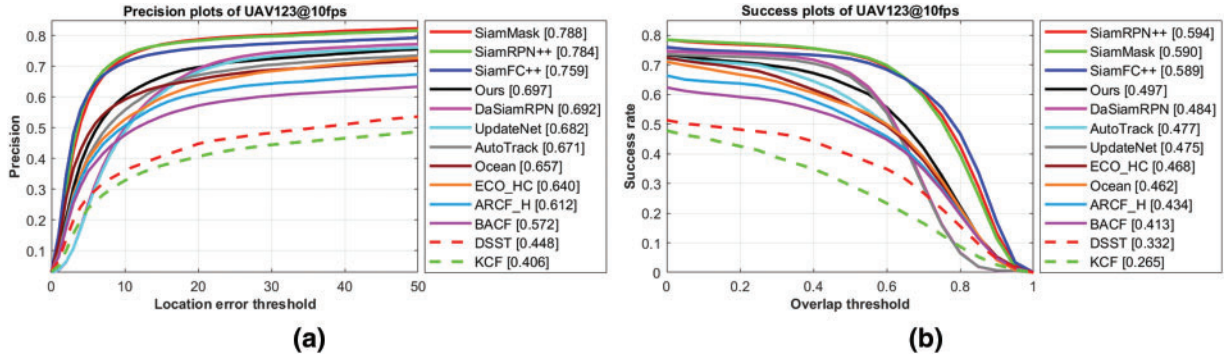

Figure 7: Precision plot (a) and success plot (b) on the UAV123@10fps

Table 1 shows the attribute-based evaluation results on the UAV123@10fps dataset to analyze performance in various challenges. There are four common attributes in UAV tracking challenges, including Similar Object (SO), Fast Motion (FM), Partial Occlusion (PO), and Scale Variation (SV). Although SiamDPL ranks fourth in these attributes, it outperforms other trackers regarding speed, as shown in Section 4.5.

On UAV20L: As shown in Fig. 8, SiamDPL outperforms most trackers on the UAV20L dataset with a precision of 0.723 and an AUC score of 0.521. Considering that the speed of SiamDPL is fast enough to meet the real-time requirements of UAV tracking, the evaluated results of UAV20L are more convincing than those of UAV123@10fps. Compared with DaSiamRPN [17], SiamDPL has a significant improvement of 9.2% on the precision and 7.9% on the AUC score, ranking third only to SiamRPN++ [7] and SiamFC++ [21].

Table 1: Evaluate SiamDPL and other 12 advanced trackers on the UAV123@10fps dataset about four challenge attributes, and the best four performances are responsively highlighted that ranked first in red, second in green, third in blue, and fourth in orange

Tracker	SO		FM		PO		SV	
	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
KCF	0.453	0.279	0.217	0.145	0.344	0.223	0.374	0.238
DSST	0.509	0.368	0.277	0.187	0.384	0.276	0.424	0.313
ECO_HC	0.655	0.478	0.483	0.330	0.572	0.404	0.594	0.430
ARCF_H	0.657	0.445	0.384	0.256	0.531	0.361	0.570	0.399
AutoTrack	0.664	0.462	0.525	0.349	0.584	0.405	0.629	0.443
BACF	0.605	0.424	0.407	0.275	0.467	0.327	0.525	0.374
UpdateNet	0.677	0.447	0.533	0.379	0.602	0.406	0.644	0.451
Ocean	0.682	0.480	0.405	0.253	0.586	0.406	0.631	0.437
DaSiamRPN	0.710	0.471	0.537	0.380	0.618	0.419	0.655	0.460
SiamRPN++	0.760	0.563	0.646	0.480	0.717	0.520	0.757	0.571
SiamFC++	0.727	0.544	0.694	0.527	0.680	0.502	0.737	0.571
SiamMask	0.731	0.544	0.682	0.498	0.708	0.507	0.761	0.569
Ours	0.720	0.486	0.607	0.401	0.619	0.422	0.664	0.468

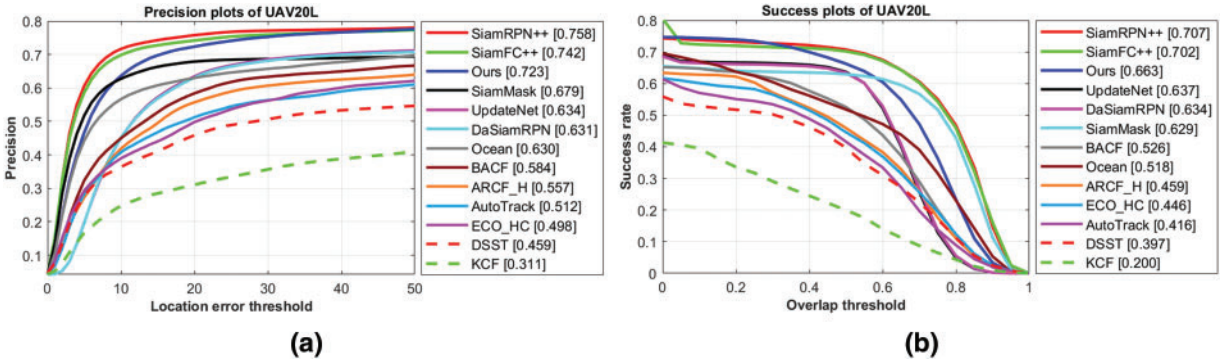

Figure 8: Precision plot (a) and success plot (b) on the UAV20L

To analyze the robustness of trackers in long-term tracking challenges, Table 2 shows the performances of each tracker in five challenging attributes of the UAV20L dataset, including Background Clutter (BC), Full Occlusion (FO), Fast Motion (FM), Scale Variation (SV) and Viewpoint Change (VC). SiamDPL performs exceptionally well in the precision of FM and VC and ranks third in BC, FO, and SV. The top two trackers (SiamFC++ [21] and SiamRPN++ [7]) consume more computational power, as shown in Section 4.5, while SiamDPL maintains robustness and real-time performance in long-term UAV tracking.

Table 2: Experiments are conducted with SiamDPL and 12 other advanced trackers on the UAV20L dataset about five challenge attributes. The best four performances are responsively highlighted that ranked first in red, second in green, third in blue, and fourth in orange

	BC		FO		FM		SV		VC	
Tracker	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
KCF	0.251	0.148	0.264	0.115	0.188	0.064	0.275	0.175	0.189	0.148
DSST	0.333	0.218	0.354	0.189	0.280	0.147	0.430	0.320	0.346	0.271
ECO_HC	0.235	0.133	0.339	0.163	0.359	0.219	0.471	0.360	0.423	0.353
ARCF_H	0.329	0.210	0.378	0.199	0.354	0.201	0.534	0.368	0.465	0.334
AutoTrack	0.374	0.219	0.403	0.198	0.419	0.234	0.487	0.330	0.420	0.303
BACF	0.329	0.209	0.378	0.200	0.408	0.234	0.562	0.399	0.500	0.373
UpdateNet	0.387	0.206	0.411	0.207	0.615	0.370	0.615	0.432	0.533	0.421
Ocean	0.407	0.268	0.484	0.278	0.616	0.395	0.611	0.430	0.577	0.418
DaSiamRPN	0.387	0.208	0.411	0.209	0.615	0.371	0.611	0.432	0.527	0.418
SiamRPN++	0.562	0.365	0.553	0.342	0.726	0.514	0.745	0.572	0.687	0.554
SiamFC++	0.571	0.384	0.574	0.370	0.665	0.483	0.729	0.569	0.656	0.550
SiamMask	0.319	0.182	0.363	0.200	0.678	0.459	0.662	0.503	0.656	0.519
Ours	0.495	0.296	0.539	0.299	0.706	0.449	0.709	0.510	0.667	0.500

4.4 Ablation Study

To demonstrate the effectiveness of the proposed dense pixel-level feature fusion module, the attention module, and the target localization module, ablation studies are conducted on the UAV20L dataset. As shown in Fig. 9, the baseline tracker is designed as an anchor-based tracker whose backbone is AlexNet. It adopts depth-wise correlation for feature fusion. The number of output channels in the classification branch is 10, while the number of output channels in the regression branch is 20.

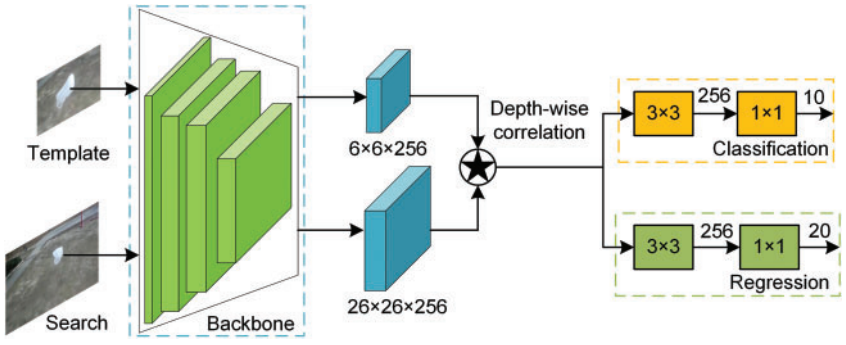

Figure 9: The network of the baseline tracker. It composes of the classification and regression, the feature extraction network (backbone), and depth-wise correlation

As shown in Table 3, CR stands for the designed classification and regression branches of the target localization module, AM represents the attention module, and the dense pixel-level feature

fusion module is referred to as DPFF. The baseline tracker achieves a precision of 0.594 and an AUC score of 0.388. Firstly adding the target localization module increases the precision from 0.594 to 0.626 and the AUC score from 0.388 to 0.431, showing that the classification and regression branches can predict a more accurate bounding box. Adding the attention module increases the precision from 0.626 to 0.647, bringing more precise predictions for the object center. With the dense pixel-level feature fusion contribution, SiamDPL achieves the best results. The dense pixel-level feature fusion enhances the matching ability and reduces the introduction of noise.

Table 3: Comparison of precision and AUC score of trackers using different components on the UAV20L dataset

Structure	Precision	AUC score
Baseline	0.594	0.388
Baseline + CR	0.626	0.431
Baseline + CR + AM	0.647	0.428
Baseline + CR + AM + DPFF (Ours)	0.724	0.521

Taken together, these results suggest that the proposed feature fusion, the attention module, and the target localization module can improve tracking performance in long-term tracking scenarios, and their cooperation has brought a positive promotion.

4.5 Speed and Complexity

All trackers are tested on NVIDIA TITAN RTX to evaluate their speed and complexity. Params represent the model's parameters to measure the space complexity of trackers. Frame Per Second (FPS) indicates the number of images that can be processed per second, which is adopted to evaluate the speed of trackers. Multiply–Accumulate Operations (MACs) are common steps that compute the product of two numbers and add that product to an accumulator. They can be used to measure the computational complexity of trackers because the convolutional neural network (CNN)-based trackers are dominated by convolution operations, which include multiplication and addition operations, and much hardware treats multiplication and addition operations as a single instruction.

Table 4 summarizes the runtime speed and complexity of Siamese trackers. It can be observed that SiamDPL has a minimum of MACs of 9.521 G and Params of 9.386 M, computing much more efficiently than other Siamese trackers. SiamRPN++ [7], SiamMask [19], and SiamFC++ [21] cannot run at real-time speed despite their high precision and superior AUC scores. With the closest speed to SiamDPL, DaSiamRPN [17] has lower precision and AUC score than SiamDPL. The data reported here support SiamDPL to be deployed and applied in resource-constrained applications so that SiamDPL can satisfy the real-time requirements of UAV tracking.

To analyze the consumption of computational resources for each module, the components of SiamDPL are compared in terms of computational complexity and parameters. It is apparent from Table 5 that the backbone incurs the most MACs, despite being one of the most miniature modules for the backbone structure. Numerous parameters are employed for classification and regression for more precise target localization. The template features constitute a small part of the parameters in the feature fusion. The objects that participate in calculating the attention module come from the feature map itself, so there are very few parameters in the attention module.

Table 4: Comparison of the speed, MACs and Params of all Siamese trackers

Trackers	Speed	MACs	Params
SiamRPN++	40 fps	62.324 G	53.951 M
SiamMask	45.3 fps	21.341 G	18.817 M
SiamFC++	41.9 fps	17.521 G	12.706 M
UpdateNet	61 fps	24.225 G	20.036 M
Ocean	49.8 fps	26.087 G	25.869 M
DaSiamRPN	87 fps	24.210 G	19.642 M
Ours	100.1 fps	9.597 G	9.498 M

Table 5: Comparison of MACs and Params of each employed module

Module	MACs	Params
Backbone	5.715 G	3.750 M
Feature Fusion	79.173 M	117.12 K
Self-Attention	212.686 M	98.7 K
Channel Attention	238.592 K	32.8 K
Target localization	3.784 G	5.598 M

Table 6 illustrates the computation of the three cross-correlation methods. The MACs of the pixel-wise correlation are the highest because it establishes relationships between each pixel of the template and search features. The naive correlation and the depth-wise correlation are applied by taking the template features as a sliding window to calculate with the search features. Several pixels at the center of the template features are not involved in the calculation with pixels of the search feature edges, reducing the computational complexity. However, due to the larger matching area, more background noise is introduced to weaken the correlation-matching ability.

Table 6: MACs and output sizes of three cross-correlation operations

Feature fusion	MACs	Output size (channel × width × height)
Naive correlation	4.064 M	$1 \times 21 \times 21$
Depth-wise correlation	4.064 M	$256 \times 21 \times 21$
Pixel-wise correlation	6.230 M	$36 \times 26 \times 26$

To compare the speed and performance of each tracker more intuitively, the PAS, which is the mean of precision and AUC score, is introduced to measure the tracking performance. As shown in Table 7, compared with the top three trackers on UAV123@10fps, SiamDPL has at least 7.7% lower PAS but is 54.8 fps faster. The performance gap between SiamDPL and the top two trackers becomes smaller on UAV20L, as shown in Table 7. The PAS of SiamDPL is only 4.56% lower than that of SiamRPN++ [7] and 3.65% lower than that of SiamFC++ [21]. Therefore, SiamDPL can achieve outstanding tracking results while maintaining the fastest speed.

Table 7: PAS and speed of trackers on the UAV123@10fps and UAV20L datasets

Trackers	Speed	PAS	
		UAV123@10fps	UAV20L
SiamRPN++	40 fps	0.689	0.6685
SiamMask	45.3 fps	0.689	0.5965
SiamFC++	41.9 fps	0.674	0.6585
UpdateNet	61 fps	0.5785	0.538
Ocean	49.8 fps	0.5595	0.537
DaSiamRPN	87 fps	0.588	0.5365
Ours	100.1 fps	0.597	0.622

4.6 Qualitative Evaluation

Some qualitative comparisons among UpdateNet [45], DaSiamRPN [17], Ocean [23], and SiamDPL are shown in Fig. 10. SiamDPL can maintain stable tracking on sequences group1, group2 of UAV20L and person7_2, person19_2 of UAV123@10fps during similar object, occlusion, and fast motion. Owing to the contributions of the dense pixel-level feature fusion, the attention module, and the target localization module, as seen in Fig. 10, SiamDPL eventually achieves significant results in UAV tracking.

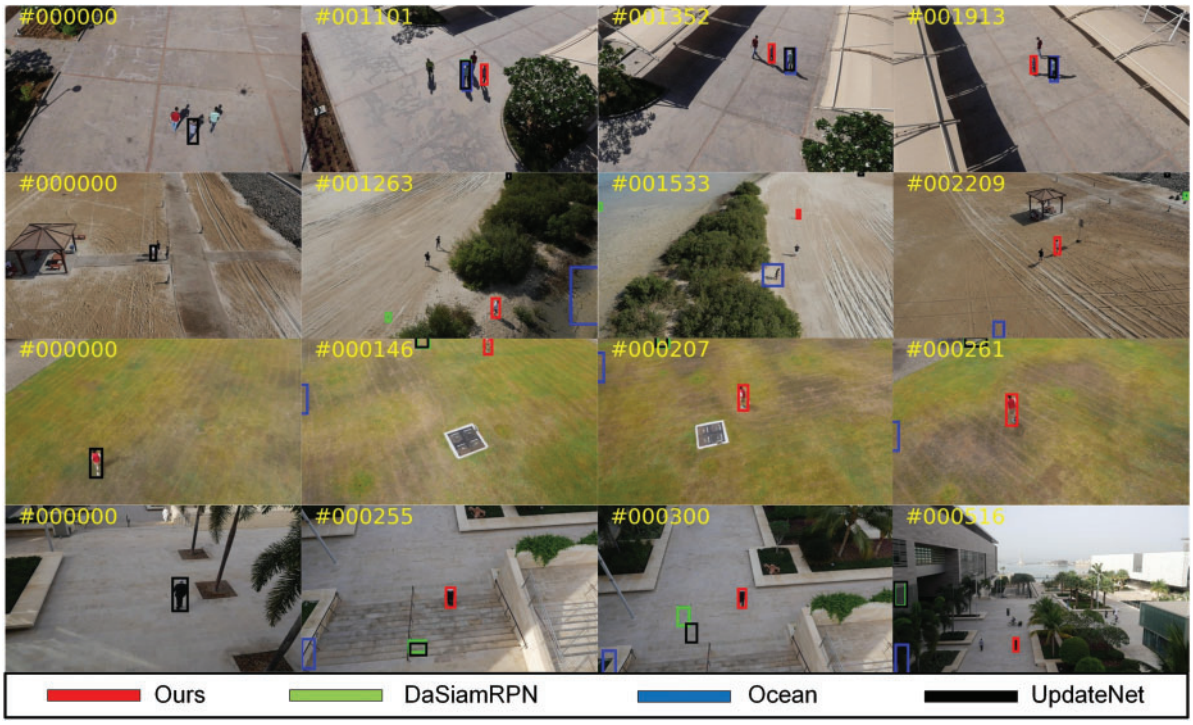

Figure 10: Screenshots of group1, group2 from UAV20L, person7_2 and person19_2 from UAV123@10fps

5 Conclusion

This paper proposed a Siamese dense pixel-level fusion network to fulfill the performance and efficiency requirements of real-time UAV tracking. The dense pixel-level feature fusion was proposed to filter out the background noise with the help of pixel-wise correlation and to enrich features through the dense connection. The attention module, consisting of self-attention and channel attention, was introduced to aggregate global information from the feature map and enhance the representation power, improving the robustness against complex backgrounds and distractors. The target localization module was designed to obtain more accurate bounding boxes. Finally, compared with several advanced trackers, SiamDPL was evaluated on two common benchmarks, demonstrating excellent performance with the lowest complexity and fastest speed for real-time UAV tracking.

Acknowledgement: The authors would like to thank editors and reviewers for their valuable work.

Funding Statement: This research was funded by the National Natural Science Foundation of China (Grant No. 52072408), author Y. C.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Gun Li, Yang Yang; data collection: Jie Sun, Xudong Sun; analysis and interpretation of results: Zhangsong Ni, Yong Chen; draft manuscript preparation: Zhengyu Huang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data and Materials in this study can be obtained from the corresponding author: Yang Yang.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Fu, K. Lu, G. Zheng, J. Ye, Z. Cao *et al.,* "Siamese object tracking for unmanned aerial vehicle: A review and comprehensive analysis," arXiv Preprint arXiv:2205.04281, 2022.

[2] L. Jiang, Y. Zheng, X. Cheng and B. Jeon, "Dynamic temporal–spatial regularization-based channel weight correlation filter for aerial object tracking," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[3] B. Li, C. Fu, F. Ding, J. Ye and F. Lin, "ADTrack: Target-aware dual filter learning for real-time anti-dark UAV tracking," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, pp. 496–502, 2021.

[4] D. Held, S. Thrun and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 749–765, 2016.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. Torr, "Fully-convolutional Siamese networks for object tracking," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 850–865, 2016.

[6] B. Li, J. Yan, W. Wu, Z. Zhu and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 8971–8980, 2018.

[7] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing *et al.,* "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc, of the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 4282–4291, 2019.

[8] B. Liao, C. Wang, Y. Wang, Y. Wang and J. Yin, "PG-Net: Pixel to global matching network for visual tracking," in *European Conf. on Computer Vision*, Glasgow, UK, pp. 429–444, 2020.
[9] Z. Wang, J. Oin, X. Xiang, Y. Tan and N. N. Xiong, "Criss-cross attentional Siamese networks for object tracking," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 2931–2946, 2022.
[10] M. H. Guo, T. X. Xu, J. J. Liu, Z. N. Liu, P. T. Jiang *et al.,* "Attention mechanisms in computer vision: A survey," *Computational Visual Media*, vol. 8, no. 3, pp. 331–368, 2022.
[11] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu *et al.,* "LightTrack: Finding lightweight neural networks for object tracking via one-shot architecture search," in *2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 15180–15189, 2021.
[12] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2014.
[13] M. Danelljan, G. Bhat, F. Shahbaz Khan and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 6638–6646, 2017.
[14] Z. Huang, C. Fu, Y. Li, F. Lin and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Seoul, Korea (South), pp. 2891–2900, 2019.
[15] C. Fu, B. Li, F. Ding, F. Lin and G. Lu, "Correlation filters for unmanned aerial vehicle-based aerial tracking: A review and experimental evaluation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 125–160, 2021.
[16] S. Liu, D. Liu, G. Srivastava, D. Połap and M. Woźniak, "Overview and methods of correlation filter algorithms in object tracking," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1895–1917, 2021.
[17] Z. Zhu, Q. Wang, B. Li, W. Wu and J. Yan, "Distractor-aware Siamese networks for visual object tracking," in *Proc. European Conf. on Computer Vision*, Salt Lake City, Utah, USA, pp. 101–117, 2018.
[18] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 4591–4600, 2019.
[19] Q. Wang, L. Zhang, L. Bertinetto, W. M. Hu and P. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, California, USA, pp. 1328–1338, 2019.
[20] S. Guo, Y. Li, X. Chen and Y. Zhang, "Anchor-free Siamese network based on visual tracking," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 3137–3148, 2022.
[21] Y. Xu, Z. Wang, Z. Li, Y. Yuan and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," *in Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 34, no. 7, pp. 12549–12556, 2020.
[22] Z. Chen, B. Zhong, G. Li, S. Zhang and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 6668–6677, 2020.
[23] Z. Zhang, H. Peng, J. Fu, B. Li and W. Hu, "Ocean: Object-aware anchor-free tracking," in *European Conf. on Computer Vision*, Glasgow, UK, pp. 771–787, 2020.
[24] Q. Wang, Z. Teng, J. Xing, J. Gao and W. Hu, "Learning attentions: Residual attentional Siamese network for high performance online visual tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 4854–4863, 2018.
[25] Y. Yu, Y. Xiong, W. Huang and R. S. Matthew, "Deformable siamese attention networks for visual object tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 6728–6737, 2020.
[26] 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. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 11923–11932, 2020.

[27] F. Lin, C. Fu, Y. He, F. Guo and Q. Tang, "BiCF: Learning bidirectional incongruity-aware correlation filter for efficient UAV object tracking," in *2020 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Paris, France, pp. 2365–2371, 2020.
[28] N. Min-Allah, M. B. Qureshi, S. Alrashed and O. F. Rana, "Cost efficient resource allocation for real-time tasks in embedded systems," *Sustainable Cities and Society*, vol. 48, pp. 101523, 2019.
[29] N. Min-Allah, S. U. Khan, X. Wang and A. Y. Zomaya, "Lowest priority first based feasibility analysis of real-time systems," *Journal of Parallel and Distributed Computing*, vol. 73, no. 8, pp. 1066–1075, 2013.
[30] Y. Lyu, L. Chen, C. Zhang, D. Ou, N. Min-Allah *et al.,* "An interleaved depth-first search method for the linear optimization problem with disjunctive constraints," *Journal of Global Optimization*, vol. 70, pp. 737–756, 2018.
[31] P. Lindberg, J. Leingang, D. Lysaker, K. Bilal, S. U. Khan *et al.,* "Comparison and analysis of greedy energy-efficient scheduling algorithms for computational grids," in *Energy-Efficient Distributed Computing Systems*. New York, NY, USA: Wiley, pp. 189–214, 2012.
[32] C. Fu, Z. Cao, Y. Li, J. Ye and C. Feng, "Siamese anchor proposal network for high-speed aerial tracking," in *2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Xi'an, China, pp. 510–516, 2021.
[33] Z. Cao, C. Fu, J. Ye, B. Li and Y. Li, "SiamAPN++: Siamese attentional aggregation network for real-time uav tracking," in *2021 IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic, pp. 3086–3092, 2021.
[34] J. Thangavel, T. Kokul, A. Ramanan and S. Fernando, "Transformers in single object tracking: An experimental survey," arXiv Preprint arXiv:2302.11867, 2023.
[35] Z. Cao, C. Fu, J. Ye, B. Li and Y. Li, "Hift: Hierarchical feature transformer for aerial tracking," in *2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 15457–15466, 2021.
[36] D. Xing, N. Evangeliou, A. Tsoukalas and A. Tzes, "Siamese transformer pyramid networks for real-time UAV tracking," in *2022 IEEE/CVF Winter Conf. on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, pp. 2139–2148, 2022.
[37] B. Yan, X. Zhang, D. Wang, H. Lu and X. Yang, "Alpha-refine: Boosting tracking performance by precise bounding box estimation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, USA, pp. 5289–5298, 2021.
[38] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 4700–4708, 2017.
[39] J. Fu, J. Liu, Y. Li, Y. Bao and H. Tian, "Dual attention network for scene segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, pp. 3146–3154, 2018.
[40] S. Woo, J. Park, J. Y. Lee and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proc. of the European Conf. on Computer Vision*, Munich, Germany, pp. 3–19, 2018.
[41] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 1251–1258, 2017.
[42] M. Mueller, N. Smith and B. Ghanem, "A benchmark and simulator for uav tracking," in *European Conf. on Computer Vision*, Amsterdam, The Netherlands, pp. 445–461, 2016.
[43] M. Danelljan, G. Häger, F. Khan and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. of the British Machine Vision Conf. (BMVC)*, Nottingham, UK, pp. 1–14, 2014.
[44] H. K. Galoogahi, A. Fagg and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 1135–1143, 2017.
[45] L. Zhang, A. Gonzalez-Garcia, J. Weijer, M. Danelljan and F. S. Khan, "Learning the model update for siamese trackers," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Seoul, Korea (South), pp. 4010–4019, 2019.
[46] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li *et al.* "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, FL, USA, pp. 248–255, 2009.

[47] T. Y. Lin, M. Maire, S. Belongie, J. Hays and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conf. on Computer Vision*, Zurich, Switzerland, pp. 740–755, 2014.

[48] E. Real, J. Shlens, S. Mazzocchi, X. Pan and V. Vanhoucke, "Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 5296–5305, 2017.