drones

MDPI

Article

Learning Template-Constraint Real-Time Siamese Tracker for Drone AI Devices via Concatenation

Zhewei Wu, Qihe Liu *, Shijie Zhou, Shilin Qiu, Zhun Zhang and Yi Zeng

School of Information and Software Engineering, University of Electronic Science and Technology of China, No. 4, Section 2 Jianshebei Road, Chengdu 610054, China; wuzhewei0914@foxmail.com (Z.W.); sjzhou@uestc.edu.cn (S.Z.); qiushilin@std.uestc.edu.cn (S.Q.); zhunzhang@std.uestc.edu.cn (Z.Z.); zengyiwilliam@foxmail.com (Y.Z.)
* Correspondence: qiheliu@uestc.edu.cn

Abstract: Significant progress has been made in object tracking tasks thanks to the application of deep learning. However, current deep neural network-based object tracking methods often rely on stacking sub-modules and introducing complex structures to improve tracking accuracy. Unfortunately, these approaches are inefficient and limit the feasibility of deploying efficient trackers on drone AI devices. To address these challenges, this paper introduces ConcatTrk, a high-speed object tracking method designed specifically for drone AI devices. ConcatTrk utilizes a lightweight network architecture, enabling real-time tracking on edge devices. Specifically, the proposed method primarily uses the concatenation operation to construct its core tracking steps, including multi-scale feature fusion, intra-frame feature matching, and dynamic template updating, which aim to reduce the computational overhead of the tracker. To ensure tracking performance in UAV tracking scenarios, ConcatTrk implements a learnable feature matching operator along with a simple and efficient template constraint branch, which enables accurate tracking by discriminatively matching features and incorporating periodic template updates. Results of comprehensive experiments on popular benchmarks, including UAV123, OTB100, and LaSOT, show that ConcatTrk has achieved promising accuracy and attained a tracking speed of 41 FPS on an edge AI device, Nvidia AGX Xavier. ConcatTrk runs 8× faster than the SOTA tracker TransT while using 4.9× fewer FLOPs. Real-world tests on the drone platform have strongly validated its practicability, including real-time tracking speed, reliable accuracy, and low power consumption.

Keywords: object tracking; UAV tracking; edge AI devices

check for updates

Citation: Wu, Z.; Liu, Q.; Zhou, S.; Qiu, S.; Zhang, Z.; Zeng, Y. Learning Template-Constraint Real-Time Siamese Tracker for Drone AI Devices via Concatenation. *Drones* **2023**, *7*, 592. https://doi.org/10.3390/drones7090592

Academic Editors: Wenzheng Xu, Tang Liu and Weifa Liang

Received: 23 August 2023
Revised: 14 September 2023
Accepted: 18 September 2023
Published: 20 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Visual object tracking aims to continuously estimate the location or trajectory of a specific object of interest in a video sequence. It plays a pivotal role in environmental perception and finds widespread application in domains such as unmanned aerial vehicles (UAVs) [1–8], robotics [9–11], autonomous driving [12,13], and various related areas. Consequently, visual object tracking has emerged as a highly investigated research field, drawing substantial attention from the academic community. With the development of deep learning, significant progress has been made in improving the accuracy and robustness of object tracking methods, enabling them to effectively handle various categories of target appearances and environmental changes. However, the complexity due to the multilayer architecture and high computational cost makes it challenging to deploy efficient visual object tracking methods on drone AI devices with limited power supply and computational resources.

Recently, deep-learning-based trackers [14–20] have demonstrated the advantages of tracking efficiency. Nevertheless, trackers employing stacked submodules [16,21] and complex structures [19,20,22] prove to be inefficient for edge AI devices, leading to increased power consumption. In addition, widely employed cross-correlation methods [14,15,23]

exhibit limitations in terms of tracking accuracy and robustness, which makes it challenging to fulfill the requirements of UAV tracking. Finally, UAV tracking poses challenges due to high-speed motion, including target feature changes and blurring, which emphasize the importance of efficient tracking and capturing of target feature changes. Unfortunately, most current methods only consider feature matching within a single frame and do not account for dynamic adjustments to the tracking template. Although some methods introduce dynamic template updating [24–26], they often fail to ensure efficient tracking. In summary, the urgent problem that needs to be addressed is how to achieve reliable tracking accuracy with low power consumption and computational cost.

To address these challenges, in this paper, we present a lightweight Siamese tracker, termed ConcatTrk. The primary advantage of ConcatTrk is its ability to perform real-time tracking on drone AI devices without significant accuracy reduction, which is not achievable with extant tracking networks. The network contains three key components: the multi-scale feature fusion module, the learnable feature matching module, and the template-constraint branch. The multi-scale feature fusion module is utilized to concatenate and reduce the dimensionality of feature maps from different convolution levels, which aims to obtain a fused feature vector with multi-scale features characterizing the target. Additionally, the learnable feature matching module takes dynamically updated target feature vectors and search region features as inputs and produces a response map with attention via a concatenation operation, enabling single-frame feature matching within the module. Furthermore, the template-constraint branch periodically updates high-quality target features into the tracking features by concatenation, thus enhancing the tracking model's accuracy with minimal computational overhead.

The construction of ConcatTrk highlights the utilization of concatenation as the fundamental step for achieving multi-scale feature fusion, intra-frame feature matching, and dynamic template updating. Our motivation for adopting this approach is twofold. First, it aims to reduce the computational overhead in the feature fusion process. Second, it strives to overcome the limitation associated with commonly used cross-correlation methods. This limitation pertains to their inability to effectively learn the sample distribution from the training data. As a result, ConcatTrk is able to retain robust performance in complex scenarios, as shown in Figure 1.

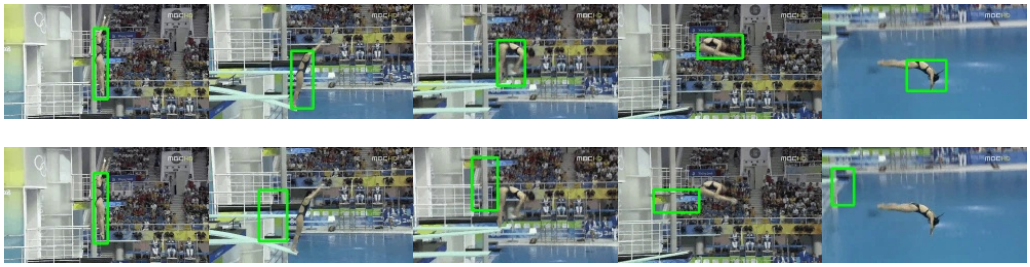


Figure 1. Visualized comparison of the effect of template-constraint branch (*Diving* from OTB100 dataset). The upper and lower rows show the tracking results with and without the template-constraint branch, respectively. Benefiting from the dynamic capture of feature changes, ConcatTrk is able to track accurately when the target appearance changes radically.

To evaluate the performance of our proposed ConcatTrk, extensive experiments are conducted on three benchmark datasets, namely UAV123 [27], OTB100 [28], and LaSOT [29]. Our tracker demonstrated competitive accuracy on all the datasets while maintaining a real-time tracking speed of 41 FPS on the edge AI device Nvidia AGX Xavier.

The main contributions of this work are as follows:

1. We transfer and improve a learnable feature matching module, which performs the feature matching task more discriminatively than the non-parametric cross-correlation method.

2. We propose a simple and effective template-constraint branch for dynamically capturing feature changes of a target and set up a filtering strategy to prevent invalid features from contaminating the tracking template.
3. We design a lightweight tracker, ConcatTrk, with an end-to-end and cost-effective structure that performs a balance between tracking speed and accuracy on three benchmarks.
4. We deploy and evaluate ConcatTrk on a drone platform under real-world conditions, showing strong tracking capabilities in challenging scenarios as well as low power consumption.

2. Related Work

2.1. Siamese Tracking

The Siamese neural network (SNN) is a type of architecture consisting of two artificial neural networks. These networks share weights and take in two samples as input, outputting their representations embedded in a high-dimensional space for comparing their similarities. SNNs have become popular in object tracking due to their ability to embed two samples in the same feature space. SNN structure is simple yet flexible, and researchers have made significant improvements and achieved remarkable progress on this architecture. These advancements include methods such as multi-scale feature fusion and feature matching.

Multi-scale feature fusion is a commonly used approach for extracting target features in object tracking, utilized in methods such as SiamRPN++ [16], SiamCAR [30], C-RPN [21], PGNet [31], etc. It benefits from intermediate features with final features as the representation of input images and obtains different levels of target features through multi-scale feature fusion operations. This approach leverages the small perceptual field and strong geometric detail feature representation of low-level networks. For instance, SiamRPN++ [16] extracts multi-level intermediate features from the backbone network and feeds them into three separate SiamRPN [15] modules for independent position prediction. The predictions from different levels are then aggregated through a fusion operation. Similarly, SiamCAR [30] employs a similar feature extraction strategy and performs multi-scale position prediction through cross-correlation operations. TCNN [32] adopts a tree-based structure to organize multiple distinct convolutional networks and performs online updates to represent different target appearance features separately. However, UAV tracking scenarios, the computational overhead introduced by stacking sub-modules, and repeated cross-correlation operations can be problematic for edge devices with limited computational resources.

In terms of feature matching operations, SiamFC [14] introduces the naive cross-correlation approach, which convolves the template features with the search region features, ultimately outputting a response map representing the potential presence of the target with a single-channel representation. As different object categories are activated in different channels of the feature map, SiamRPN++ [16] proposes deep cross-correlation, which performs convolution operations along the channel dimension, aiming to learn more discriminative response maps. Alpha-refine [23] introduced pixel-wise cross-correlation, convolving each pixel feature vector of the template features individually with the search features. Subsequently, pixel-to-global correlation and saliency-associated correlation are also proposed in PGNet [31] and SAOT [33], respectively. Nevertheless, such methods are essentially convolutional operations, which are less capable of learning discriminative features from the training data and maintaining robustness in challenging scenarios.

Differently, in this paper, the proposed multi-scale feature fusion module and learnable feature matching module can compensate for the drawbacks noted above and attain a trade-off between accuracy and speed.

2.2. Temporal Information Exploitation

The input of deep trackers is a sequence of consecutive image frames, and, therefore, the temporal dimension should not be overlooked when designing a tracker. The temporal information enables the tracker to better adapt to variations in target features over time.

Unfortunately, most current trackers only perceive the tracking task as feature matching within a single-frame image, disregarding the temporal information, such as OCEAN [34] and HIFT [35].

There are also trackers dedicated to capturing the temporal changes in target features continuously to obtain an accurate representation of the target's current appearance in the search template features. For example, GCT [26] introduces graph neural networks to incorporate spatial–temporal features to handle feature variations in the target. UpdateNet [24] utilizes a template update subnet to adaptively fuse the latest target features in a nonlinear manner. GradNet [25] aims to exploit gradient information related to target changes to obtain an optimal representation of target features. Moreover, ATOM [36] utilizes recurrent neural networks (RNN) to model the temporal changes in target features.

Nevertheless, these methods suffer from high computational consumption as well as the inability to address the feature contamination introduced by deformation, occlusion, and other cases. Different from the previous work, this paper proposed a simple and effective template-constraint branch, which is used to extract the target features at the temporal level and set a filtering strategy to filter the contaminated target features. It is able to improve the tracking accuracy of the tracker with a small computational overhead.

3. Proposed Methods

Now we introduce ConcatTrk Network in detail. In the tracking process, the input consists of a sequence of continuous images denoted as I, with a total length of N. The initial target bounding box, denoted as y_0, is provided in the first image I_0. The initial template region Z_0 and the search region X_0 can be obtained by performing a cropping operation. The tracker then predicts the possible location y_i of the target in the subsequent images. Figure 2 illustrates the decomposition of the network structure into four subcomponents: the multi-scale feature fusion module, the learnable feature matching module, the predict head, and the template-constraint branch.

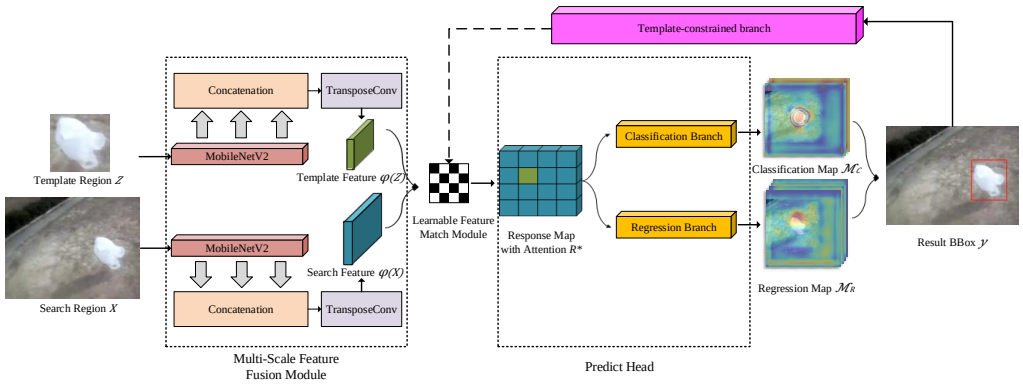


Figure 2. Overview of the ConcatTrk network architecture. Starting from the left, the network contains a multi-scale feature fusion module, a learnable feature matching module, a predict head, and a template-constraint branch.

3.1. Multi-Scale Feature Fusion Module

Because low-level features have small perceptual fields and excellent detailed feature representation, whereas high-level features retain more semantic information, the fusion of different levels of depth features is an effective way to improve feature representation. There have also been a number of multi-scale feature fusion methods (e.g., FPN networks [37], HRNet networks [38], etc.) but they occupy relatively excessive computational resources. The proposed multi-scale feature fusion module is based on the concept of "fusion before matching". Features from different convolution layers are first concatenated and dimensionally reduced to generate fused features, which are then input into the learnable feature matching module. This approach avoids redundant computations in the "matching first"

methods (such as SiamRPN++ [16], SiamBAN [18]), thereby improving tracking speed. The workflow of our multi-scale feature fusion module is shown in Figure 3.

ConcatTrk sets up a pair of parallel convolutional networks as feature extractors for the input images. It contains two branches: a branch for extracting the features of the search region, which takes the search region X as input, and another branch for extracting the template features, which takes the template region Z as input. Both branches consist of a fully convolutional backbone network with shared weights, whose processing is denoted by $\varphi(\cdot)$. They output the feature maps $\varphi(X)$ and $\varphi(Z)$ about the search region and the template region, respectively.

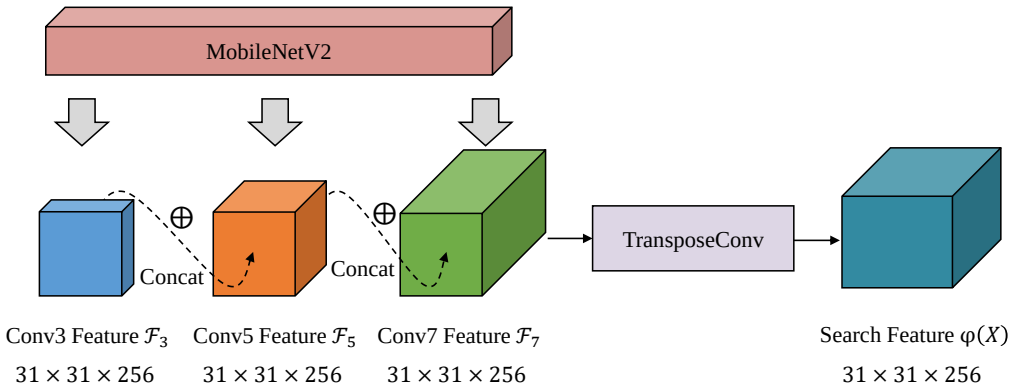


Figure 3. Workflow of multi-scale feature fusion module.

To facilitate the deployment of tracker on edge devices with limited resources, the modified MobileNetV2 [39] is selected as the backbone network for ConcatTrk. The convolution parameters of the backbone network are adjusted to ensure that the intermediate feature maps from different levels possess the same spatial dimensions. Within ConcatTrk, the extraction of intermediate features is focused on convolution layers 3, 5, and 7 of the backbone. These layers are denoted as $\mathcal{F}_3(\cdot)$, $\mathcal{F}_5(\cdot)$, and $\mathcal{F}_7(\cdot)$, respectively.

Taking the search branch as an example, the three intermediate feature maps $\mathcal{F}_3(X)$, $\mathcal{F}_5(X)$, and $\mathcal{F}_7(X)$ are first concatenated in the channel dimension, as written in Equation (1).

$$\mathcal{F}_{3,5,7}(X) = concat(\mathcal{F}_3(X), \mathcal{F}_5(X), \mathcal{F}_7(X)) \tag{1}$$

The shape of the $\mathcal{F}_{3,5,7}(X)$ is $31 \times 31 \times 768$, and then it is inputted to the transposed convolution blocks for dimensionality reduction to obtain the multi-scale feature map $\varphi(X)$, and the final shape of $\varphi(X)$ is $31 \times 31 \times 256$, which is shown in Equation (2).

$$\varphi(X) = TransConv(\mathcal{F}_{3,5,7}(X)) \tag{2}$$

3.2. Learnable Feature Matching Module

Correlation-type methods have demonstrated surprising capability, serving as a popular feature matching module in Siamese trackers. However, these methods are essentially convolutional operations to calculate the similarity score between the template and the search region, which are simple non-parametric processes. This leads to the inability of correlation-type methods to learn the sample distribution from the training data and further results in the loss of semantic information and degradation of tracking accuracy.

To remedy this drawback, we migrate and improve a concatenation-based matching operator as our learnable feature matching module, noted in AutoMatch [40]. The introduction of the learnable feature matching module does not significantly increase the computational cost, but it is more discriminative. It contains an embedding module, a feature response module, and an attention module, as shown in Figure 4.

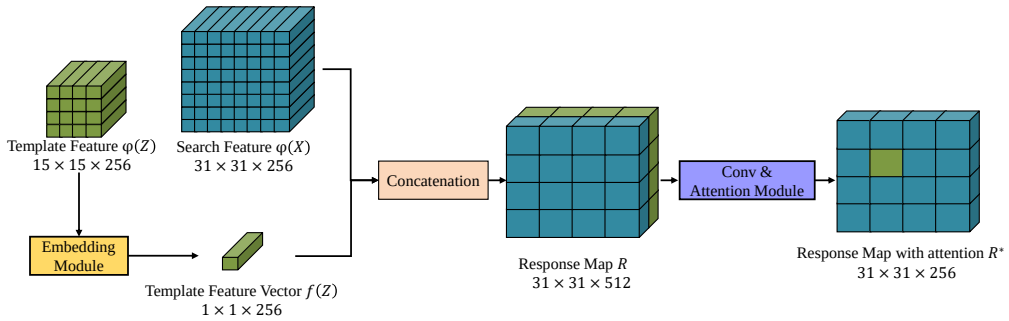


Figure 4. Workflow of the learnable feature matching module.

For an intuitive description, we do not consider the work process of template-constraint branching at this time. First, the feature map $\varphi(Z)$ from the template branch is input to the embedding module, which consists of several convolution blocks; $\varphi(Z)$ is embedded as a $1 \times 1 \times 256$ feature vector $f(Z)$. Then, $f(Z)$ is concatenated into each element of $\varphi(X)$ along the channel dimension and input to the feature response module to calculate the response map $\mathcal{R}$. The concatenation and feature response operations are denoted by the symbol $\star$ in Equation (3).

$$\mathcal{R} = f(Z) \star \varphi(X) \tag{3}$$

When the template-constraint branch is working, the template feature vector $f(Z)$ will fuse the temporal information characterizing the latest appearance of the target into the feature vector $f^t(Z)$ according to the tracking results of the previous frames. Therefore, Equation (3) can be rewritten as Equation (4). The details about the template-constraint branch are elaborated in Section 3.4.

$$\mathcal{R} = f^t(Z) \star \varphi(X) \tag{4}$$

Importantly, the response map $\mathcal{R}$ lacks spatial and channel attention. To address this limitation, an attention module is employed to activate attention within $\mathcal{R}$. Specifically, the non-local layer [41] is utilized to capture intra-frame attention, whereas the SEModule [42] is leveraged to capture inter-channel attention. The resulting response map with attention is denoted as $\mathcal{R}^*$.

3.3. Predict Head

The predict head computes the categories directly for each location (i, j) in $\mathcal{R}^*$ and regresses to the target bounding box via an end-to-end approach. It is implemented by a fully convolutional network, which is able to avoid tricky parameter tuning and reduce human intervention.

This module can be decomposed into two subbranches: the branch that computes the fore-background classification and the branch that regresses the target bounding box. The input of the predict head is the response map $\mathcal{R}^*$ from the learnable feature matching module. The classification branch outputs a classification map $\mathcal{M}_C$ with two channels, where each position (i, j) represents the probability that the search area characterized by that position is the fore-background. The regression branch outputs a regression map $\mathcal{M}_R$ with four channels, where each position (i, j) represents a 4D vector (l, t, r, b), representing the distance from the left, top, right, and bottom border of the target bounding box respectively.

Since this module contains both the classification task and the regression task, it is critical to calculate the classification loss and the regression loss separately according to their respective outputs. Consequently, the classification task is addressed using the cross-

entropy loss, whereas the regression task is handled using the IoU loss. Thus, the total loss function of the whole network is

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{reg} \tag{5}$$

3.4. Template-Constraint Branch

The template-constraint branch is deployed as an additional module of the network in tracking post-processing, which is used to capture and fuse the high-dimensional feature representation of the target appearance that changes in the temporal order via concatenation. The proposed template-constraint branch is simple yet effective, involving no complex gradient iterations or deeper network architecture. It filters out contaminated template images based on cosine similarity and updates high-quality template features online via concatenation. The experimental results in Section 4.3.2 demonstrate that this branch only slightly reduces tracking speed while delivering significant improvements in accuracy.

The template-constraint branch accepts the tracking result y_i and the ith image I_i as its input. It first obtains the resultant region $result_i$ of the current frame by crop operation and then performs the similarity discrimination. If the cosine similarity between $result_i$ and the initial search region Z is in the threshold interval $[a, b]$, the multi-scale fused feature vector $f(result_i)$ will be concatenated with the current tracking template vector in the channel dimension, inputted into the convolution module for dimensionality reduction, and finally get the template feature vector $f^t(Z)$ fused with the temporal dimensionality information. The value of $f^t(Z)$ will be updated periodically and will participate in the prediction of the tracking network during the next threshold interval. The operation logic of the branch is formalized in Equation (6).

$$f^t(Z) = \begin{cases} TransConv(cat(f^t(Z), f(result_i))), & cossimilar(result_i, Z) \in [a, b] \\ f^t(Z), & otherwise \end{cases} \tag{6}$$

In particular, in the initial frame since the template constraint branch has not yet been run, both $f_t(Z)$ and $f(Z)$ represent meanings of the multi-scale feature vectors of the initial target position.

The motivation for setting the threshold interval is to exclude undesirable cases of minor changes and contamination of the target appearance features. In order to ensure operational efficiency, the temporal constraint branch runs only after a fixed interval ξ. The threshold interval $[a, b]$ and update interval ξ are specified as hyperparameters and further optimized by a hyperparameter search strategy in the fine-tuning phase after training.

4. Results and Comparison

4.1. Implementation and Training Details

ConcatTrk is implemented in PyTorch with Intel i9-9900X, 32G RAM, and Nvidia RTX 3090 as the hardware environment for the training phase. The whole network is trained on ImageNet VID and COCO datasets based on an end-to-end training approach. The template image size is 127×127 pixels, and the search image size is 255×255 pixels. In particular, the ground-truth classification label during the training phase is consistent with SiamBAN [18].

The backbone network of the ConcatTrk is MobileNetv2 [39], which is pre-trained on the ImageNet dataset. The training strategy of the whole network is SGD strategy, the batchsize is set to 32, the whole training process contains 20 epochs, the learning rate warming up strategy is used in the first 5 epochs to increase the learning rate from 0.001 to 0.005, and, in the last 15 epochs, the learning rate is continuously reduced to 0.00005, weight decay is set to 0.0001, and momentum is 0.9.

4.2. Results and Comparison

To evaluate ConcatTrk in detail, comparison experiments are performed on three popular benchmarks, including OTB100 [28], UAV123 [27], and LaSOT [29]. Multiple scenes along with the tracked target are presented in Figure 5, which serves as evidence that the experiments on the three benchmarks validate the robustness of the tracker. ConcatTrk is comprehensively compared with 18 other outstanding trackers, including SiamRN [43], TransT [20], SiamBAN [18], SiamFC++ [44], SiamRPN++ [16], GradNet [25], etc.

(**a**)

(**b**)

(**c**)

Figure 5. Examples of diverse tracking scenes and tracked targets from the test benchmark. The three subfigures (**a**–**c**) are selected from the UAV123, OTB100, and LaSOT datasets, respectively.

4.2.1. Evaluation Metrics

In the evaluation, we employed one pass evaluation (OPE) metrics to assess the success score and precision score of the tracker on the test dataset. OPE initializes the first frame of the test video sequence based on the ground truth bounding box and does not intervene in the subsequent tracking process, even in cases of target loss. The calculation details for the success score and precision score are described below.

The success score measures the accuracy of target localization by comparing the intersection over union (IoU) between the predicted bounding box $\hat{y^i}$ and the ground truth bounding box y^i. It is calculated as the ratio of the intersection area between $\hat{y^i}$ and y^i to the union area. Specifically, for a given threshold $T \in [0, 1]$ and a test set consisting of N images, if the IoU of the bounding box in the ith frame exceeds T, the frame is considered a success; otherwise, it is considered a failure. By plotting the curve of the proportion of successful frames in the test set against the threshold T and calculating the area under the curve (AUC), the success score can be obtained. This process can be expressed as Equation (7).

$$SuccScore = \int_0^1 \frac{1}{N} \sum_{i=1}^{N} \mu(\frac{\hat{y^i} \cap y^i}{\hat{y^i} \cup y^i} - T)dT \tag{7}$$

The function $\mu(\cdot)$ represents the step function used to filter the successfully tracked image frames, which can be expressed as Equation (8).

$$\mu(t) = \begin{cases} 1 & ,t > 0 \\ 0 & ,otherwise \end{cases} \tag{8}$$

The success score serves as a quantitative measure of the tracker's ability to accurately localize the target throughout the sequence. The curve provides insights into the tracker's performance across a range of IoU thresholds, allowing us to assess its robustness and sensitivity to different levels of overlap between the predicted and ground truth bounding boxes.

The calculation method of the precision score is similar to that of the success score, with the difference being that the precision score is computed based on the center location error (CLE) between $\hat{y}^i$ and y^i in the ith frame.

4.2.2. Results of Tracking Speed

In order to evaluate speed performance on edge devices fairly, all trackers are deployed on the edge computing platform Nvidia AGX Xavier and tested on the UAV123 dataset. NVIDIA AGX Xavier is a high-performance system-on-a-chip (SoC) designed specifically for AI applications, offering a range of capabilities. With power consumption ranging from 10 W to 30 W and computational power reaching up to 32TOPS, it delivers exceptional performance. This makes it well-suited for deployment in diverse edge devices that demand efficient execution of AI applications, including autonomous vehicles, robots, and drones. AGX Xavier's wide-ranging applications in these domains highlight its significance in enabling advanced AI-driven functionalities in such edge devices.

Table 1 summarizes the performance of SOTA trackers. We compared the success score, precision score, tracking speed (FPS), and FLOPs. ConcatTrk is able to obtain competitive results while maintaining tracking speeds of 41 FPS. Figure 6 shows that ConcatTrk would meet real-time tracking requirements (i.e., >30 FPS) on the edge device while obtaining reliable tracking accuracy, which is not achieved by the other trackers.

Table 1. Comparisons with state-of-the-art trackers in terms of the success score, precision score, FPS, and FLOPs on the UAV123 dataset. The best three performances are, respectively, shown in **red**, **green**, and **blue**.

Tracker	Succ. Score	Pre. Score	Avg. FPS	FLOPs
TransT	0.660	0.852	5	16.7 G
SiamBAN	0.631	0.833	6	48.8 G
ConcatTrk(Ours)	**0.623**	**0.807**	**41**	**3.4 G**
SiamFC++	0.617	0.799	13	17.5 G
SiamRPN++	0.611	0.804	6	48.9 G
SiamDWfc	0.536	0.776	19	12.9 G
SiamFC	0.475	0.702	22	2.7 G
GradNet	0.376	0.555	18	4.2 G

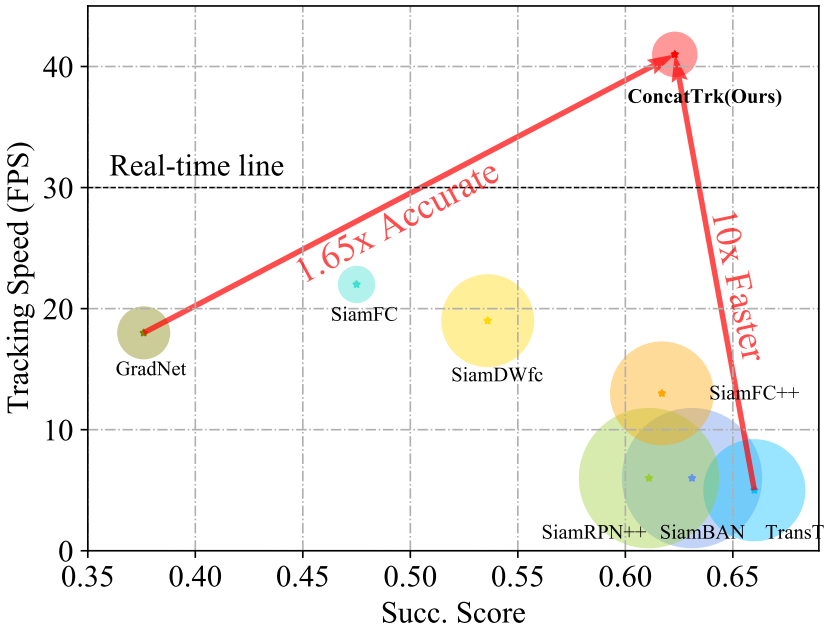


Figure 6. Comparisons with state-of-the-art trackers in terms of the success score, FPS, and model FLOPs on the UAV123 dataset. The circle diameter is in proportion to the FLOPs.

ConcatTrk gains the best tracking speed (41 FPS), surpassing the best and the second-best accuracy tracker TransT(4 FPS) and SiamBAN(8 FPS), while using 4.9× and 14.3× fewer FLOPs, respectively. In comparison to GradNet, which also incorporates dynamically updating target templates, ConcatTrk demonstrates tracking accuracy that is 1.65× higher and speed that is 2.27× faster. The real-time tracking speed of ConcatTrk allows its deployment on edge devices, such as robots, cameras, and self-driving terminals, with higher application value.

4.2.3. Results from UAV123

The UAV123 [27] dataset is an important benchmark in the field of aerial tracking, consisting of 123 sequences taken from the perspective of UAV aerial photography. Each sequence is labeled with specific challenging scenes, including scale variation, partial occlusion, full occlusion, out-of-view, fast motion, camera motion, background clutter, similar object, aspect ratio change, viewpoint change, low resolution, etc.

Quantitative evaluation of the UAV123 dataset is shown in Figure 7. In Figure 7a,b, ConcatTrk is able to achieve comparable tracking accuracy. In terms of the success score, compared with the best and second-best tracker TransT(0.660) and SiamBAN(0.631), ConcatTrk(0.623) has a 5.6% and 1.3% accuracy degradation, respectively. In terms of the precision score, ConcatTrk(0.807) has a 5.3% and 3.1% precision degradation, respectively. However, ConcatTrk is able to trade an acceptable loss for more than 8× the tracking speed boost.

It is worth noting that ConcatTrk has an excellent result in the background clutter scenario. Shown in Figure 7c,d, ConcatTrk(0.495/0.714) gains the top-1 success score and precision score with 16.5% and 17.4% improvement over the TransT(0.425/0.608). The reason is that the learnable position-by-position similarity calculation is able to obtain similar responses with strong discriminative properties.

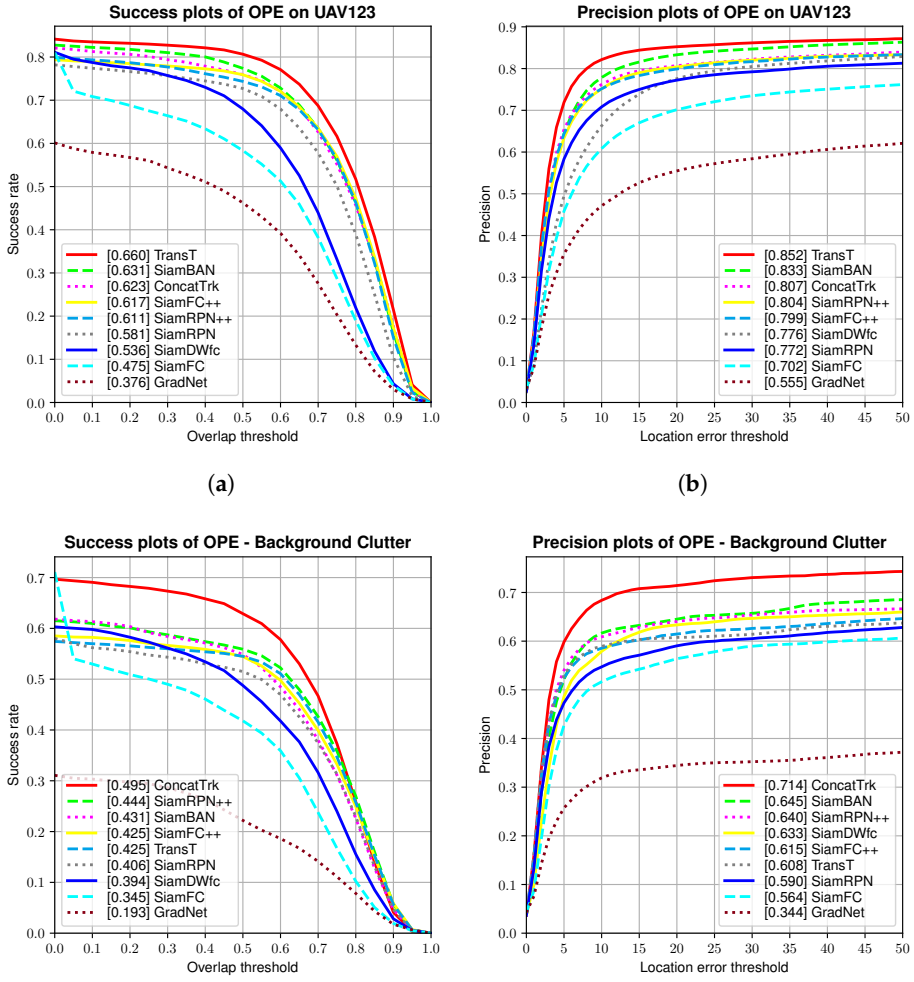


Figure 7. Comparisons on UAV123 dataset. Subfigure (**a**,**b**) represent the success score plot and precision score plot on UAV123 dataset, and subfigure (**c**,**d**) represents the success score plot and precision score plot in the background clutter scenarios.

4.2.4. Results from OTB100

The OTB100 [28] dataset contains 100 challenging frame sequences captured from everyday life scenes and labeled with 9 attributes to represent specific difficult scenes, including illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutter, and low resolution.

We compared seven trackers, including SiamRN [43], TransT [20], GradNet [25], SiamRPN [15], USOT [45], and SiamFC [14] to obtain success and precision plots with the OPE evaluation method in Figure 8.

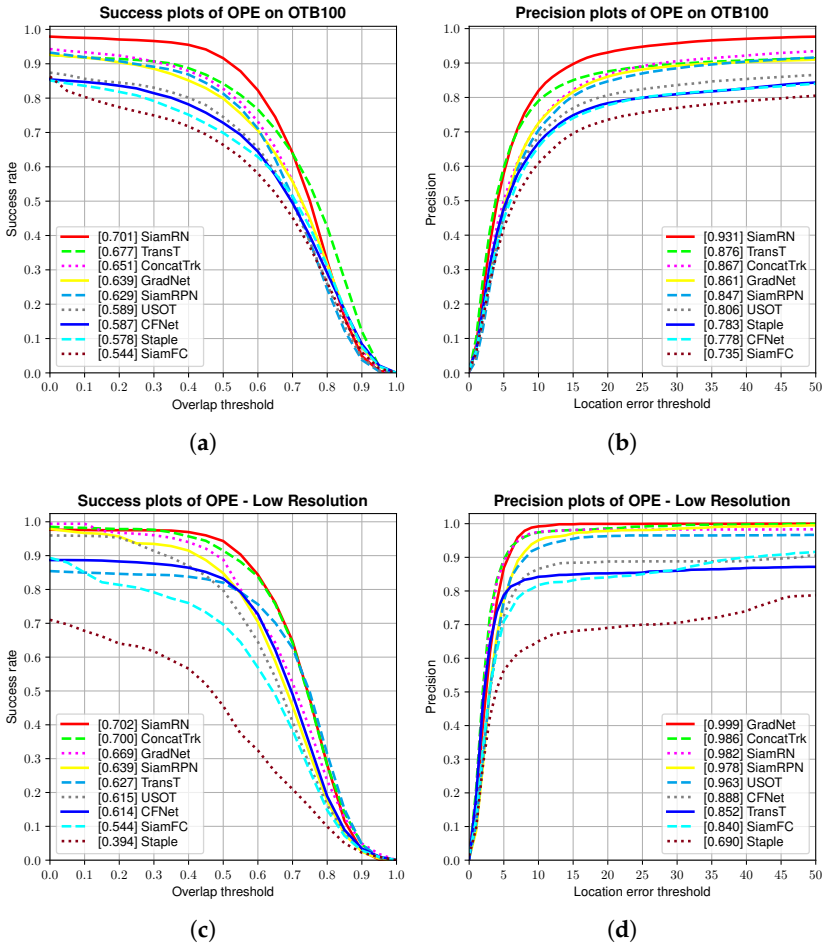


Figure 8. Comparisons on OTB100 dataset. Subfigure (**a**,**b**) represent the success score plot and precision score plot on OTB100 dataset. Subfigure (**c**,**d**) represent the success score plot and precision score plot in the low resolution scenario.

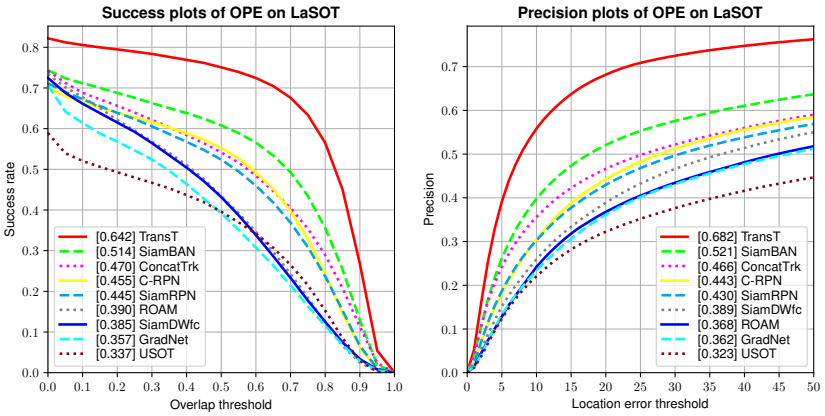
In Figure 8a,b, compared with the SOTA tracker SiamRN(0.701), ConcatTrk(0.651) is able to achieve a speedup of 483% with a 7.1% loss in accuracy. Compared with tracker GradNet(0.639), ConcatTrk(0.651) is able to obtain an accuracy improvement of 1.88%.

In particular, ConcatTrk has excellent performance in the low resolution scenario, where the success score (0.700) is only 0.28% lower than the 1st tracker SiamRN (0.702), and the pre score (0.986) is only 1.3% lower than the 1st tracker GradNet (0.999). The reason for this is that the position-by-position similarity calculation approach of the learnable matching module is able to obtain similar responses with strong discriminative properties.

4.2.5. Results from LaSOT

LaSOT [29] is a recently released large-scale single object tracking benchmark with a total of 280 tracking sequences containing numerically challenging scenes and more than 70 classes of targets, with high quality manual annotation for each frame. The LaSOT dataset places higher demands on the spatial discriminability and long-term generalization of trackers.

We compared several SOTA trackers, including TransT [20], SiamBAN [18], C-RPN [21], ROAM [46], SiamDW [47], GradNet [25], and USOT [45], and plotted success and precision plots under the OPE evaluation method. As shown in Figure 9, the success score of ConcatTrk (0.470) has a gap with the top-1 tracker TransT(0.642). However, ConcatTrk gains comparable results with SiamBAN(0.514), proving that SiamSTC achieves a balance between tracking performance and efficiency.



(**a**) (**b**)

Figure 9. Comparisons on LaSOT dataset. Subfigure (**a**,**b**) represent the success score plot and precision score plot on LaSOT dataset.

4.3. Ablation Experiment

4.3.1. Impact of Learnable Feature Matching Module

The motivation for introducing the learnable feature matching module is to address the limitation of the correlation-type methods that cannot learn the sample distribution from the training data to improve the discriminability of the response maps.

To ensure a fair comparison, we establish a baseline using the ConcatTrk network without the multi-scale feature fusion module and template-constraint branch; detailed studies are conducted on the OTB100 dataset. Our learnable feature matching module is compared with two commonly used alternatives: depth-wise correlation [16] and pixel-wise correlation [23].

As shown in Table 2, the learnable feature matching method is capable of bringing improvement to tracking accuracy. In particular, the improvement is more significant in challenging scenarios such as low resolution, out-of-view, and scale variation.

Table 2. Ablation experiments on the learnable feature matching module. The best two performances are, respectively, shown in **red** and **green**. The Δ denotes the improvement in comparison with the second-best tracker.

Attributes	ALL		Low-Resolution		Out-of-View		Scale-Variation	
	Succ. Score	Pre. Score	Succ. Score	Pre. Score	Succ. Score	Pre. Score	Succ. Score	Pre. Score
Depth-wise XCorr [16]	0.586	0.798	**0.597**	**0.826**	0.454	0.654	0.575	0.775
Pixel-wise XCorr [23]	**0.610**	**0.801**	0.553	0.761	**0.487**	**0.660**	**0.589**	**0.787**
LFM(Ours)	**0.613**	**0.808**	**0.698**	**0.988**	**0.537**	**0.683**	**0.622**	**0.816**
Δ(%)	+0.491	+0.874	+17.01	+19.61	+10.267	+3.484	+5.603	+3.685

4.3.2. Impact of Template-Constraint Branch

The template-constraint branch is created to continuously capture and update target feature changes for an improved feature representation, leading to better accuracy and discrimination of the tracker. The effectiveness of the branch is evaluated on the OTB100 dataset by comparing the accuracy of the tracker with and without the template-constraint branch as well as investigating its impact on tracking speed. The results are denoted by ConcatTrk and ConcatTrk-noTCB to indicate whether the template-constraint branch is used.

As shown in Table 3, in terms of speed, the introduction of the template-constraint branch brings about 3.5% speed loss to the tracking network; in terms of accuracy, the template-constraint branch can bring 3.99% accuracy gain to the tracking model. Moreover, in most of the challenging scenarios, the tracker can obtain the accuracy gain brought by the template-constraint branch, especially in the scenarios of illumination variation and background clutter.

Table 3. Ablation experiment on the template-constraint branch. The Δ denotes the improvement in comparison with ConcatTrk-noTCB. The better performance is highlighted in **red**.

	Succ. Score	Pre. Score	FPS
ConcatTrk	**0.651**	**0.871**	140
ConcatTrk-noTCB	0.626	0.828	**145**
Δ(%)	+3.99	+4.71	−3.5%

4.4. Real-World Test

In this section, ConcatTrk is further deployed on a drone platform, as illustrated in Figure 10, to validate its practicality in real-world applications. The real-world test aims to evaluate the power consumption of ConcatTrk and to verify its robustness in challenging scenarios. During the testing process, TensorRT was not utilized to accelerate model computation.

Figure 10. Drone platform in a real-world test.

Experimental Environment. In terms of hardware, the drone platform is based on the AmovLab P450. The platform is equipped with an Intel RealSense D435i camera as the image acquisition device, capable of capturing RGB images at a resolution of 1920 × 1080. Additionally, the platform is integrated with the Nvidia AGX Xavier as the onboard embedded processor, providing CPU and GPU computational resources. Regarding the software environment, the operating system is Ubuntu 18.04. Robot operating system (ROS) [48] is employed for inter-module message passing and functionality invocation on the drone, such as flight control, image acquisition, and visualization results publishing.

Power Consumption. We utilized the jtop toolkit to record the average power consumption of the drone under no-load conditions and during the execution of ConcatTrk to evaluate its energy consumption during real-world testing. The detailed results are presented in Figure 11. Under unloaded conditions, the drone only activated necessary functional modules and camera sensors. The working conditions additionally launched ConcatTrk on top of the no-load conditions. In this setting, the average total power consumption was approximately 7957 mW under no-load conditions and 12,895 mW under working conditions. It can be observed that ConcatTrk contributed to a power consumption of approximately 4938 mW, with the majority of the power being consumed by the CPU, GPU, and power module, accounting for 80.3% of the overall increase.

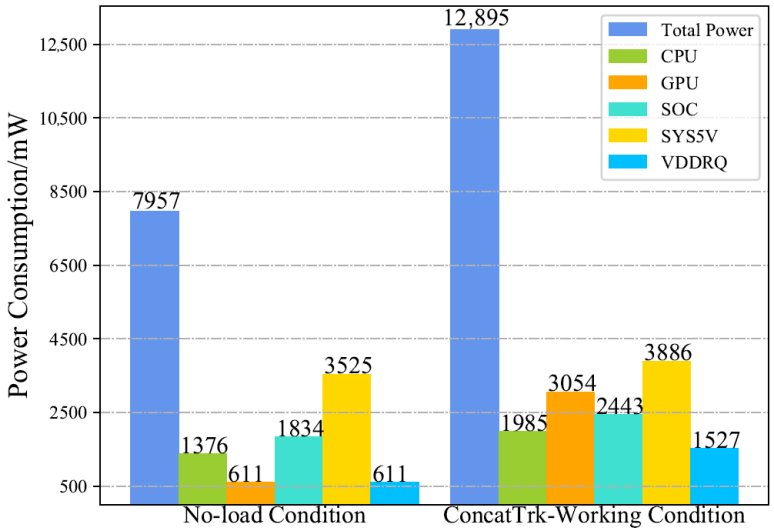


Figure 11. Power consumption of the drone platform in a real-world test. The power consumption of the entire system is divided into five components: CPU, GPU, SOC, VDDRQ, and SYS5V. Specifically, SOC represents the power consumption of the internal SOC processor, VDDRQ represents the power consumption of the power module, and SYS5V represents the power consumption of other components in the system.

Challenging Scenarios. Figure 12 illustrates the visualized results of the real-world test, where we introduced representative challenging scenarios, including partial occlusions (frames #301, #514), lighting variations (frames #408, #446, #514), deformations (frames #63, #150), and out-of-plane rotations (frames #3, #63, #246, #408). Due to ConcatTrk's efficient network architecture, it is capable of maintaining reliable tracking performance even in complex scenarios.

(**a**) #3 frame (**b**) #63 frame (**c**) #150 frame (**d**) #246 frame

(**e**) #301 frame (**f**) #408 frame (**g**) #446 frame (**h**) #514 frame

Figure 12. Visualization results of the real-world test on the drone platform with various challenging scenarios. The caption of the subfigures represent their order in the video sequence.

Furthermore, in practical deployment, the image capture and visualization processes may impact the performance of the tracker. However, ConcatTrk maintains a real-time tracking speed of 37.6 FPS. Overall, ConcatTrk demonstrates outstanding potential for real-world applications, enabling efficient operation on UAV AI platforms.

5. Conclusions

In this paper, a lightweight end-to-end Siamese tracker, ConcatTrk, is proposed for efficient tracking while maintaining real-time speed on edge devices. ConcatTrk includes a learnable feature matching module that enhances tracking accuracy by acquiring matching

experience from training data, and it outperforms non-parametric correlation-type methods. Additionally, the tracker includes a template-constraint branch that adjusts the search region in post-processing and dynamically captures changes in target appearance features, allowing for more robust and accurate performance in long-term tracking tasks. Extensive experiments have proven that ConcatTrk achieves excellent tracking efficiency on several challenging benchmarks. Real-world tests also revealed that ConcatTrk has outstanding application value. We hope our work will contribute to the application of object tracking methods in resource-constrained scenarios.

Author Contributions: Conceptualization, Z.W. and Q.L.; Data curation, Z.W.; Formal analysis, Z.W.; Funding acquisition, S.Z. and Q.L.; Investigation, Z.W.; Methodology, Z.W. and Q.L.; Project administration, Q.L. and S.Z.; Supervision, S.Z.; Validation, Z.W. and Q.L.; Visualization, Z.W., Z.Z. and Y.Z.; Writing—original draft, Z.W. and Q.L.; Writing—review and editing, Z.W. and S.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key R&D Projects in Sichuan Province under Grant 2020YFG0472, and the National Natural Science Foundation of China under Grant 62272089.

Data Availability Statement: Publicly available datasets were analyzed in this study. UAV123 dataset is available in [27]. OTB100 dataset is available in [28]. LaSOT dataset is available in [29].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wu, H.; Nie, J.; He, Z.; Zhu, Z.; Gao, M. One-shot multiple object tracking in UAV videos using task-specific fine-grained features. *Remote Sens.* **2022**, *14*, 3853. [CrossRef]
2. Heidari, A.; Jafari Navimipour, N.; Unal, M.; Zhang, G. Machine Learning Applications in Internet-of-Drones: Systematic Review, Recent Deployments, and Open Issues. *ACM Comput. Surv.* **2023**, *55*, 247. [CrossRef]
3. Li, B.; Li, Q.; Zeng, Y.; Rong, Y.; Zhang, R. 3D Trajectory Optimization for Energy-Efficient UAV Communication: A Control Design Perspective. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 4579–4593. [CrossRef]
4. Wang, B.; Zhu, D.; Han, L.; Gao, H.; Gao, Z.; Zhang, Y. Adaptive Fault-Tolerant Control of a Hybrid Canard Rotor/Wing UAV Under Transition Flight Subject to Actuator Faults and Model Uncertainties. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 4559–4574. [CrossRef]
5. Wang, B.; Zhang, Y.; Zhang, W. A composite adaptive fault-tolerant attitude control for a quadrotor UAV with multiple uncertainties. *J. Syst. Sci. Complex.* **2022**, *35*, 81–104. [CrossRef]
6. Dai, X.; Xiao, Z.; Jiang, H.; Lui, J.C. UAV-Assisted Task Offloading in Vehicular Edge Computing Networks. IEEE *Trans. Mobile Comput.* **2023**, 1–15. [CrossRef]
7. Cao, B.; Li, M.; Liu, X.; Zhao, J.; Cao, W.; Lv, Z. Many-Objective Deployment Optimization for a Drone-Assisted Camera Network. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2756–2764. [CrossRef]
8. Zhao, J.; Gao, F.; Jia, W.; Yuan, W.; Jin, W. Integrated Sensing and Communications for UAV Communications with Jittering Effect. *IEEE Trans. Netw. Sci. Eng.* **2023**, *12*, 758–762. [CrossRef]
9. Sandoval, L.A.C. Low Cost Object Tracking by Computer Vision Using 8 Bits Communication with a Viper Robot. In Proceedings of the 2023 8th International Conference on Control and Robotics Engineering (ICCRE), Niigata, Japan, 21–23 April 2023; pp. 232–237. [CrossRef]
10. Lee, M.F.R.; Chen, Y.C. Artificial Intelligence Based Object Detection and Tracking for a Small Underwater Robot. *Processes* **2023**, *11*, 312. [CrossRef]
11. Nebeluk, R.; Zarzycki, K.; Seredyński, D.; Chaber, P.; Figat, M.; Domański, P.D.; Zieliński, C. Predictive tracking of an object by a pan–tilt camera of a robot. *Nonlinear Dyn.* **2023**, *111*, 8383–8395. [CrossRef]
12. Gragnaniello, D.; Greco, A.; Saggese, A.; Vento, M.; Vicinanza, A. Benchmarking 2D Multi-Object Detection and Tracking Algorithms in Autonomous Vehicle Driving Scenarios. *Sensors* **2023**, *23*, 4024. [CrossRef] [PubMed]
13. Nie, C.; Ju, Z.; Sun, Z.; Zhang, H. 3D Object Detection and Tracking Based on Lidar-Camera Fusion and IMM-UKF Algorithm Towards Highway Driving. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 1242–1252. [CrossRef]
14. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
15. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980. [CrossRef]
16. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 4277–4286. [CrossRef]

17. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338.
18. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 6668–6677.
19. Wang, N.; Zhou, W.; Wang, J.; Li, H. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 1571–1580.
20. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 8126–8135.
21. Fan, H.; Ling, H. Siamese cascaded region proposal networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7952–7961.
22. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning spatio-temporal transformer for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 19–25 June 2021; pp. 10448–10457.
23. Yan, B.; Zhang, X.; Wang, D.; Lu, H.; Yang, X. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 5289–5298.
24. Zhang, L.; Gonzalez-Garcia, A.; Weijer, J.v.d.; Danelljan, M.; Khan, F.S. Learning the model update for siamese trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 4010–4019.
25. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 6162–6171.
26. Gao, J.; Zhang, T.; Xu, C. Graph convolutional tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4649–4659.
27. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 445–461.
28. Wu, Y.; Lim, J.; Yang, M. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
29. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. 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, USA, 16–20 June 2019; pp. 5374–5383.
30. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. 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, USA, 16–20 June 2020; pp. 6269–6277.
31. Liao, B.; Wang, C.; Wang, Y.; Wang, Y.; Yin, J. Pg-net: Pixel to global matching network for visual tracking. In Proceedings of the European Conference on Computer Vision, Seattle, WA, USA, 16–20 June 2020; pp. 429–444.
32. Nam, H.; Baek, M.; Han, B. Modeling and Propagating CNNs in a Tree Structure for Visual Tracking. *arXiv* **2016**, arXiv:1608.07242.
33. Zhou, Z.; Pei, W.; Li, X.; Wang, H.; Zheng, F.; He, Z. Saliency-associated object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 19–25 June 2021; pp. 9866–9875.
34. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the European Conference on Computer Vision, Seattle, WA, USA, 16–20 June 2020; pp. 771–787.
35. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. HiFT: Hierarchical feature transformer for aerial tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 19–25 June 2021; pp. 15457–15466.
36. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. *arXiv* **2019**, arXiv:1811.07628.
37. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
38. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 5693–5703.
39. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
40. Zhang, Z.; Liu, Y.; Wang, X.; Li, B.; Hu, W. Learn to match: Automatic matching network design for visual tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 19–25 June 2021; pp. 13339–13348.
41. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7794–7803.
42. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
43. Cheng, S.; Zhong, B.; Li, G.; Liu, X.; Tang, Z.; Li, X.; Wang, J. Learning to filter: Siamese relation network for robust tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Online, 19–25 June 2021; pp. 4421–4431.

44. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12549–12556.
45. Zheng, J.; Ma, C.; Peng, H.; Yang, X. Learning to track objects from unlabeled videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 19–25 June 2021; pp. 13546–13555.
46. Yang, T.; Xu, P.; Hu, R.; Chai, H.; Chan, A.B. ROAM: Recurrently optimizing tracking model. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 6717–6726.
47. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
48. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.