LightOcean: A Lightweight And Efficient Network For Real-time UAV Tracking

Haiyang Chen*†, Weiqiang Wang*†, Xingzhou Zhang‡§, ✉Wei Zhou*†, Weisong Shi¶
*Engineering Research Center of Cyberspace, Yunnan University, Kunming 650091, China
†School of Software, Yunnan University, Kunming 650091, China
‡Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
§University of Chinese Academy of Sciences, Beijing 100190, China
¶Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA

***Abstract*—Siamese-based trackers have significantly advanced in visual object tracking over the past years. However, most of these trackers emphasize tracking accuracy over efficiency, which limits their real-world deployment on edge platforms with widespread applications such as unmanned aerial vehicles (UAVs). In this paper, we propose LightOcean, a lightweight and efficient aerial tracker. Specifically, we design a dynamic template feature update module and a pixel-level cross-correlation module to improve the robustness and adaptability of the tracker without additional model computation. The former allows the tracker to capture additional space-time information during the tracking process to adapt to the appearance changes of the object. The latter utilizes pixel-level cross-correlation and attention mechanisms to generate similarity maps, improving the accuracy of predicted boundaries. By combining the lightweight and efficient backbone network and the prediction head, LightOcean significantly increases the tracking speed while maintaining accuracy. By running three common UAV benchmarks on Jetson Nano, a typical edge-embedded device, LightOcean presents better performance. The tracking speed is 4x faster than the state-of-the-art tracker, Ocean, while the energy consumption and memory usage are reduced by 80% and 12%. What's more, the accuracy remained stable, and the precision of LightOcean is 78.6%, which is 2.7% higher than Ocean.**

***Index Terms*—Edge computing, UAVs, Object tracking, Lightweight network**

I. Introduction

With the advantages of small size, high flexibility, and strong safety, Unmanned Aerial Vehicles(UAVs) have been widely used in commercial, agricultural, transportation, and other cloud-edge collaboration fields. Benefiting from the fast development in edge computing and computer vision, more and more real-time computer vision tasks are being deployed in edge devices like UAVs. As the fundamental tasks of computer vision, object tracking have a wide range of application prospects in the UAV scene, such as wildlife rescue [35], vehicle tracking[27], cinematography[29], etc.

In the tracking tasks, the UAVs will continuously locate and follow the designated target, which requires a real-time and accurate tracking algorithm. However, there are some challenges in achieving accurate and real-time UAV tracking due to the complexity of aerial scenarios. Firstly, the UAV

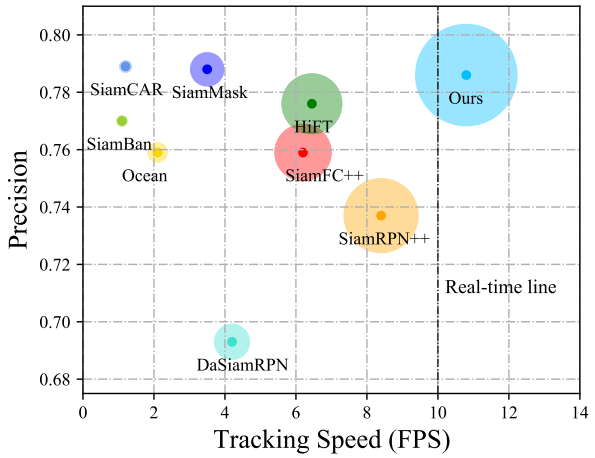


Fig. 1: A comparison of state-of-the-art trackers and the proposed LightOcean. The size of the circle represents the combined weight of tracking speed(X-axis) and precision(Y-axis). The larger the circle, the better the performance.

tracking tasks are dynamic. They are more susceptible to illumination changes, rapid movement, scale and size changes, complex background information, and other factors, which puts forward higher requirements for the robustness and adaptability of the tracking algorithm. Secondly, due to the limitation of energy consumption, the computing resources that UAVs can allocate to the tracking are limited, which is difficult to meet the requirements of trackers with high computational complexity.

In the field of visual object tracking, trackers are mainly divided into two types: correlation filter-based trackers, and deep learning-based trackers. Among them, the tracker based on traditional correlation filtering runs at high speed on the CPU due to its high efficiency in the Fourier domain. Still, these trackers cannot play an influential role when faced with the everyday challenges of UAV tracking, such as occlusion and disappearance. While another type of trackers, siamese-based trackers, has achieved state-of-the-art results on most UAV benchmarks. However, the success of siamese-based trackers relies on complex network structures and online

update mechanisms. Achieving superior performance comes at the expense of substantial computational and resource consumption, which limits their application in real UAV tracking. All in all, existing trackers do not achieve a satisfactory balance between tracking efficiency and performance.

In this paper, we propose a lightweight tracking framework, `LightOcean` tracker, which effectively achieves the above balance. Specifically, we first analyze the latency and resource consumption of different backbone networks on edge devices and apply ShuffleNetV2 with the best edge performance to the tracker as a feature extraction network. Subsequently, we noticed that although the lightweight backbone network can effectively improve the tracking efficiency, it is not enough to extract robust discriminative features due to the reduction of the network depth and the number of parameters. To improve tracking performance, we introduce a dynamic template update network, which allows the tracker to dynamically adapt to the appearance changes of the target without increasing the model computation. This network decides whether to update the dynamic template features according to the confidence score and uses cross attention to improve the critical information in the dynamic template features. In addition, we design a pixel-level feature fusion network, which reduces the interference of background information in the search image, improves the flexibility and efficiency of the tracker. Finally, we optimize the prediction network of the model by using lightweight convolutions, which further reduce the model's computational load and inference latency while slightly affecting the tracking performance. Figure 1 powerfully demonstrates the speed advantage of `LightOcean` while maintaining comparable accuracy and robustness to the deeper tracker.

The contributions of this paper are as follows.

- We design a lightweight tracking framework, `LightOcean`, which is specially used for UAV tracking tasks with complex scenes and limited resources. Combined with a lightweight backbone network and prediction heads, `LightOcean` is able to run on edge platforms with low latency and resource consumption.
- We introduce a dynamic template update module and a pixel-level feature fusion module and apply them to the tracker, which improves the model's robustness and adaptability without increasing the model's computational cost.
- We evaluate the performance of `LightOcean` on three UAV benchmarks, UAV123@10FPS, UAV20L, and DTB70. Running on typical edge embedded devices, the tracking speed is 4x faster than the state-of-the-art tracker, Ocean, while the energy consumption and memory usage are reduced by 80% and 12%.

II. Related work

A. Object Tracking

Correlation filter based object tracking. The method based on correlation filtering is common and traditional in target tracking, which has a tremendous advantage in tracking speed. MOSSE [2] and KCF [15] were the first to promote the development of correlation filtering in the field of target tracking. After that, some trackers [9, 10, 31] improve tracking performance from color features, multi-scale features, multi-channel features, and depth features. However, these artificially designed features are challenging to cope with illumination changes, rapid movement, and scale and size changes in UAV tracking, which make the correlation filtering algorithm with poor robustness and adaptability unsuitable for UAV tracking.

Deep learning based object tracking. Recently, deep tracking tracker, especially the siamese-based tracker have gained significant popularity in the visual tracking field due to their excellent robustness and adaptability. The pioneering work of Siamese trackers, SiamFC [1], first introduced the feature cross-correlation operation into the Siamese architecture without updating the template image while ensuring a sure accuracy at high speed. By referring to the target detection task, SiamRPN [22] introduces region proposal networks into the tracking task. SiamRPN++ [23] improves on SiamRPN by using a spatial aware sampling strategy to remove the impact of padding operation and proposes a model architecture to perform depth-wise and layer-wise aggregations. Ocean [38], SiamFC++ [36] replace the anchor base with an anchor-free mechanism to solve the limitation of the anchor-based RPN method that requires a lot of prior knowledge and is computationally complex. In addition, ATOM [8] and DIMP [7] achieved excellent performance by combining discriminative online classifier and siamese structure. Although these methods have achieved excellent performance on high-performance GPUs, yet they bring additional computation and energy consumption, which makes them difficult to deploy at the edge and operate on embedded platforms such as UAVs.

B. Lightweight And Efficient Neural Network

With the increasingly complex structure of deep neural network models and higher and higher computational load, the demand for hardware resources is gradually increasing, which greatly limits its application in real scenarios, such as edge embedded applications. One way to solve this problem is to manually design a lightweight and efficient network. For example, SqueezeNet [19] is the first work to focus on model size, which reduces the number of parameters of the model by reducing the size of the convolution kernel and group convolution. MobileNetV1 [16] uses a depthwise separable operation instead of standard convolution to reduce parameters of network, and convolves different convolution kernels on each input channel. As an improved network of MobileNetV1, MobileNetV2 [32] proposes inverted residual

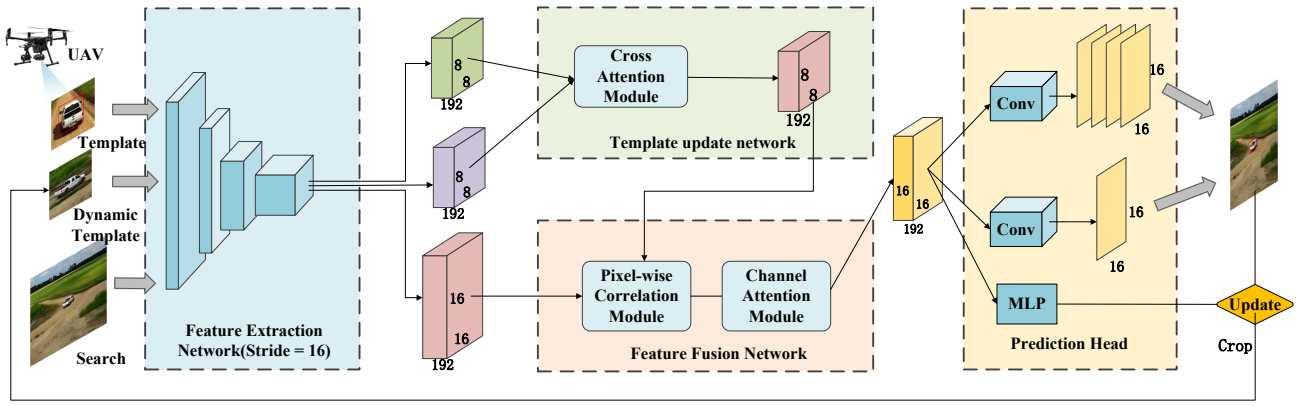


Fig. 2: The architecture of LightOcean, which consists of four parts: feature extraction network, template update network, feature fusion network, and classification regression prediction network

layers by introducing the residual idea in ResNet. More recent work, ShuffleNetV2 [28] proposes four practical guidelines for efficient network design, arguing that FLOPS cannot fully measure the complexity of the model, and the factors that affect the operation speed of the model also include GPU, memory usage, etc. In this work, we follow lightweight and efficient principles to design the network structure, such as using an efficient backbone network and replacing standard convolutions with less computationally intensive convolutions as much as possible.

III. LightOcean: A Lightweight and Efficient UAV Tracker

This section will provide a detailed description of the proposed LightOcean framework. As shown in Figure 2, LightOcean can be divided into four networks: a feature extraction network, a template update network, a feature fusion network, and a prediction head network. LightOcean accepts three images as input: a static template image I_z cropped from the first frame, a dynamic template image I_d continuously updated during the tracking process, and a search image I_x. The three input images are first passed through a parameter-sharing backbone network to generate corresponding features. Then the static and dynamic template features will be fed into a cross-attention module to enhance the target features in the dynamic template features directionally. The enhanced dynamic and static template features will be fused through a learnable parameter to generate more representative template features. Then, in the feature fusion network, the similarity map between the search feature and the template feature will be obtained through pixel-level inter-operation and go through a channel attention template to guide the tracker to pay more attention to the foreground information. The prediction head network consists of three branches: the classification and regression branches are used to classify and locate the target to obtain the tracking result, and the confidence branch is used to judge whether the tracking result of the current frame is stable. After every fixed frame, the tracker will decide whether to update the dynamic template based on the confidence score.

A. Feature extraction network

The feature extraction network has a critical impact on the performance and efficiency of the model. To choose a suitable backbone network, we first compare the performance of different backbone networks on edge devices and apply the best-performing ShuffleNetV2 to our tracker. More specifically, the original ShuffleNetV2 as a classification network has a downsampling step size of 32, which is not conducive to the precise positioning of the target. Therefore, we removed the last stage of the original ShuffleNetV2, kept only the first four stages for feature extraction, and modified the downsampling step size of the fourth stage to a unit step size to ensure the size of the output feature map. Furthermore, we add dilated convolutions in the fourth stage to increase the receptive field of the model.

The output of the backbone network is a feature map with a downsampling step size of 16 relatives to the input image, and its number of channels is 192. To facilitate the calculation of the subsequent feature fusion module and improve the efficiency, we pass the feature map through a superficial adjustment layer consisting of a layer of 1*1 convolution and a layer of normalization to obtain the final output result.

B. Template update network

The template features are cropped from the initial frame in the Siamese-based tracker. However, when the target changes significantly over time, the semantic information of the initial state will not be able to match the current object state accurately. This situation is prevalent with rapidly changing UAV tracking tables and is a significant cause of tracking mission failures or object drift. Therefore, we introduce an additional dynamic template feature, which dynamically

adapts to the object's appearance changes without increasing the model's computational complexity.

For the dynamic template feature F_d, we expect the tracker to focus on the object region instead of embedding all pixel features into the tracker equally, which suppresses the interference of background information. Inspired by [34], We design a cross-attention module to enhance the target information in dynamic template features by establishing the interrelationship between dynamic template and static template features. we first compute the similarity between each pixel of the static template feature F_d and the dynamic template feature F_d to generate a similarity matrix w, and then normalize w through a softmax layer, which can be expressed as:

$$w_{ij} = \frac{\exp\left(f(F_s^i, F_d^j)\right)}{\sum_{k \in F_s} \exp\left(f(F_s^k, F_d^j)\right)} \tag{1}$$

where w_{ij} represents the measure of the tracker's attention to the S_d^j pixel node in S_d according to the viewpoint of the S_t^i pixel node in S_t. Intuitively, the more similar the representations between two pixel nodes, the greater the correlation between them, and the more likely they are foreground. Therefore, we perform a matrix multiplication operation to update the dynamic template features, so that the tracker can adaptively focus on the foreground regions in F_d according to the suggestions of the template features. Finally, we perform a matrix concat operation on the fusion features and the search features to obtain a more powerful feature representation, and then use a 1×1 Conv-BN-ReLU block to reduce its dimension to reduce the complexity of the model, it can be denoted as:

$$F_d^{'} = concat(F_d, F_d w) \tag{2}$$

The initial static template feature provides a stable object model and accurate bounding box, which can always be used as the main reference information. Therefore, we linearly integrate the feature-enhanced dynamic template with the static template, so that the tracker can obtain stable spatial information from the static template features and also obtain temporal information from the dynamic template features:

$$F_z = \lambda F_s + (1 - \lambda) F_d^{'} \tag{3}$$

C. Feature fusion network

In the deep tracker of the siamese structure, the role of the cross-correlation operation is to fuse the template features and search features together in a specific way. Among them, up-channel correlation [1, 22] and depth-wise correlation [23, 38] are commonly used cross-correlation methods in the Siamese-base tracker. Both methods use the entire template feature as the convolution kernel and make the sliding window to calculate the similarity of the search feature. However, this simple global matching method is easily interfered with by background, affecting the tracker's robustness in complex and variable UAV scenarios. Recently, Alpha-Refine [37] proposed a new pixel-level correlation operation, which utilizes spatial pixels to calculate the similarity between template and search features to generate more delicate correlation features, which alleviates the interference of background noise. We follow this idea and enhance the critical channel feature representation with a channel attention mechanism.

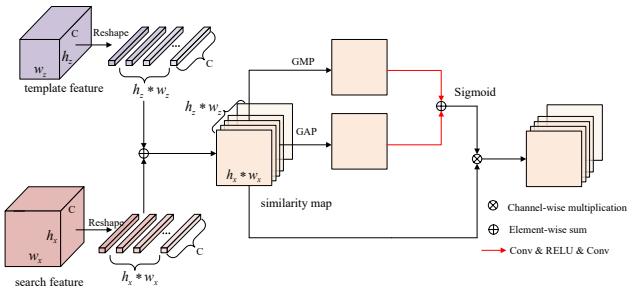


Fig. 3: Pixel-wise Correlation with Channel Attention Module

Pixel-wise Correlation Module. As illustrated in Figure 3, given the search features F_x, we first reshape it into a set of convolution kernels of size $1 * 1$ and length C along the spatial dimension, where the C represents the number of channels of the feature. This set S_x is represented as:

$$S_x = \{S_x^1, S_x^2, S_x^3, \cdots, S_x^p\} \tag{4}$$

Each item of the set represents the pixel point information of the search feature, there are $p = h_x * w_x$ items, and h_x and w_x are the template height and width of the search features F_x. Similarly, the same process is applied to template feature F_z to generate pixel information sets $S_z = \{S_z^1, S_z^2, S_z^3, \cdots, S_z^q\}$. Here q is the size of the template feature.

To obtain pixel-level fused features, we generate the similarity response r between each pixel in the set S_t and set S_s by the vector dot-product function f. Taking $S_t^i \in S_t$ and $S_s^j \in S_s$ as an example, the score r_{ij} is proportional to the similarity between the two pixels, which is expressed as:

$$r_{ij} = f(S_t^i, S_s^j) \tag{5}$$

Channel Attention Module. The similarity map obtained by pixel-level cross-correlation can accurately represent the similarity between each pixel of the template feature and the search feature and preserve the target's boundary information and scale information to the greatest extent. However, this method of local matching inevitably leads to the degradation of global communication, and it will be difficult for the tracker to distinguish the foreground and background regions. Therefore, inspired by the attention mechanism, we design a channel attention module to enhance target information and suppress background information with similar

features. Specifically, given a similarity map Rs, we first aggregate the spatial information in Rs using global max pooling(GMP) and global average pooling(GAP) sums. The generated max-pooled and average-pooled features are then fed into a parameter-sharing FPN network to explore the interrelationships between channels better. The similarity feature map Rc after channel attention enhancement can be obtained by the following formula:

$$Rc = \sigma(FFN(GAP(Rs)) + FFN(GMP(Rs))) * Rs \quad (6)$$

Here σ represents the Sigmoid function and $*$ denotes the channel-wise multiplication.

D. Prediction head network

Compared with anchor-based methods, anchor-free based methods have lower parameters and performance. Therefore, We design a lightweight prediction head containing three sub-networks to accomplish the task: a classification head, a regression head, and a confidence score head. The classification header is responsible for distinguishing foreground and background regions in the response maps and consists of two lightweight Conv-BN-ReLU [6] blocks and a $1 * 1$ linear standard Conv. The regression head is used to predict the distance from each pixel within the target object to the ground-truth bounding box and consists of four lightweight Conv-BN-ReLU [6] blocks and a $1 * 1$ linear standard Conv. To reduce the latency of converting the response maps from GPU to CPU, we adjusted the size of the response maps for classification and regression head from 25 to 16. Furthermore, during tracking, not all cases need to update the dynamic template frame. When the target has been completely occluded or out of view, the cropped template is no longer reliable. For simplicity, we consider the current tracking state to be credible as long as the search image contains the target. Therefore, we add a confidence score header to judge whether the current tracking state is reliable, consisting of a three-layer perceptron and a sigmoid activation.

IV. Experiments

In this section, we first introduce the training and testing details of the model and the setting of hyperparameters. Then, we evaluate the tracker accuracy on three public UAV tracking benchmarks, including UAV123@10FPS[30], UAV20L[30], DTB70 [24], and compare with other state-of-the-art UAV trackers to demonstrate the superiority of our proposed method. At the same time, edge-embedded devices are tested to verify their practicability in real scenarios. Finally, comprehensive ablation experiments analyze the impact of individual parts on the tracker.

A. Implementation details

Our proposed model is implemented using Python and PyTorch. The whole experiment is performed on a server with 4 NVIDIA A100-PCIe-40GB. To obtain the best experimental results, we selected three edge-friendly backbone networks, AlexNet, ShuffleNetV2, and MobileNetV2, for training. All backbone networks have a downsampling stride and are pretrained on imageNet.

Training. The proposed tracker is trained on COCO [26], LaSOT [12], GOT-10K [17], VisDrone [39] datasets, each minimum training unit consists of two template images and a search image. The size of the search and template images are 255*255 and 127*127, respectively. We use SGD [21] for synchronization optimization on four devices with a batch size of 256. For more training details, the learning rate decays exponentially from 1×10^{-3} to 5×10^{-3}. Weight decay and momentum are set to 10^{-4} and 0.9, respectively. The training process is divided into two stages to decouple the localization and classification tasks.

In stage 1, we train other parameters except for the confidence score header. Static template frames and dynamic template frames are randomly extracted from the same video sequence, and the interval between them is less than 200. The search frame is randomly selected from the range of less than 100 frames near the dynamic template frame to train the localization ability of the model. Stage 1 trains a total of 50 epochs, and each epoch has $6 * 10^6$ training units. We use the IoU loss and the binary cross-entropy loss to jointly train the regression and classification networks, and the total loss is defined as:

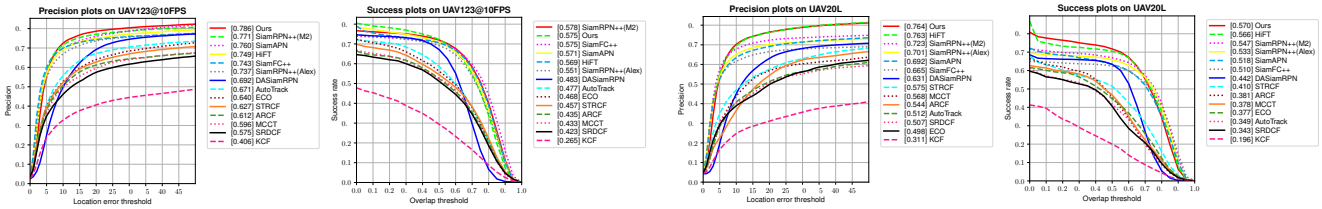
$$L_{s1} = L_{cls} + L_{reg} \quad (7)$$

In stage 2, we freeze other parameters except the confidence score header to avoid affecting the localization ability of the model. In this stage, 20 epochs are trained, each with $6 * 10^4$ training units. The extraction process for positive sample search images is the same as in stage 1. For negative sample search images, we sample from the same video as much as possible; otherwise, we randomly sample a frame from other video sequences as a negative sample. The binary cross-entropy loss is defined as:

$$L_{s2} = y_i \log P_i + (1 - y_i) \log (1 - P_i) \quad (8)$$

Here y_i is the ground truth label, and P_i is the predicted confidence.

Testing. During inference, we follow a similar tracking process to Ocean[38]. Static and dynamic template features are initialized in the first frame, and then a cropped search image is fed into the network. Generate corresponding bounding boxes and confidence scores from the classification and regression response maps. When the update interval $t = 150$ is reached, if the confidence score is higher than the threshold $p = 0.5$, the dynamic template feature is updated; otherwise, the confidence score will be checked every gradually increasing interval until the confidence score is higher than the threshold $p = 0.5$. The updated dynamic template features are cropped from the search images and fed into the backbone network.



(a) UAV123@10FPS (b) UAV20L

Fig. 4: Overall performance of all trackers on (a) UAV123@10FPS, (b) UAV20L.

TABLE I: Comparison results on the DTB70 benchmark. The best performances are highlighted in red, green and blue.

Tracker	SiamRPN++ (AlexNet)	SiamRPN++ (MobileNetV2)	SiamAPN (AlexNet)	SiamFC++ (AlexNet)	HiFT (AlexNet)	AutoTrack	ECO	STRCF	ARCF	MCCT	SRDCF	KCF	LightOcean (ours)
Prec.	0.793	0.785	0.784	0.785	0.802	0.716	0.635	0.649	0.694	0.604	0.512	0.468	0.795
Succ.	0.586	0.593	0.586	0.584	0.594	0.478	0.448	0.437	0.472	0.405	0.363	0.280	0.594

TABLE II: Comparison with deep trackers. The best performances are highlighted in red, green and blue.

Tracker	TransT (Res50)	ATOM (Res18)	PrDIMP (Res18)	Ocean (Res50)	SiamFC++ (GoogleNet)	SiamGAT (GoogleNet)	SiamMask (Res50)	SiamRPN++ (Res50)	SiamBAN (Res50)	SiamCAR (Res50)	LightOcean (ours)
Pre.	0.848	0.857	0.837	0.759	0.759	0.788	0.788	0.784	0.770	0.789	0.785
Suc.	0.651	0.648	0.644	0.539	0.589	0.602	0.590	0.594	0.585	0.596	0.575

B. Comparison with Light-Weight Trackers

In this section, We compare our proposed tracker with existing 12 lightweight trackers on 3 common UAV benchmarks (UAV123@10FPS [30], UAV20L [30], DTB70[30]). There are 6 Siamese-base trackers (SiamFC++ [36], SiamFC [1], SiamRPN++ [23], HiFT [3]), and 5 DCF-base trackers (ECO[11], KCF [15], ARCF [18], AutoTrack[25], BACF[20]). To be fair, all Siamese-based trackers use lightweight backbone networks such as AlexNet and MobileNet.

UAV123@10FPS. UAV123@10FPS [30] contains 123 low-altitude drone sequences, with a total frame count of more than 112k, shooting many scenes and targets from various angles in the air. Since the sampling rate of UAV123@10FPS is 10fps, the interval between frames becomes more extensive, and the motion of the target changes more drastically, which poses a considerable challenge to the robustness of the tracker. The results in Figure 4(a) show that LightOcean achieves competitive performance with a precision score and an success score of 0.786 and 0.578, respectively, which are higher than most current trackers.

UAV20L. UAV20L consists of 20 long-term tracking sequences with more than 58K frames, which are used to evaluate long-term tracking scenarios common in real-world aerial tracking. As shown in Figure 4(b), attributing to the introduction of dynamic template features, LightOcean can adapt to changes in the appearance of targets in long-distance tracking, thereby achieving more excellent tracking performance. Specifically, LightOcean achieves a precision score of 0.764 and an success score of 0.570, outperforming the recent SOTA aerial tracker HiFT and improving the score by 5.6% over SiamRPN++.

DTB70. DTB70 is also a widely used UAV tracking dataset, consisting of 70 video sequences from aerial viewpoints with high diversity, containing a large number of camera-moving scenes with rapidly changing motions. As shown in Table I, the success score of our tracker is 0.594, which is comparable to the state-of-the-art HiFT tracker and outperforms SiamRPN++ and SiamFC++. In summary, LightOcean is able to achieve better robust results in various UAV tracking scenarios compared to other state-of-the-art tracking, confirming the effectiveness of our proposed method.

TABLE III: Attribute-based evaluation of the LightOcean and other 12 SOTA tracker.The best performances are highlighted in red, green and blue respectively.

Attributes	Out of View		Similar Object		Scale Variation		Full Occlusion	
Tracker	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.	Prec.	Succ.
SiamRPN++_A	0.642	0.474	0.730	0.523	0.703	0.522	0.521	0.313
SiamRPN++_M	0.720	0.527	0.731	0.539	0.742	0.553	0.574	0.363
SiamAPN	0.688	0.504	0.699	0.512	0.732	0.548	0.530	0.320
SiamFC++	0.677	0.522	0.727	0.544	0.718	0.553	0.521	0.335
HiFT	0.685	0.514	0.673	0.490	0.725	0.549	0.525	0.323
AutoTrack	0.554	0.406	0.664	0.462	0.629	0.443	0.444	0.243
ECO	0.535	0.399	0.655	0.478	0.594	0.430	0.465	0.256
STRCF	0.523	0.389	0.630	0.455	0.580	0.419	0.426	0.232
ARCF	0.449	0.330	0.657	0.445	0.570	0.399	0.392	0.200
MCCT	0.493	.0365	0.627	0.451	0.547	0.396	0.421	0.236
SRDCF	0.492	0.389	0.585	0.421	0.531	0.390	0.418	0.229
KCF	0.309	0.222	0.453	0.278	0.374	0.238	0.281	0.135
LightOcean	0.750	0.543	0.767	0.542	0.769	0.558	0.627	0.377

Attribute-based performance. To further evaluate the performance of LightOcean in different tracking scenar-

ios, this paper selects four attribute-based tracking scenarios for testing, which are widely used in actual UAV tracking. As shown in Table III, LightOcean ranks first in both success and accuracy rates compared to the other 12 state-of-the-art trackers. Specifically, since our tracker applies an additional dynamic template feature, the latest state information of the object can be dynamically obtained during the tracking process. Therefore, our tracker can effectively adapt to complex environmental changes and achieve more robust results in scale and care-changing scenes. At the same time, benefiting from the pixel-level feature fusion network, LightOcean can more effectively distinguish and fuse the background information before and after. Therefore, the performance of LightOcean in occlusion and out-of-view scenarios is also significantly improved.

C. Comparison with Deep Trackers

To further verify the robustness of our proposed tracker, we compare it with trackers applying deeper backbone networks. The deep trackers involved in the comparison include, SiamRPN++ [23], TransT [4], PrDIMP [7], ATOM [8], Ocean [38], SiamFC++ [36], SiamGAT [13], SiamMask [33], SiamBAN [5], SiamCAR [14]. The results are shown in Table II. Although our proposed tracker uses the lightweight network ShuffleNet as the feature extraction network, it still achieves competitive performance compared with trackers using hierarchical backbone networks. On UAV123@10FPS, LightOcean differs from SiamGAT by less than one point in accuracy, which is better than the original algorithm Ocean. Therefore, the above results further verify the effectiveness of the method proposed in this paper.

D. Comparison on edge platforms

To further verify the effectiveness of our proposed tracker on real edge applications, we compare the model's inference latency, power consumption, and memory footprint with other SOTA deep Siamese trackers on a typical embedded edge device, Jetson Nano. Jetson Nano is an embedded CPU-GPU heterogeneous device produced by NVIDIA, which is widely used in various edge applications due to its low cost and high-cost performance. To avoid other hardware and software factors affecting the evaluation results, all environments of the edge platform are consistent and only perform tracking tasks. Evaluation environment on Jetson Nano: Ubuntu 18.04, JetPack 4.4.1, CUDA 10.2, Pytorch 1.6.0. The overall comparison results are shown in Figure 5.

Latency. The latency of the model is calculated by dividing the total latency of inferring a video sequence by the total number of frames in the video, regardless of the time the model takes to load and warm up. As shown in Figure 5, LightOcean has lower latency than other SOTA trackers and can run in real-time at a speed of 10.8FPS on Nano. It is nearly 6X faster than TransT with the best performance, 5X faster than the original algorithm Ocean, and more than twice that of SiamRPN++ (MobileNetV2), which is also a lightweight network.

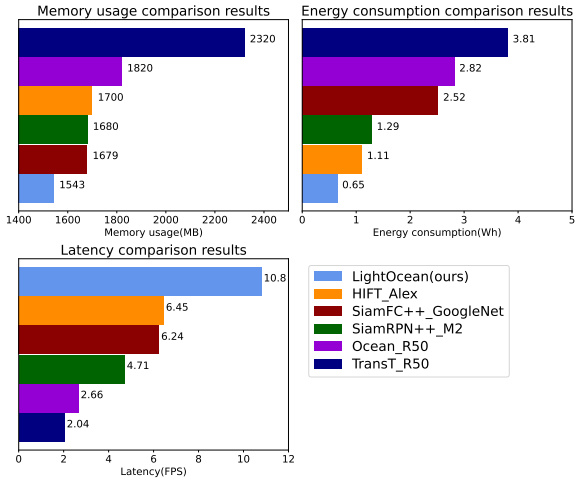


Fig. 5: Memory usage, energy consumption and latency comparison results on Jetson Nano

Energy consumption. The energy consumption of the model is measured by an external watt-hour meter on the edge device. Similarly, we do not consider the cost of model loading and warm-up. It can be observed that LightOcean is energy efficient, and it only consumes 0.65Wh of power to infer a video sequence of 3000 frames of pictures on Jetson Nano. Compared with other trackers, the average energy consumption of LightOcean is one-sixth of TransT, one-fifth of Ocean, and half of SiamRPN++. Being more energy efficient makes LightOcean trackers more competitive on UAV platforms with limited battery capacity.

Memory usage. The memory usage represents the maximum memory occupied by the tracker during the running process, which we measure by the jtop tools. The results show that LightOcean has a maximum memory usage of 1.6GB during model inference, which is 30% less memory than TransT and 10% less memory than Ocean. Although LightOcean needs to save additional dynamic template features in the inference process, its memory usage is still lower than that of lightweight network, which shows that our proposed method is more memory-efficient. Compared with other SOTA trackers, LightOcean is more edge-friendly, providing lower inference latency, energy consumption, and memory usage, verifying its effectiveness on real edge devices.

E. Ablation study

In this section, we verify the effectiveness of our proposed method in terms of the backbone network, cross-correlation operation, and dynamic template update. We evaluate UAV123@10FPS on a typical jetson nano to explore the impact of different components on tracking performance and efficiency. As shown in the Table IV, the baseline model has the same network architecture as Ocean. We first analyze the performance impact of different lightweight

backbone networks on the tracker. We noticed that the inference speed of using ShuffleNetV2 as the backbone network is 10.1 FPS, which is four times faster than baseline, 10% faster than MobileNetV2, and 50% faster than AlexNet. At the same time, ShuffleNetV2 is comparable to MobileNetV2 in terms of accuracy, better than Alexnet, and saves more memory and energy. By replacing the depth cross-correlation module with a pixel-level feature fusion module, the interference of background information is reduced, and the accuracy is improved by 5%. At the same time, since the classification branch and regression branch share the parameters of the same feature fusion module, the tracker's inference delay and resource consumption are further reduced. In addition, by introducing a credible, dynamic template feature, the expressiveness of the template feature is finally improved, and 3.1% increases the accuracy with almost no increase in model computation.

TABLE IV: Ablation study on UAV123@10FPS.

	Pre.	#Latency(FPS)	#Mem(MB)	#Power(Wh)
BaseLine	0.759	2.6	1820	2.82
AlexNet	0.658	6.5	1480	0.87
MobileNetV2	0.708	8.8	1617	0.94
ShuffleNetV2	0.726	10.1	1463	0.72
PixCorr	0.762	11.1	1423	0.63
OnlineUpdate	0.786	10.8	1531	0.65

V. Conclusion

This paper proposes a novel lightweight tracker, `LightOcean`, based on the Siamese network structure for airborne real-time tracking. This paper designs a dynamic template update module and a pixel-level cross-correlation module to improve tracking performance without compromising tracking efficiency. Combined with a lightweight and efficient feature extraction network and prediction head, the proposed `LightOcean` achieves a balance of tracking accuracy and efficiency. This paper evaluated the performance on a typical embedded edge platform, Jetson-nano, with three typical UAV benchmarks, UAV123@10FPS, UAV20L, and DTB70. `LightOcean` presents lower energy consumption and memory footprint than other trackers. The tracking speed is 4x faster than the state-of-the-art tracker, Ocean, while the energy consumption and memory usage are reduced by 80% and 12%. Consequently, we believe that our work can boost the development of UAV tracking-related applications.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62162067 and 62101480, in part by the Yunnan Province Science Foundation under Grant No.202005AC160007, No.202001BB050076, Research and Application of Object detection based on Artificial Intelligence, in part by the China Postdoctoral Science Foundation under Grant No.2021M693227, and in part of Zhejiang Lab under Grant No.2020KE0AB02.

References

[1] Luca Bertinetto et al. "Fully-convolutional siamese networks for object tracking". In: *European conference on computer vision*. Springer. 2016, pp. 850–865.

[2] David S Bolme et al. "Visual object tracking using adaptive correlation filters". In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2544–2550.

[3] Ziang Cao et al. "HiFT: Hierarchical Feature Transformer for Aerial Tracking". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 15457–15466.

[4] Xin Chen et al. "Transformer tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8126–8135.

[5] Zedu Chen et al. "Siamese box adaptive network for visual tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6668–6677.

[6] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[7] Martin Danelljan, Luc Van Gool, and Radu Timofte. "Probabilistic regression for visual tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 7183–7192.

[8] Martin Danelljan et al. "Atom: Accurate tracking by overlap maximization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4660–4669.

[9] Martin Danelljan et al. "Convolutional features for correlation filter based visual tracking". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2015, pp. 58–66.

[10] Martin Danelljan et al. "Discriminative scale space tracking". In: *IEEE transactions on pattern analysis and machine intelligence* 39.8 (2016), pp. 1561–1575.

[11] Martin Danelljan et al. "Eco: Efficient convolution operators for tracking". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6638–6646.

[12] Heng Fan et al. "Lasot: A high-quality benchmark for large-scale single object tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 5374–5383.

[13] Dongyan Guo et al. "Graph attention tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 9543–9552.

[14] Dongyan Guo et al. “SiamCAR: Siamese fully convolutional classification and regression for visual tracking”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6269–6277.

[15] João F Henriques et al. “High-speed tracking with kernelized correlation filters”. In: *IEEE transactions on pattern analysis and machine intelligence* 37.3 (2014), pp. 583–596.

[16] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[17] Lianghua Huang, Xin Zhao, and Kaiqi Huang. “Got-10k: A large high-diversity benchmark for generic object tracking in the wild”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.5 (2019), pp. 1562–1577.

[18] Ziyuan Huang et al. “Learning aberrance repressed correlation filters for real-time UAV tracking”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 2891–2900.

[19] Forrest N Iandola et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[20] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. “Learning background-aware correlation filters for visual tracking”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 1135–1143.

[21] Yann LeCun et al. “Backpropagation applied to handwritten zip code recognition”. In: *Neural computation* 1.4 (1989), pp. 541–551.

[22] Bo Li et al. “High performance visual tracking with siamese region proposal network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8971–8980.

[23] Bo Li et al. “Siamrpn++: Evolution of siamese visual tracking with very deep networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4282–4291.

[24] Siyi Li and Dit-Yan Yeung. “Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models”. In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[25] Yiming Li et al. “AutoTrack: Towards high-performance visual tracking for UAV with automatic spatio-temporal regularization”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11923–11932.

[26] Tsung-Yi Lin et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[27] Shaohua Liu et al. “Vehicle tracking by detection in UAV aerial video”. In: *Science China Information Sciences* 62.2 (2019), pp. 1–3.

[28] Ningning Ma et al. “Shufflenet v2: Practical guidelines for efficient cnn architecture design”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 116–131.

[29] Ioannis Mademlis et al. “High-level multiple-UAV cinematography tools for covering outdoor events”. In: *IEEE Transactions on Broadcasting* 65.3 (2019), pp. 627–635.

[30] Matthias Mueller, Neil Smith, and Bernard Ghanem. “A benchmark and simulator for uav tracking”. In: *European conference on computer vision*. Springer. 2016, pp. 445–461.

[31] Horst Possegger, Thomas Mauthner, and Horst Bischof. “In defense of color-based model-free tracking”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 2113–2120.

[32] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[33] Qiang Wang et al. “Fast online object tracking and segmentation: A unifying approach”. In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2019, pp. 1328–1338.

[34] Xiaolong Wang et al. “Non-local neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 7794–7803.

[35] Sean Ward et al. “Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision”. In: *2016 IEEE aerospace conference*. IEEE. 2016, pp. 1–8.

[36] Yinda Xu et al. “Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 12549–12556.

[37] Bin Yan et al. “Alpha-refine: Boosting tracking performance by precise bounding box estimation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 5289–5298.

[38] Zhipeng Zhang et al. “Ocean: Object-aware anchor-free tracking”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 771–787.

[39] Pengfei Zhu et al. “Visdrone-vid2019: The vision meets drone object detection in video challenge results”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2019, pp. 0–0.