2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

EFTrack: A Lightweight Siamese Network for Aerial Object Tracking

Wenqi Zhang[1], Yuan Yao[1], Xincheng Liu[1], Kai Kou[1] and Gang Yang[1]

Abstract— **Visual object tracking is a very important task for unmanned aerial vehicle (UAV). Limited resources of UAV lead to strong demand for efficient and robust trackers. In recent years, deep learning-based trackers, especially, siamese trackers achieve very impressive results. Though siamese trackers can run a relatively fast speed on the high-end GPU, they are becoming heavier and heavier which restricts them to be deployed on UAV platform. In this work, we propose a lightweight aerial tracker based on the siamese network. We use EfficientNet as the backbone, which has less parameters and stronger feature extract ability compared with ResNet-50. After a pixel-wise correlation, a classification branch and a regression branch are applied to predict the front/back score and offset of the target without the predefined anchor. The results show that our tracker works efficiently and achieves impressive performance on UAV tracking datasets. In addition, the real-world test shows that it runs effectively on the Nvidia Jetson NX deployed on DJI UAV.**

I. INTRODUCTION

Due to the lower cost and higher flexibility, the unmanned aerial vehicle (UAV) is widely used in various missions, and the market size continues to grow yearly. More specifically, good processing performance and rich payloads make it an ideal tool in military and civilian surveillance system. Compared with humans, UAV can easily, safely and efficiently perform tasks including but not limited to disaster rescue, power line inspection, and traffic monitoring[1]. Object tracking is one of the hot spots and fundamental problems in computer vision research. As a branch of object tracking, aerial tracking has received extensive attention due to its broad application prospects[2]. The task of aerial tracking is to give the position of the object in the initial frame, and then predict the position of the object in subsequent frames. A significant difference between aerial tracking and other tracking tasks is that the limited payload capacity and fast flight speed make it require real-time speed and low resource consumption. Besides, the object tracking tasks in the UAV remote sensing also face a number of challenges, such as image degradation, variable object intensity, small object size, background complexity, motion blur *etc*[3]. Obviously, an efficient and effective aerial tracker needs to be developed.

Generally, object tracking methods are mainly divided into two categories: correlation filter-based online trackers and deep learning-based offline trackers. In contrast, The former method is CPU-friendly[4][5] while the latter approach requires GPU[6]. In the past few years, methods based on discriminative correlation filtering (DCF) have been widely used on aerial platforms[7]. The most important feature and amazing highlight of DCF-based methods is the transfer of calculation into Fourier domain, which significantly increases the speed of DCF-based trackers[8]. Among them, most trackers can reach 30 frames per second (FPS) on a single CPU platform, which ideally meets the need of UAV. The most representative DCF-based tracker is KCF[9], which trains an object detector during the tracking process and uses the object detector to detect whether there is an object at the predicted position of the next frame. On the one hand, methods based on correlation filtering have the advantages of low cost and high efficiency. On the other hand, they cannot achieve good results in some challenging scenarios, such as occlusion, deformation, and motion blur, which still have a clear gap compared with offline trackers based on deep learning. However, the latter suffers from inefficiencies and needs to achieve a satisfactory balance.

Among the deep learning-based trackers, siamese tracker plays a significant role in object tracking. The pioneer works of siamese tracker are SINT[10] and SiamFC[11]. which utilize siamese network to learn the similarity of the object and search area and make the tracking problem a search problem of the target in the hole image. Based on this, a large number of follow-up siamese trackers have been proposed and achieved tremendous performance. SiamRPN[12] introduces region proposal network into tracking and uses anchor-based approach to predict the location of object. Then following studies are mainly divided into two parts: designing more powerful backbone network[13][14] and proposing more effective proposal network[15]. In the meanwhile, anchor-free methods have been proposed, such as SiamBAN[16], SiamCAR[17], Ocean[18], SiamFC++[19], *etc.* which have been well applied. These efforts have indeed achieved remarkable results. However, these models are becoming increasingly heavy and expensive compared with early methods and cannot be deployed on embedded systems. Table I shows the difference in hardware metrics between desktop GPUs and embedded devices.

TABLE I
COMPARISON OF NVIDIA JETSON XAVIER NX AND RTX TITAN

Device	RTX TITAN	Jetson Xavier NX
CUDA Cores	4608	384
Tensor Core	576	48
Total Video Memory	24GB	8GB
Memory Interface	384-bit	128-bit
Total Memory Bandwidth	672 GB/s	59.7 GB/s
Thermal Design Power	280Watts	20Watts

In this work, a lightweight aerial tracker based on siamese

[1] All authors are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China yaoyuan@nwpu.edu.cn

network is proposed. An anchor-free method is adopted for the generation of the location of the object. In more detail, the framework consists of a siamese network backbone and a fully convolutional prediction head. Unlike most of the siamese trackers, the backbone chooses EfficientNet[20], instead of modified ResNet-50[13]. The predict head has two branches, one for classification and the other for regression. The classification branch predicts the foreground-background category score, and the regression branch predicts the relative offset from the four sizes of bounding boxes. With a small size, the model only has 1.08M parameters and 408.66 MFlops and can be deployed on embedding system like NVIDIA Jetson Xavier NX, a widely used onboard computer that can run deep neural networks.

The contributions of this work are as follows:

- A simple and efficient lightweight aerial tracker based on siamese network is proposed, which can significantly improve the usability and generality of siamese tracker in aerial scenarios.
- Extensive experiments show that the proposed method is effective and efficient. Competitive performance is observed based on three challenging aerial tracking datasets. According to the real-world experiments on NVIDIA Jetson Xavier NX, this tracker still maintains robustness for aerial tracking, with a running speed exceeding 20 FPS.

II. RELATED WORKS

Generally, discriminative correlation filter (DCF)-based trackers are widely used in the tracking of UAV[7], due to the advantages of high speed and outstanding performance. DCF-based approach takes the object tracking algorithm into a new level. Robustness and accuracy are significantly improved with high processing speed. In recent years, siamese-based trackers have received increasing attention due to their state-of-the-art performance. Therefore, efficient and effective siamese-based tracker is an ideal choice for aerial tracking.

With the development of convolutional neural network (CNN), deep learning shows strong power in computer vision. SINT[10] first introduce siamese network into object tracking. It transfers tracking into a matching problem. SiamFC[11] proposed an end-to-end fully convolutional network approach to learn the similarity of the template and search area. It runs efficiently at 86 FPS due to its simple structure and no runtime updates. DSiam[21] learns feature transformation to handle the object variation and inhibiting background. However, these methods require a multi-scale testing to detect scale variation. Also, they cannot handle aspect ratio changes. In order to get more precisely bounding boxes, SiamRPN[12], inspired by Faster R-CNN[22], introduce region proposal network (RPN) into object tracking. Tracking issues are considered one-time detections. Then SiamRPN++[13], SiamDW[14] and SiamMask[23] remove the influence of factor such as padding in different ways so that siamese network benefit from deep network such as ResNet-50[24], MobileNet[25] or deeper networks instead of AlexNet[26]. Besides these offline trackers, online trackers like PrDiMP[27] achieve better results due to its online update strategy. Anchor-based methods need to carefully design and fix the parameter of anchor boxes to address the scale and aspect ratio of the objects, In other words, they require a lot of tricks to achieve good results. Anchor-free methods are proposed to solve these problems. Ocean[18] builds the object-aware anchor-free network. SiamBAN[16] propose a box adaptive network (BAN) that makes use of the expressive power of the fully-convolutional network (FCN). SiamCAR[17] decomposes the visual object tracking into two sub-problems: pixel category classification and object bounding box regression at this pixel. SiamFC++[19] proposed a set of generic object tracker design guidelines. Briefly, the tracking task is viewed as a combination of a classification task and an estimation task, where the classification task determines the location of the object, and the estimation task obtains the state of the object.

To meet the needs of UAV tracking, some studies propose efficient tracking methods. SiamAPN[28] used anchor proposal network to precisely generate anchor. SiamAPN++[29] conducted a special attention aggregation network. These adaptive anchor generation methods improve the robustness and generalization in some challenging scenarios and decrease the number of anchor, thereby reducing computing stress. LightTrack[30] designed a lightweight and efficient object tracker using neural architecture search, which bridges the gap between academia and industry.

III. PROPOSED METHOD

In this section, the proposed aerial tracker will be introduced in detail. As shown in Fig. 1, the aerial tracker consists of three sub-networks, namely the feature extraction network, the correlation network, and the prediction head network.

A. Feature Extraction Network

Following the architecture of siamese tracker, feature extraction network has two branches that shared the same architecture and weight. The proposed model takes an image pair as input, namely, an exemplar image and a search image. The exemplar image represents the target, usually the crop of the target in the initial frame. The search image represents the search area in subsequent frames. EfficientNet-B0 is used as the backbone, instead of ResNet-50. Compared with ResNet-50 EfficientNet has fewer parameters and FLOPs, as shown in Table II. Appropriate network stride plays an essential role in the tracking task, as the tracker requires detailed spatial information to make predictions. Most Siamese trackers reduce the backbone network stride in the last few layers and use atrous convolution to increase the receptive field, we simply use the first 6 stages of EfficientNet-B0 shown in TABLE III as our backbone.

B. Correlation Network

Generally, Siamese trackers use naive cross correlation or depth-wise cross correlation to aggregate features of the exemplar image and search image. The above two correlation

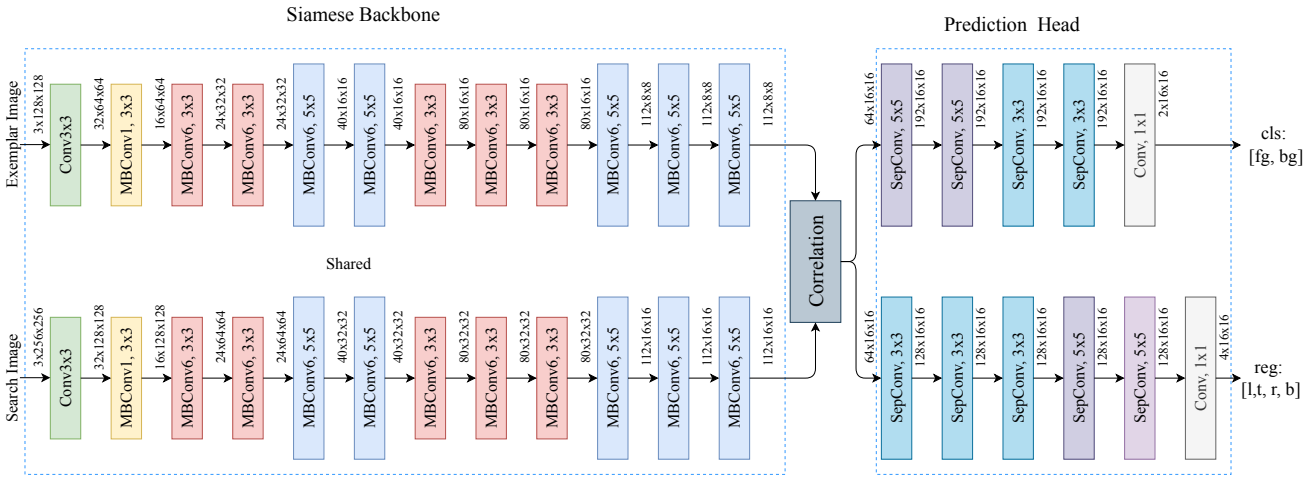


Fig. 1. Network Structure

TABLE II
PARAMETER AND FLOPS COMPARED BETWEEN EFFICIENTNET-B0 AND RESNET-50

Model	Top1-acc	Top5-acc	Parameters	FLOPs
EfficientNet-B0	76.3%	93.2%	5.3M	0.39B
ResNet-50	76.0%	93.0%	26M(4.9×)	4.1B(11×)

TABLE III
STRUCTURE OF EFFICIENTNET-B0

Stage	Operator	Resolution	Channels	Layers
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv1, k3x3	112 × 112	24	2
4	MBConv1, k5x5	56 × 56	40	2
5	MBConv1, k3x3	28 × 28	80	3
6	MBConv1, k5x5	28 × 28	112	3
7	MBConv1, k5x5	14 × 14	192	4
8	MBConv1, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

methods use the whole exemplar features as the kernel to correlate with the search region features, and generate the response map. Different from previous methods using multi-scale features, this method only uses single-scale features. In this work we adopt pixel-wise cross correlation[30] as our correlation network. $K \in \mathbb{R}^{C\times H_z\times W_z}$ is used to denote the features of exemplar and $S \in \mathbb{R}^{C\times H_x\times W_x}$ for the feature of search region. Pixel-wise cross correlation decomposes the exemplar features into H_zW_z small kernels $K_j \in \mathbb{R}^{C\times 1\times 1}$ and then correlate with the features of search region to generate the correlation maps $C \in \mathbb{R}^{H_zW_z\times H_x\times W_x}$. The process can be described as:

$$C = \{C_j | C_j = K_j * S\} \quad j \in \{1, \ldots, H_z * W_z\} \tag{1}$$

where $*$ denotes naive correlation. Pixel-wise correlation ensures that each correlation map encodes information of a local region on the target and avoids blurring features. After the correlation a channel attention module is used to adjust the weights of each channel, which can enhance features distinguishably.

C. Prediction Head

As shown in Fig. 1, the prediction head has two branches, a classification branch and a regression branch. Both branches receive the features from response map. The feature maps are processed using multiple SepConv blocks, and the prediction head uses 1x1 Conv layer to output prediction result. SepConv consists of two convolution layers, one BatchNorm layer and one ReLU6 layer, as shown in Fig. 2. By design, the classification branch needs to output two channels for foreground-background classification, and the regression head needs to output four channels for the prediction of bounding box's offset.

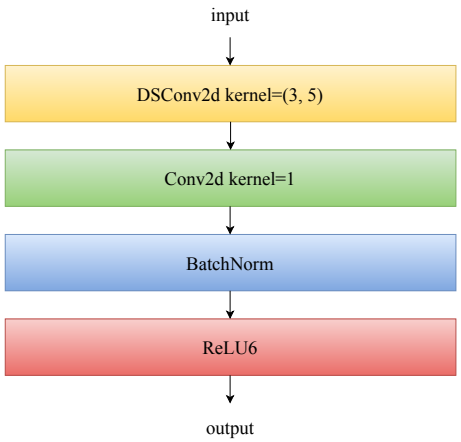


Fig. 2. SepConv Structure

The two prediction branches have different numbers of network blocks, including four blocks in classification branch and five blocks in regression branch, because the regression branch needs more information to generate bounding boxes.

The SepConv block has two different kernel sizes of 3 and 5, with padding operations of 1 or 2 to maintain the size of feature map. In general, the larger the convolution kernel, the larger the receptive field, the more information can be seen in the image, and the better the global features obtained. In the classification branch, two SepConv blocks with the kernel of 5 are first adopted to better obtain the information of the object. Then, two SepConv blocks with the kernel of 3 are adopted to enable better aggregation of information about the object. The channel of classification branch is set to 192. For the regression branch, three SepConv blocks with the kernel of 3 are used first, and then two SepConv blocks with the kernel of 5 are used, because generating prediction frames requires more confidence in local details. The channel of regression branch is set to 128. After these treatment, a convolution layer is used to change the network channels to fit the output shape.

The output of the prediction head is shown in Fig. 3. Each of the point on the classification map $P^{cls}_{w\times h\times 2}$ or the regression map $P^{reg}_{w\times h\times 4}$ can be mapped to the input search image. Assuming that there is a point marked (i,j) on classification map, its corresponding coordinates on the search image can be expressed as:

$$\begin{aligned} P_i &= w - (\left\lfloor \frac{w}{2} \right\rfloor - i) \times s \\ P_j &= h - (\left\lfloor \frac{h}{2} \right\rfloor - j) \times s \end{aligned} \tag{1}$$

where w and h represent the width and height of the search image, and s denotes the total strides of the feature extraction network. For the regression map, he offset of the bounding box is calculated directly. Since the regression target is a positive real number, $exp(x)$ is applied at the last stage of the regression branch, to make sure the output $d \in (0,+\infty)$.

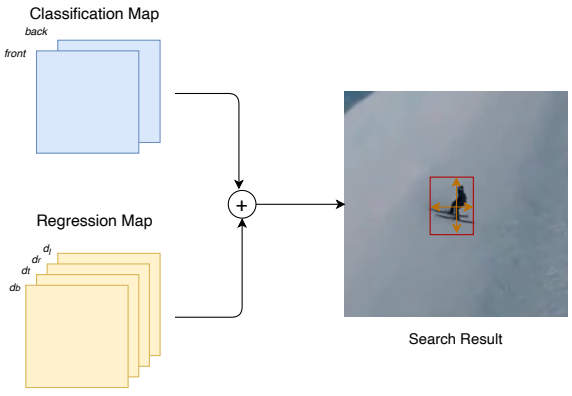


Fig. 3. Illustrations of Prediction Head

D. Ground-truth and Loss

Following the design of SiamBAN [16], let g_w, g_h, (g_{x1}, g_{y1}), (g_{xc}, g_{yc}), (g_{x2}, g_{y2}) represent the width, height, top-left point, center point and bottom-right point. the point falls in

$$\frac{(p_i - g_{xc})^2}{(\frac{g_w}{4})^2} + \frac{(p_j - g_{yc})^2}{(\frac{g_h}{4})^2} = 1 \tag{2}$$

signed with a positive label, and the point falls out of

$$\frac{(p_i - g_{xc})^2}{(\frac{g_w}{2})^2} + \frac{(p_j - g_{yc})^2}{(\frac{g_h}{2})^2} = 1 \tag{3}$$

signed with a negative label. For a point (P_i, P_j) with positive label, the regression targets are:

$$\begin{aligned} d_l &= P_i - g_{x1} \\ d_t &= P_j - g_{y1} \\ d_r &= g_{x2} - P_i \\ d_b &= g_{y2} - P_j \end{aligned} \tag{4}$$

The loss function is defined as follows:

$$\mathscr{L}_{total} = \alpha \mathscr{L}_{cls} + \beta \mathscr{L}_{reg} \tag{5}$$

where the $\mathscr{L}_{cls}$ is cross entropy loss and $\mathscr{L}_{reg}$ is IoU loss, and the coefficient is simply set to 1.

IV. EXPERIMENTAL RESULTS

A. Training And Inference

1) Training details: Our backbone network is initialized with the weight pre-trained on ImageNet for image classification, which has proven to be a very important step for tracking[30]. We used COCO[31], ImageNet DET[32], ImageNet VID[32], YouTuBe-BoundingBoxes Dataset[33], LaSOT[34] and GOK-10K[35] as our training datasets. In both training and testing, we used single scale images with 128 pixels for exemplar image and 256 pixels for searching image.

Stochastic gradient descent (SGD) is used to train our trackers. The training machine has dual RTX TITAN GPUs. we use synchronized SGD over two GPUs with a total of 32 pairs per minibatch (16 pairs per GPU). We trained the network with 50 epochs in total, for the first 10 epochs, the backbone is frozen. We use a warmup learning rate of 0.001 to 0.005 in the first 5 epochs. For the rest epochs, the network is trained with learning rate exponentially decayed from 0.005 to 0.00005 while in the last 10 epochs fine-tuned the backbone network with one-tenth of the current learning rate. Weight decay of 0.0001 and momentum of 0.9 are used.

2) Inference: During the inference, the exemplar frame's feature is initiated by the first input frame, the features of exemplar are stored and do not update during the entire tracking process. For the following frames, we crop the search image and extract the feature based on target position in previous frame, and predict the location and the offset of the target based on classification map and regression map. The bounding box can be calculated by the formula:

$$\begin{aligned} P_{x1} &= P_i - d^{reg}_l \\ P_{y1} &= P_j - d^{reg}_t \\ P_{x2} &= P_i + d^{reg}_r \\ P_{y2} &= P_j + d^{reg}_b \end{aligned} \tag{6}$$

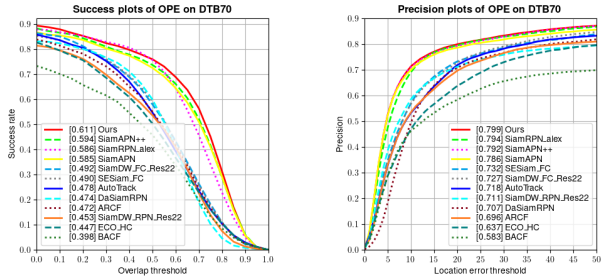
After generating the prediction bounding box, a cosine window and scale change penalty is used to smooth the output, which is essential for object tracking.

B. Evaluation

1) Evaluation method: The experiments are based on one-pass evaluation (OPE)[36]. Success and precision are used to measure the performance of the tracker. The precision of the tracker is measured by center location error, which the Euclidean distance between center of the ground truth and prediction box. Precision plot reflects the percentage of pictures that meet the condition that the Euclidean distance between the center point of the prediction box and the ground truth is less than the specified threshold, generally 20 pixels. Success rate is measured by intersection over union (IoU) between prediction box and ground truth. Area-under-the-curve (AUC) on the success plot is used for ranking.

2) UAV dataset benchmark: We use three UAV tracking datasets to evaluate our tracker, namely DTB70[37], UAV123_10fps[3], and UAVDT[38]. We compare our tracker with 11 state-of-the-art trackers: SiamRPN_Alex[12], AutoTrack[39], ARCF[4], BACF[40], ECO_HC[41], DaSiamRPN[21], SiamDW_RPN_Res22[14], SeSiamFC[42], SiamAPN[28], SiamDW_FC_Res22[14], SiamAPN++[29].

DTB70 dataset is high diversity, consisting of 70 short term videos captured by drone cameras. Most of the image sequences focus on tracking people and cars with specially camera motion designed. The success and precision plots are shown in the Fig. 4. Compared with the algorithms for correlation filtering, our algorithm performs much better than them, both in success and precision. Specifically, our tracker achieves a success score of 0.611, which outperforms SiamAPN++ (0.594) and SiamRPN_alex (0.586) with a large margin.

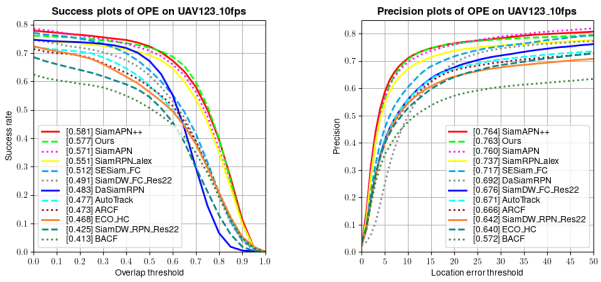


(a) Success Plot (b) Precision Plot

Fig. 4. Success and precision plots on DTB70

UAV123_10fps is downsampled from UAV123 dataset. The original UAV123 dataset includes 123 sequences with average sequence length of 915 frames. This dataset is characterized by clean background and high perspective variation. As the distance between frames becomes larger, some of the original challenges will become more difficult. Fig. 5 shows the result of success and precision on UAV123_10fps. The AUC of success 0.577 is higher than most of the trackers, while the precision is only 0.001 lower than SiamAPN++ (0.764).

UAVDT dataset consist 80k frames captured by an UAV platform at various urban locations. The sequences contain between 83 and 2970 frames. The videos sequences are

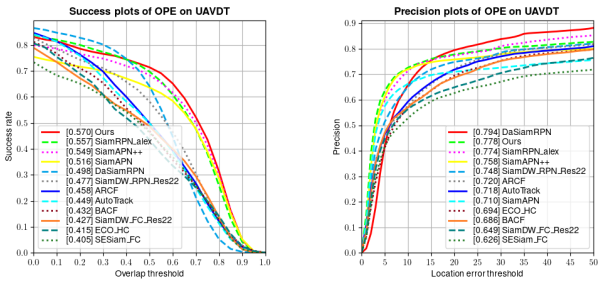


(a) Success Plot (b) Precision Plot

Fig. 5. Success and precision plots on UAV123_10fps

recorded at 30 frames per seconds with the resolution of 1080×540 pixels. The tracking performance of all trackers are shown in Fig. 6. Our method ranks the first in terms of success 0.570, while maintaining superior performance on precision 0.778.

The test results of the above three datasets show that our tracker is able to demonstrate excellent performance in UAV tracking scenarios. Besides, some qualitative evaluations are shown in Fig. 7. It clearly shows that our tracker maintains remarkable performance under fast motion and illumination variation scenes with other trackers.



(a) Success Plot (b) Precision Plot

Fig. 6. Success and precision plots on UAVDT

C. Speed Test

Our approach aims to design an efficient and effective tracker and handle the challenges in aerial tracking. In UAV tracking missions, speed is a critical evaluation as both the UAV and the target are moving continuously. Therefore, in order to better illustrate the performance and performance of our tracker compared to other deep learning-based trackers, we conducted a further study on NVIDIA Jetson Xavier NX. It brings supercomputer performance to the edge in a small form factor system-on-module (SOM), and it is a common onboard computer for small UAV. DTB70 is used to test the processing speed and prediction precision. The results are shown in Fig. 8. Our tracker achieves the fastest speed among the others. While the precision is a little bit lower than SiamBAN, our tracker's is nearly 5 times faster than it. In addition, owing to the lightweight and efficient design, our tracker can run real-time and maintain impressive precision in aerial tracking.

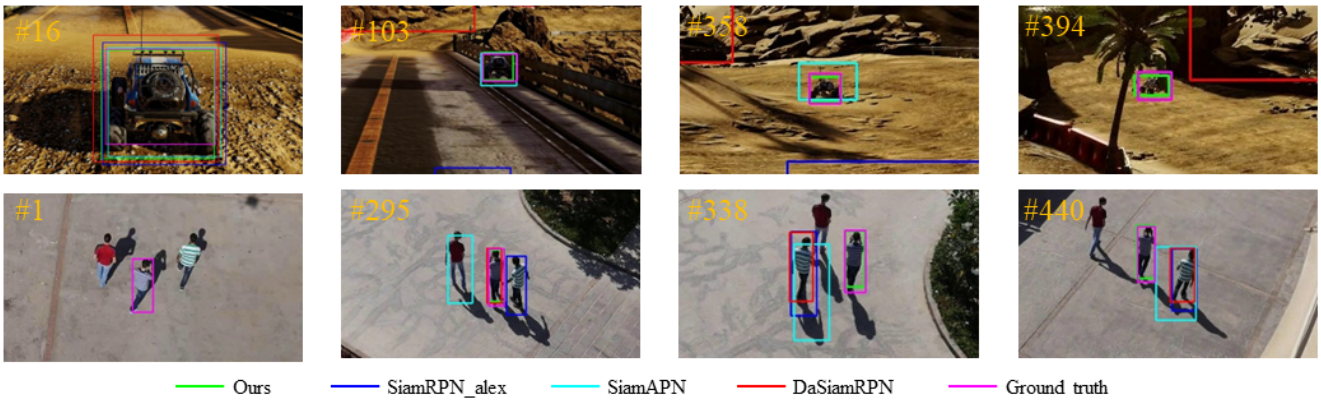


Fig. 7. Visualization of the tracking results of EFTrack and other trackers on some challenging sequences, where the green boxes indicate our tracker's results and the pink boxes indicate ground truth. From top to bottom, the sequences are car3_s and group1_1 from UAV123_10fps. Our tracker shows remarkable aerial tracking performance.

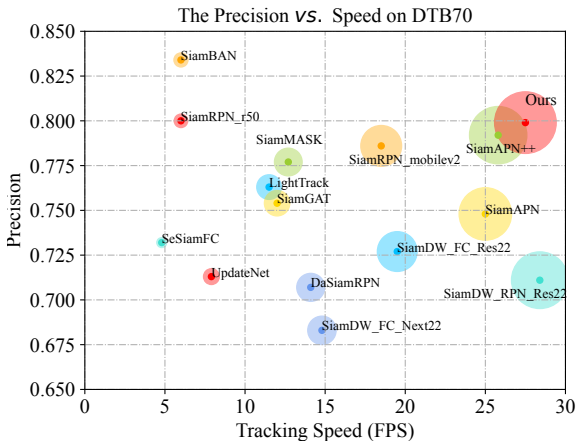


Fig. 8. Comparisons to trackers with siamese tracker on DTB70 using NVIDIA Jetson Xavier NX. Our tracker achieves nice results compared with other trackers and processes quickly.

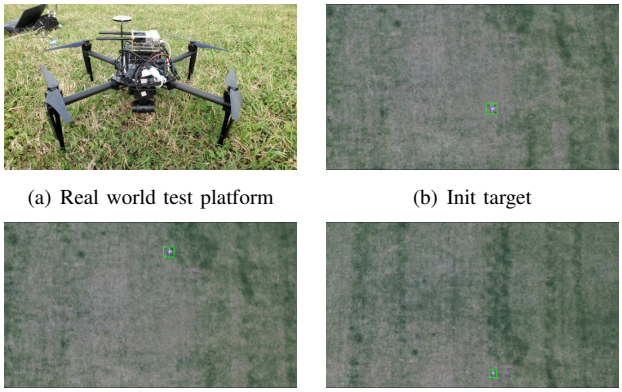


Fig. 9. Real-world test on UAV.

D. Real world experiment

In this subsection, we deploy our tracker on the UAV platform to test its effectiveness in a real-world application scenario. NVIDIA Jetson Xavier NX and DJI M100 are used as our onboard computer and UAV platform as shown in Fig. 9. A flying UAV is used as our tracking target. The whole tracking process faces three main challenges: fast moving target, camera motion and scale change. During the real-world UAV tests, the utility of GPU maximum to 40%@1.1GHz with Jetson Clocks activating, and power consumption has an average of 7 watts, and there is no noticeable lag, which can meet the requirements of aerial tracking. The real-world tests on our UAV demonstrate the usefulness of our trackers.

V. CONCLUSIONS AND DISCUSSION

In this work, we propose a lightweight aerial tracker based on siamese network architecture. The simplicity of network structure as well as the employment of siamese backbone, correlation and prediction head can handle most scenarios during aerial tracking. Tracking is not required to know the target class, similarly to solving a comparison problem, a simple comparison problem with a tiny network is sufficient. Consequently, this work can bring some new insights into tracking task development.

Limited by the short-term training method, our network's performance on long-time tracking still needs to be improved. During the inference time, The arithmetic power of the GPU is not fully utilized. The characteristic of a network like EfficientNet is the use of a large number of low FLOPs and heavy data read and write operations. These operations with high data read and write volumes, and the GPU's access bandwidth limitations, lead to excessive time for the model reading and writing data from the video memory. For further works, TensorRT and ONNX can be used in these studies to accelerate the inference speed.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61876151 and 62032018 and in part by the Industry-University-Research Innovation Fund for the China Universites under Grant 2021ZYA09001.

REFERENCES

[1] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "Lsar: Multi-uav collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55 817–55 832, 2019.

[2] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep learning for unmanned aerial vehicle-based object detection and tracking: a survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 91–124, 2021.

[3] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," in *Proceedings of the European conference on computer vision*, 2016, pp. 445–461.

[4] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time uav tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2891–2900.

[5] C. Fu, J. Ye, J. Xu, Y. He, and F. Lin, "Disruptor-aware interval-based response inconsistency for correlation filters in real-time aerial tracking," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 8, pp. 6301–6313, 2020.

[6] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[7] C. Fu, B. Li, F. Ding, F. Lin, and G. Lu, "Correlation filters for unmanned aerial vehicle-based aerial tracking: a review and experimental evaluation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 125–160, 2021.

[8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE computer society conference on Computer Vision and Pattern Recognition*, 2010, pp. 2544–2550.

[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2014.

[10] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 1420–1429.

[11] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision – ECCV 2016 Workshops*, 2016, pp. 850–865.

[12] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "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.

[13] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siamrpn++: Evolution of siamese visual tracking with very deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4282–4291.

[14] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 4591–4600.

[15] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 7952–7961.

[16] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020, pp. 6668–6677.

[17] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "Siamcar: Siamese fully convolutional classification and regression for visual tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020, pp. 6269–6277.

[18] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proceedings of the European conference on computer vision*, 2020, pp. 771–787.

[19] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, 2020, pp. 12 549–12 556.

[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, 2019, pp. 6105–6114.

[21] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1763–1771.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[23] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proceedings of the European conference on computer vision*, 2018, pp. 101–117.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25, 2012.

[27] M. Danelljan, L. V. Gool, and R. Timofte, "Probabilistic regression for visual tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020, pp. 7183–7192.

[28] C. Fu, Z. Cao, Y. Li, J. Ye, and C. Feng, "Siamese anchor proposal network for high-speed aerial tracking," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 510–516.

[29] Z. Cao, C. Fu, J. Ye, B. Li, and Y. Li, "Siamapn++: Siamese attentional aggregation network for real-time uav tracking," in *IEEE International Conference on Intelligent Robots and Systems*, 2021, pp. 3086–3092.

[30] B. Yan, H. Peng, K. Wu, D. Wang, J. Fu, and H. Lu, "Lighttrack: Finding lightweight neural networks for object tracking via one-shot architecture search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15 180–15 189.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European conference on computer vision*, 2014, pp. 740–755.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[34] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, and H. Ling, "Lasot: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019, pp. 5374–5383.

[35] L. Huang, X. Zhao, and K. Huang, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2019.

[36] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 2411–2418.

[37] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[38] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proceedings of the European conference on computer vision*, 2018, pp. 370–386.

[39] Y. Li, C. Fu, F. Ding, Z. Huang, and G. Lu, "Autotrack: Towards high-performance visual tracking for uav with automatic spatio-temporal regularization," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 923–11 932.

[40] H. Kiani Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1135–1143.

[41] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 6638–6646.

[42] I. Sosnovik, A. Moskalev, and A. W. Smeulders, "Scale equivariance improves siamese tracking," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2021, pp. 2765–2774.