A Real-time and Lightweight Method for Tiny Airborne Object Detection

Yanyi Lyu Zhunga Liu Huandong Li Dongxiu Guo Yimin Fu
Northwestern Polytechnical University
{lvyanyi, lihuandong, gdx, fuyimin96}@mail.nwpu.edu.cn
liuzhunga@nwpu.edu.cn

Abstract

With wide applications of unmanned aerial vehicles (UAVs), the detection of airborne objects has become crucial to ensure the flight safety of UAVs and prevent their illegal use. Although object detection has achieved great success in past years, it is still a challenging problem to detect tiny airborne objects. To solve this problem, we propose a simple and effective Tiny Airborne object Detection (TAD) method. It locates potential objects using inconsistent motion cues between airborne objects and backgrounds instead of the low-quality representation of tiny objects. This enables TAD to sensitively detect tiny objects with limited appearance information. Specifically, we first establish correspondences of pixels between adjacent frames based on the local similarity of spatial feature vectors to achieve motion modeling. Next, the local similarity of motion patterns is computed to explicitly describe the motion consistency of each position with its surrounding pixels. Then, a simple network is used to output the heatmap that reflects the probability of object presence. A higher probability of containing an object will be assigned to positions with a greater difference in motion from their surrounding pixels. Finally, an independent network branch is employed to regress center offsets and scale information of objects, which are used to correct the error in the estimated object position from the heatmap and obtain the final bounding box, respectively. Experiments on three challenging datasets demonstrate that the proposed method can achieve advanced performance. Notably, TAD is highly lightweight, and the detection speed is significantly better than existing methods.

1. Introduction

Recently, UAV technology has rapidly developed and been extensively applied to various fields. The trend in UAV development is towards greater intelligence and autonomy, imposing more requirements on UAVs' environmental perception capability. Meanwhile, UAVs have low cost and low detectability, making them ideal for attacks, smuggling, and illegal surveillance. Accurate positioning of aerial UAVs is a prerequisite for preventing UAV accidents. Therefore, the detection of airborne objects holds significant research importance.

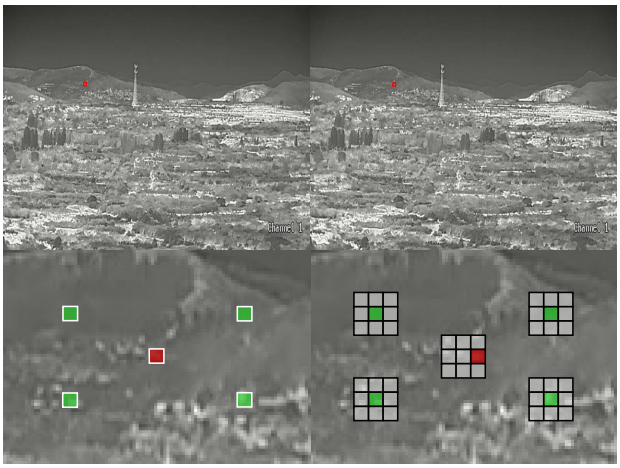

Figure 1. Illustration of the tiny size of airborne object and motivation for the proposed method. The first row shows two adjacent frames, in which the object is enclosed by a red rectangular box. The second row shows local zoom-in views near the object. The red block represents the area corresponding to the object, while the green block represents the pixel area in the background. Each block searches for the most similar pixel area around its location. It can be observed that the object's position has shifted.

Compared to common object detection, detecting airborne objects faces unique challenges, such as the tiny size and high speed of airborne objects and the large motion of cameras [28]. Tiny objects lack distinct visual features, making them difficult to detect and easily obscured by clutter [42]. Especially for objects moving against complex backgrounds, distinguishing them from the environment is more challenging. In addition, the high speed of airborne objects brings additional computational efficiency requirements. A practical method must be able to detect objects within a limited time window. Significant delays are unacceptable in certain situations, such as autonomous obstacle

avoidance [27] and safe multi-drone flights [43].

Recent advances in deep learning have achieved impressive results in object detection [24, 36, 37], but this prosperity cannot conceal the unsatisfactory situation of small object detection [5, 44]. Typically, object detectors leverage sub-sampling operations to reduce redundant information in the picture, which is beneficial for normal-sized objects. Unfortunately, the limited appearance information of small objects is regarded as a noisy signal in this process and is almost wiped out, which is fatal for tiny object detection. Some work has been proposed to prevent feature degradation of small objects by fusing low-level features [7, 38, 40], generating high-resolution images [2], or generating richer features [6, 19, 26]. However, the performance improvements of these methods come at the cost of additional network branches or redundant calculations, which is unfavorable for fast object detection. [13, 18, 21, 30] detect small objects by utilizing the relationship between objects and the environment. While these methods are beneficial for certain tasks, such as tiny face detection, they are ineffective for detecting airborne objects that are isolated from the environment.

Despite the limited visual appearance of small objects, a noticeable difference in motion can be detected between the object and the background as illustrated in Fig. 1. This observation has motivated some research to emphasize the significance of motion information in detecting small objects [1, 12]. Current methods mainly utilize optical flow estimation [1, 20, 25] or background subtraction [8, 41] to leverage motion cues. However, these methods do not perform well in scenes with large background changes caused by fast camera movements. Moreover, optical flow-based methods can cause additional computational burden, making them unsuitable for real-time applications.

To address the above challenges, we propose an efficient method for tiny airborne objects in this paper. The improvement or reconstruction of object representations typically demands sophisticated feature extractors or fusion strategies, which can potentially escalate detection latency. In contrast, a motion-based detector can balance performance and speed if a simple and effective method of describing motion is adopted. Different from existing methods that exploit motion cues by optical flow estimation or background subtraction, we model motion patterns by calculating the local similarity of spatial feature vectors. This motion description method is applicable in dynamic backgrounds and can be accelerated through parallel computation. Next, we describe the consistency of the motion directly by calculating the local similarity of the motion patterns. Then, a simple network is employed to regress the center of the object. Finally, TAD uses an independent branch to extract the features near the potential object to predict the bounding box coordinates. As the tiny size is uninformative, locating tiny objects is sufficient in many practical application scenarios. For ultimate efficiency, we remove the bounding boxes regression branch of TAD and release its lightweight version, TAD-Lightning, which has fewer parameters and a faster running speed.

We evaluate our method on AOT[1] dataset, NPS-Drones [20] dataset and Anti-UAV benchmark [16] that contain a large amount of tiny airborne objects. Compared with the current popular methods, TAD achieved impressive results in terms of both speed and accuracy.

Our major contributions can be summarized as follows:

- A heuristic motion description method is proposed to achieve fast pixel-level motion modeling. It explicitly characterizes the motion inconsistency to guide the detection process.
- A motion consistency-based detection network is proposed to effectively detect tiny airborne objects with limited appearance information. It enables ultra-fast detection and achieves competitive performance compared to state-of-the-art methods.
- The proposed method only requires a small number of parameters and computational resources, making it suitable for practical applications.

2. Related Work

In recent years, various methods have been proposed to solve the challenge of detecting small objects in static images. The primary obstacle to small object detection is the loss of limited representations during feature extraction. Therefore, multi-scale learning methods have been developed to preserve discriminant features of small objects [9, 11, 22, 35, 38]. This paradigm, exemplified by FPN [22], enhances object representation by integrating low-level details with high-level semantic features. The current state-of-the-art method for infrared small object detection, UIU-Net [35], is based on this paradigm. It incorporates a small U-Net network into a larger U-Net backbone, thereby facilitating multi-level and multi-scale representation learning of small objects. Another line of efforts intends to bridge the gap between the representation of large and small objects rather than reusing low-level features [2, 6, 19, 26]. Following this idea, Bai *et al*. [2] proposed a multi-task generative adversarial network to super-resolve the patches of RoIs. The representation of the small object is recovered by up-sampling small objects to large scales. Similarly, some methods work at the feature level to reconstruct the representation of tiny objects [6, 19, 26]. Compared to image-level super-resolution methods, feature-level super-resolution methods are more efficient and take contextual

[1] https://registry.opendata.aws/airborne-object-tracking.

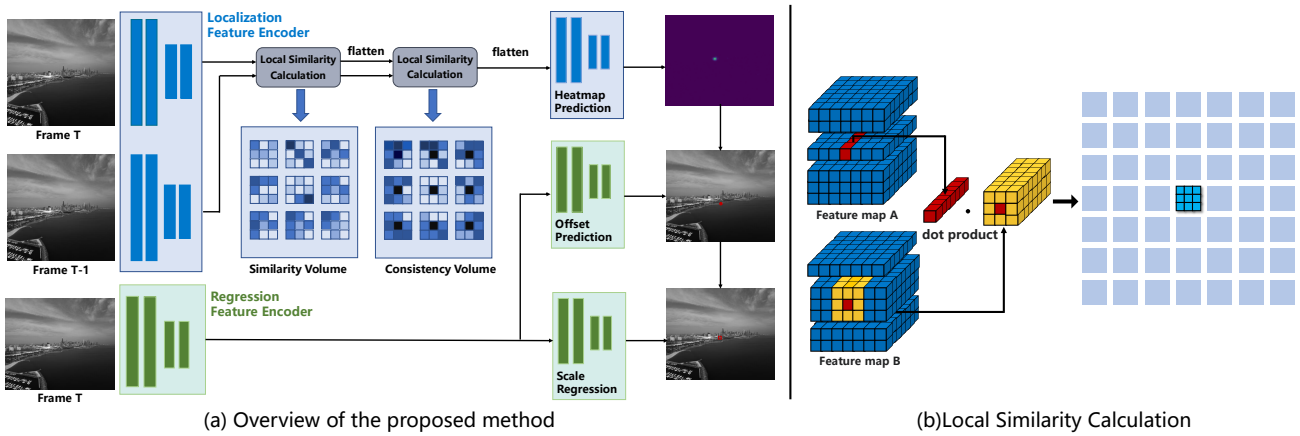

Figure 2. The framework of TAD and the calculation process of similarity volume. TAD divides the detection task into the localization and bounding boxes regression. Aiming at practical applications, TAD-Lightning removes the bounding boxes regression branch to reduce parameters and achieve faster speed.

information into account. There is also a class of methods to facilitate the detection of small objects by utilizing the correlation between the objects and environmental information or other easily detected objects [13,18,21,30]. For example, Tang *et al*. proposed PyramidBox [30] to accurately detects small faces from features that incorporate contextual information based on the prior knowledge that the head and body always appear together with the face. The interaction between different objects can also be considered as contextual information, such as FS-SSD [21]. It exploits spatial distances between intra-class and inter-class objects to recall low-score detections. Apart from the above methods, data augmentation is simple and effective for small object detection. Kisantal *et al*. [17] copied small objects and pastes them randomly into the images, significantly improving detection accuracy of small objects.

However, applying existing methods directly to aerial scenes presents three potential challenges. First, feature fusion, super-resolution, or learning contextual relationships may increase the computational burden. It is important to carefully consider whether detection speed can be traded for performance when detecting fast-moving airborne objects. Second, the tiny size of airborne objects often falls outside the general dataset definition of small objects, which means that methods designed for small objects may face performance bottlenecks when detecting tiny objects. Third, data augmentation-based and super-resolution-based methods may only be effective for specific datasets and scenarios, making it difficult to generalize these approaches to aerial scenes.

In addition to the tiny size, another characteristic of airborne objects is that they are always in motion. Therefore, some methods rely on motion cues to detect tiny airborne objects. These methods can be divided into two categories based on the different ways of extracting motion cues. One category is based on background subtraction [14, 33], and the other is based on optical flow estimation [1, 20, 25]. Background subtraction methods compare the current frame to a background image to detect moving regions, where regions with significant differences are considered objects and regions with slight differences are considered background. However, these methods are limited to static or slowly changing backgrounds. In contrast, optical flow-based detectors are available in dynamic scenes. Li *et al*. [20] characterized the spatio-temporal features of moving objects through optical flow estimation, and subsequently detected objects based on motion pattern differentiation between objects and their backgrounds. Nevertheless, the traditional optical flow estimation methods are sensitive to changes in lighting conditions and may yield inaccurate or unreliable results for objects with large motions. Moreover, they typically require multiple iterations per pixel point, which is computationally expensive. While several deep learning-based methods have been proposed to overcome these constraints [15, 29, 31], the integration of an optical flow estimation network into a detection network is cumbersome and does not facilitate end-to-end optimization. Furthermore, optical flow estimation methods trained on specific datasets may exhibit limited generalizability to other scenarios.

3. Proposed Method

In this section, we introduce the technical detail of our tiny airborne object detection method, TAD, and its lightweight version TAD-Lightning. Generally, our method is divided into four steps: (1) feature extraction, (2) calculation of similarity volume and consistency volume, (3)

prediction of the heatmap, and (4) regression of bounding boxes. The overall framework is demonstrated in Fig. 2.

3.1. Feature Extraction

TAD divides the detection task into two subtasks: a classification task for localization and a regression task for bounding box regression. These two subtasks are learned through two separate lightweight branches. The two branches use feature extractors that have the same structure but do not share weights. This design brings two benefits: First, joint training of multiple tasks may result in sub-optimal solutions for both tasks. By decoupling the two tasks, competition between the branches is mitigated, thereby improving overall performance. Second, using two independent branches in the network makes TAD modular, allowing the combination of branches to be tailored to specific application requirements.

Because motion-based TAD does not rely on specific feature descriptions, the selection of feature extractors for TAD is flexible. We implement the feature encoder network in the same way as RAFT [31]. To be specific, the network consists of two convolutional layers and six residual blocks. We first change the channels of the input images to 64 using a single convolutional layer and then add two residual blocks at 1/2 resolution, two residual blocks at 1/4 resolution, and two residual blocks at 1/8 resolution to extract features. Finally, a 1×1 convolution layer is adopted to produce the desired output. Given an images with shape $H \times W \times C$, the shape of the output is $H/8 \times W/8 \times D$, where we set D=256.

The feature extractor in the localization branch takes the current frame and the adjacent historical frame as input, while the feature extractor in the bounding box regression branch only extracts features from the current frame. As TAD-Lightning solely predicts heatmaps for localization, it utilizes only one feature extractor.

3.2. Local Similarity Calculation

TAD localizes potential objects by identifying pixel regions that exhibit inconsistent motion with the background. To achieve this, TAD requires effective modeling of the motion cues in the input images. A simple yet effective approach is to describe motion by establishing correspondences between pixels in adjacent frames. Considering that the motion of objects is continuous, we propose the local similarity calculation to describe the motion. As shown in Fig. 2, the method calculates the cosine similarity between the feature vector at each position in the feature map and its surrounding feature vectors. As the same pixel region in two images should have the highest similarity, the correspondence between pixels can be determined based on the result of the local similarity calculation. When object motion is irregular or fast, a larger similarity calculation range can be selected to expand the search window.

We refer to the tensor obtained by performing the local similarity calculation once as the similarity volume, which provides information about the motion trends of objects. Specifically, we iterate through each position (i, j) in the first image feature $f^1 \in \mathbb{R}^{H/8 \times W/8 \times D}$ and generate the similarity volume $C \in \mathbb{R}^{H/8 \times W/8 \times k \times k}$ by calculating the cosine similarity between (i, j) and its neighbors (m, n) in the second image feature $f^2 \in \mathbb{R}^{H/8 \times W/8 \times D}$. The hyperparameter k determines the range within which the most similar pixels are searched and can be flexibly adjusted based on actual conditions. In our experiments, we set $k = 3$. Note that k can only be an odd number other than 1. If the airborne objects move faster or the camera undergoes more abrupt motion, a larger value of k may be set. However, we recommend selecting a smaller k value whenever possible. A smaller search range can reduce the risk of erroneous associations and save computational resources. The process is formulated as:

$$S^k_{ijmn} = f^1_{ij} \cdot f^2_{(i-\lfloor k/2 \rfloor + m)(j-\lfloor k/2 \rfloor + n)}, 0 \leq m, n < k. \tag{1}$$

The $k \times k$ values of S_{ij} represent the similarity between the pixel with coordinates (i, j) in f^1 and its neighbors in f^2. The peak of S_{ij} is the relative position of the pixel (i, j) in f^2 after motion. By this way, we describe the motion of the object and the background without complicated calculations.

The motion-based detection paradigm actually utilizes the inconsistency of motion cues between the object and background to detect objects. The gap between motion trend and motion consistency can affect the convergence speed and results of the detector. Therefore, we perform local similarity calculation on the similarity volume to generate a tensor called the consistency volume, which directly describes the consistency of motion. The similarity volume is flattened in the last two dimensions and serves as the input of local similarity calculation. Similar to the computation of similarity volume, the consistency volume $C \in \mathbb{R}^{H/8 \times W/8 \times k \times k}$ is computed as follows:

$$C^k_{ijmn} = \sum_{k_1=0}^{k} \sum_{k_2=0}^{k} S_{ijk_1k_2} \cdot S_{(i-\lfloor k/2 \rfloor + m)(j-\lfloor k/2 \rfloor + n)k_1k_2}. \tag{2}$$

We call a set of $k \times k$ elements within each consistency volume as a consistency matrix. The values in the consistency matrix C_{ij} measure the consistency of the motion of the pixel at coordinate (i, j) with its surrounding neighborhood. Since background motion is rigid, pixels in the background possess consistent motion direction locally. As a result, each element in the consistency matrix corresponding to the background has a high value. In contrast, the motion of the object is non-rigid. Therefore, a consistency matrix with a low value is highly likely to correspond to an object.

The traversal operation to calculate the similarity is very time-consuming. Since we do not compute all pixel pairs in the input image, we cannot directly use the matrix dot product to speed up the calculation. In order to improve computational efficiency, we implement the parallel processing of local similarity calculation that trades space for time.

Taking the calculation of similarity volume as an example, the feature map $f^1 \in \mathbb{R}^{H/8\times W/8\times D}$ is firstly duplicated to $f^{1*} \in \mathbb{R}^{H/8\times W/8\times k\times k\times D}$, and the $f^2 \in \mathbb{R}^{H/8\times W/8\times D}$ is also extended to $f^{2*} \in \mathbb{R}^{H/8\times W/8\times k\times k\times D}$ using a sliding window. The corresponding positions of f^{1*} and f^{2*} are then multiplied and summed along the last channel. Experimental results show that the optimized local similarity calculation method gains great efficiency improvement. During inference, the parallel implementation accelerates 20 times on the CPU and 286 times on GPU compared with traversal operation.

3.3. Heatmap Prediction

Due to the explicit modeling of motion differences in TAD, object localization can be achieved with a simple network architecture. The network utilizes two convolutional layers to classify each consistency matrix within the consistency volume, thereby generating object presence probabilities at each spatial location. Specifically, the consistency volume C is flattened along its last two dimensions and is used as input to the prediction head. The predicted results take the form of a $H/8 \times W/8 \times 2$ tensor, with the last two channels representing the probability of consistent and inconsistent motion, respectively. It is worth noting that TAD does not rely on object appearance information, thereby making it independent of object representation. This feature enables TAD to detect tiny objects with limited appearance.

3.4. Bounding Boxes Regression

TAD has an additional bounding box regression branch compared to TAD-Lightning. In addition to the feature encoder introduced in Sec. 3.1, the two important components of the bounding box regression branch are the offset prediction head and the scale regression head. They are responsible for predicting the offset of the object center and the length and width information of the object, respectively.

The coordinates estimated from the heatmap indicate the location of the object in the feature map. As the feature map is obtained by downsampling the image, the coordinates need to be upscaled to obtain the location of the object in the image. However, this process introduces quantization errors. The offset prediction head refines the location of the object center by predicting the offset between the true object center and the rough object center. The appearance feature of the first frame $f_a^1 \in \mathbb{R}^{H/8\times W/8\times D}$ is used as input, and the output $O \in \mathbb{R}^{H/8\times W/8\times 2}$ is the offset of the center in the x and y directions. The scale regression head has the same structure and input as the offset prediction head. It outputs the width and height of bounding boxes.

3.5. Training and Inference

The training of TAD is conducted in stages to achieve better detection performance. We first train the localization branch and then train the bounding box regression branch with the network parameters of the localization branch frozen. The loss function for heatmap prediction is formulated as:

$$L_h = -\frac{1}{N}\sum_{xy}\begin{cases}\left(1-\widetilde{Y}_{xy}\right)^{\alpha}\log\left(\widetilde{Y}_{xy}\right) & \text{if } Y_{xy}=1,\\ \left(1-Y_{xy}\right)^{\beta}\left(\widetilde{Y}_{xy}\right)^{\alpha}\log\left(1-\widetilde{Y}_{xy}\right) & \text{Otherwise,}\end{cases} \tag{3}$$

on which N is the number of objects, Y_{xy},$\widetilde{Y}_{xy}$ are the prediction and ground truth at the position with coordinates (x, y). α and β are hyperparameters of focal loss.

We supervised offset prediction and scale regression with the following loss:

$$L = \lambda_o L_{offset} + \lambda_s L_{scale}, \tag{4}$$

in which L_{offset} and L_{scale} are the smooth L1 distance between the predicted and ground truth, and λ_o, λ_s are hyperparameters to balance these two tasks. We use $\lambda_o = 0.5$ and $\lambda_s = 1$ in our experiments.

During inference, given a pair of consecutive RGB images, the feature encoders will generate the feature maps, respectively. Then, the local similarity calculation is employed on the feature map to generate similarity volume. The similarity volume serves as the input of another local similarity calculation for the consistency volume. Finally, we predict the heatmap based on the consistency volume to locate the object center. Meanwhile, the bounding boxes regression branch extracts the feature map of current frames to predict center offsets and scale information of objects.

We use the max pooling operation instead of IoU-based non-maxima suppression (NMS) to remove duplicated detections. The kernel size and stride of the max pooling operation are set to 3, meaning only peaks in each 3*3 region are reserved. We consider the value corresponding to peaks as detection confidence and remove peaks with confidence under the threshold. We take the preserved peak as the object center (x, y) and then shift the center according to the predicted offset $(\hat{x}, \hat{y})$. Combined with the scale information (w, h) given by the scale regression head, we can decode final bounding boxes that are represented by (t, l, b, r), where (t, l), (b, r) are respectively the coordinates of the top left corner and bottom right corner. The formula is as follows:

$$\begin{aligned} t &= x - \hat{x} - w/2 \times W,\\ l &= y - \hat{y} - h/2 \times H,\\ b &= x - \hat{x} + w/2 \times W,\\ r &= y - \hat{y} + h/2 \times H, \end{aligned} \tag{5}$$

where W, H are the width and height of the image.

Table 1. Quantitative comparison of TAD with several state-of-the-art approaches on the AOT dataset

Framwork	Model	Recall	Acc	F1	AP@50	FPS	Parameters
Two-stage	Faster R-CNN w/fpn [22]	0.214	0.675	0.325	0.161	14.8	41.13M
	Cascade-RCNN [3]	0.257	0.224	0.239	0.171	7.3	87.92M
One-stage	Retinanet [23]	0.679	0.882	0.767	0.615	16.6	36.10 M
Anchor free	QueryDet [38]	0.687	0.867	0.767	0.647	-	-
	FCOS [32]	0.493	0.857	0.626	0.432	15.8	31.00M
Transformer	Swin-Transformer [24]	0.260	0.888	0.402	0.231	4.4	93.87 M
Ours	TAD	0.704	0.832	0.763	0.663	180.7	2.326M
	TAD-Lightning	0.723	0.855	0.784	0.696	342.1	1.024M

4. Experiments

4.1. Experiment Setup

Datasets and Metrics. Airborne Object Tracking (AOT) is a collection of flight sequences collected by aircraft equipped with high-resolution cameras. In AOT, images are 2448 pixels wide by 2048 pixels high, and objects usually appear quite small at distances that are relevant for early detection. There are three subsets of AOT, and we use the first for training and the second for validation. In order to limit the computational burden, we reduce the length and width of the image by half and then select images with object sizes less than 16×16 to evaluate TAD. These objects are 0.01% of the image size on average.

The NPS dataset is a collection of high-definition video sequences used for detecting airborne vehicles. The dataset is captured using GoPro cameras mounted on a delta-wing aircraft. Unlike the grayscale images in AOT, the images in the NPS dataset are in color. The video has a frame rate of 30 frames per second, and a resolution of 1920×1080 or 1280×960. The average size of objects in the images is 16.2×11.6. In our experiments, the dataset is divided in the same way as Dogfight. The first forty videos are used for training, and the last ten are used for testing.

Anti-UAV contains high-quality video sequences of both RGB and infrared images. It covers a variety of scenarios with multi-scale UAVs. Detecting tiny objects in infrared images is challenging because tiny objects blend into the environment when other objects or the background have temperatures similar to them. We performed a qualitative analysis of images containing tiny objects in this dataset to validate the effectiveness of the proposed method on different modal data.

We use commonly used metrics in object detection tasks to evaluate methods. We measure detection quality using Recall, Accuracy, F1-score, and AP while using FPS and Parameters to measure computational efficiency.

Implementation Details. All modules are initialized from scratch with random weights during training. The BatchNorm layers are frozen during inference. We use Adam without the decay parameter as the optimizer and set the initial learning rate to 1×10^{-4}. The batch sizes of the first and second stages are 10 and 5, respectively. We divide the training process into two stages. Both stages are trained for 48 iterations. The hyperparameters α and β are set to 2 and 4, following [23].

4.2. Comparison with State-of-the-art Methods

We compare TAD with detectors in various frameworks on the AOT dataset. All compared methods are implemented based on official codes or with the recommended configuration and training strategy provided by MMdetection. The RTX3060 is used to test detection speed. Tab. 1 reports the detailed comparison results. Our method exhibits significant superiority over existing methods on several representative metrics. Especially in terms of the FPS, Parameters, and FLOPs, which represent the lightweight and detection speed, TAD surpasses other detectors by a large margin.

Note that TAD-Lightning calculates the metrics differently than other detectors because it only predicts the center of objects and does not generate the final bounding boxes. When the center predicted by TAD-Lightning hits inside the ground truth, we take it as True Positive (TP). The reason for presenting the TAD-Lightning in Tab. 1 is two-fold: firstly, to demonstrate its reduced parameterization and increased speed, and secondly, to illustrate the upper limits of TAD's performance capabilities.

The quantitative comparison of TAD with other detectors on the NPS dataset is shown in Tab. 2. The experimental results of the comparison method are from [1]. The proposed method achieves the highest detection accuracy and average precision on the NPS dataset while being simpler and more efficient than existing methods. We observed that methods achieve better performance across various metrics on the NPS dataset compared to the AOT dataset. We at-

Table 2. Quantitative comparison of TAD with several state-of-the-art approaches on the NPS dataset

Method	Precision	Recall	F1 score	AP
SCRDet-H [39]	0.81	0.74	0.77	0.65
SCRDet-R [39]	0.79	0.71	0.75	0.61
FCOS [32]	0.88	0.84	0.86	0.83
Mask-RCNN [10]	0.66	0.91	0.76	0.89
MEGA [4]	0.88	0.82	0.85	0.83
SLSA [34]	0.47	0.67	0.55	0.46
Dogfight [1]	0.92	0.91	0.92	0.89
TAD	0.93	0.87	0.90	0.91

tribute this to several factors: first, the images we selected from the AOT dataset contained many tiny objects, such as birds and hot air balloons, which are smaller than most of the drones in the NPS dataset. Second, the majority of objects in the NPS dataset were white drones, which are easily distinguishable from the surrounding environment.

4.3. Ablation Analysis

Effect of Components. In this section, we carry out ablation analysis to verify the effectiveness of different modules. We focus on the impact of similarity volume, consistency volume, and offset prediction head on model performance. To simplify the experiments without compromising rationality, we add or remove modules based on TAD-Lightning and evaluate the performance of different combinations on the AOT dataset. The evaluation metrics for all models follow the same calculation rules as TAD-Lightning in Tab. 1.

High-quality motion description is crucial for TAD. Both similarity volume and consistency volume can characterize motion cues. To understand their contributions to performance, we train three networks using the similarity volume, consistency volume, and a combination of both to describe motion.

The experimental results in Tab. 3 demonstrate that all versions can effectively detect tiny objects in the air, but differences in performance exist. The version that only uses similarity volume lags behind the other two versions in all metrics. This is because the network needs to first learn how to model the differences between motions and then locate objects based on the inconsistency of the motions. This presents a challenge to the capacity of the network. In addition, the motion trends of the background in the image have local consistency rather than global consistency. For example, when the camera shoots forward, the upper left region tends to move to the left, while the lower right region tends to move to the right. This conflicting motion information can confuse the network. Therefore, this version is suboptimal and has room for improvement. The consistency volume directly describes the motion differences between the object and surrounding pixels, and this method of description has a consistent physical meaning at various positions in the image. Therefore, the performance of the version that uses consistency volume has been greatly improved. However, when combining similarity volume and consistency volume, the performance does not improve further, and some metrics even decline. Based on the previous analysis, we believe that the conflict between the local information contained in the similarity volume and the global information contained in the consistency volume led to the degradation of performance.

Table 3. Ablation for important modules. S, C, and O, respectively refer to similarity volume, consistency volume, and offset prediction head. We use ✓to indicate that the module is selected

#	S	C	O	Recall	Acc	F1	AP
1	✓			0.676	0.854	0.753	0.644
2		✓		0.723	0.855	0.784	0.696
3	✓	✓		0.677	0.871	0.762	0.651
4	✓		✓	0.677	0.853	0.755	0.646
5		✓	✓	0.727	0.859	0.787	0.700
6	✓	✓	✓	0.678	0.873	0.763	0.652

Table 4. Results of TAD inference with different search radii

K	Recall	Acc	F1	AP
3	0.723	0.855	0.784	0.696
5	0.683	0.901	0.777	0.661
7	0.687	0.895	0.777	0.659

We employ the offset prediction head to correct the quantization error caused by downsampling. When comparing versions with and without the offset prediction head, we can see that the former performs better. This indicates that the offset prediction head learns how to refine the center of objects, resulting in the estimated object center being closer to the actual object center.

Effect of Search Radii. We investigated the effect of different search radii when generating similarity volume in Table Tab. 4. TAD can tolerate larger perspective abrupt changes when searching for objects over a larger area. However, the large search window will decrease the consistency of motion, which will cause more false detections. Experimental results show that a trade-off between recall and accuracy can be achieved when we set K=3.

4.4. Visualization and Failure Cases

We visualize heatmaps outputted by TAD that reflect the probability of object presence. All images shown in Fig. 3

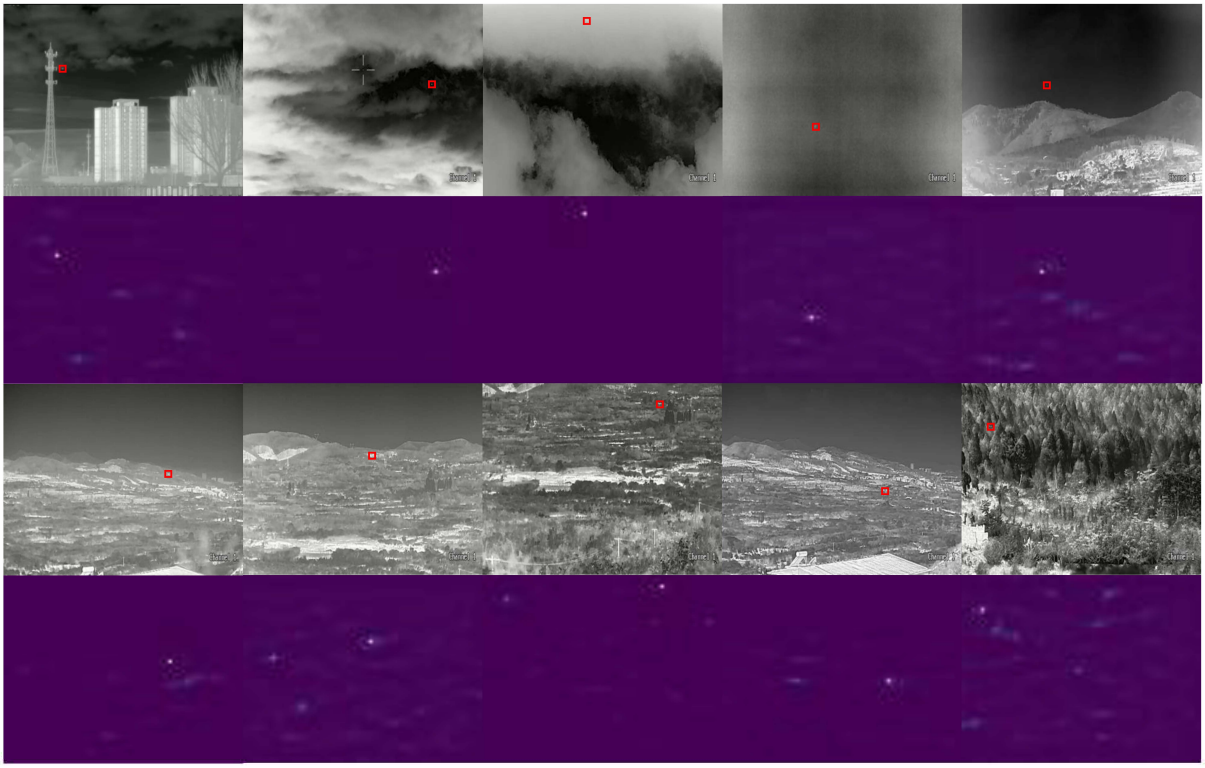

Figure 3. Visualization of the heatmap for tiny objects of TAD on Anti-UAV dataset. The images in the first and third rows are from the test set, with objects enclosed in red boxes for ease of review. The second and fourth rows show the heatmap output of TAD. The brightness in the heatmap reflects the degree of difference in motion between the position and its surrounding pixels, with brighter areas indicating a higher probability of the object being present.

are from the Anti-UAV dataset. Due to the unique imaging mechanism of infrared images, a portion of the object's appearance information, such as color and texture, has been lost in images. In addition, positions in the image with the same temperature as the object exhibit similar visual features. When the object is tiny, it becomes even more difficult to distinguish from the background. We can see that the heatmaps output by TAD can effectively locate the approximate position of objects, even in complex backgrounds. This is due to TAD does not rely on appearance information to locate objects and its powerful motion modeling ability based on the local similarity calculation. However, our method also has the following limitations: 1) inability to detect stationary objects, such as hovering drones. 2) detection of other moving objects in the image, such as fast-moving vehicles, which are not airborne objects. In addition, it should be noted that TAD is not the optimal solution for detecting large airborne objects. On the one hand, large airborne objects contain sufficient visual information, and generic object detectors can achieve good detection results. On the other hand, TAD cannot locate the center of the object when detecting large-size objects, but only the corner of the object.

5. Conclusion

In this paper, we propose a real-time and lightweight method to detect tiny airborne objects. We first introduce local similarity calculation, a parallelizable motion modeling method, to explicitly describe the consistency of motion. Then, we use the inconsistent motion between objects and the background to locate the approximate position of potential objects. Finally, an additional network branch is used to predict the precise position of objects. Qualitative and quantitative experiments on three challenging datasets have demonstrated that the proposed method achieves extremely fast detection speed without compromising performance. Due to the simple network structure and low computational complexity of the proposed method, it is highly applicable to practical applications. In the future, we will try to extend TAD to a tracking method to improve detection continuity and enable us to predict object motion.

Acknowledgements: This work was supported in part by the National Natural Science Foundation of China under Grant U20B2067.

References

[1] Muhammad Waseem Ashraf, Waqas Sultani, and Mubarak Shah. Dogfight: Detecting drones from drones videos. In *CVPR*, 2021. 2, 3, 6, 7

[2] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *ECCV*, 2018. 2

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 6

[4] Yihong Chen, Yue Cao, Han Hu, and Liwei Wang. Memory enhanced global-local aggregation for video object detection. In *CVPR*, 2020. 7

[5] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *arXiv preprint arXiv:2207.14096*, 2022. 2

[6] Chunfang Deng, Mengmeng Wang, Liang Liu, Yong Liu, and Yunliang Jiang. Extended feature pyramid network for small object detection. *IEEE Transactions on Multimedia*, 24:1968–1979, 2021. 2

[7] Kaiwen Duan, Dawei Du, Honggang Qi, and Qingming Huang. Detecting small objects using a channel-aware deconvolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1639–1652, 2019. 2

[8] Zhihang Fu, Yaowu Chen, Hongwei Yong, Rongxin Jiang, Lei Zhang, and Xian-Sheng Hua. Foreground gating and background refining network for surveillance object detection. *IEEE Transactions on Image Processing*, 28(12):6077–6090, 2019. 2

[9] Yuqi Gong, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, and Zhenjun Han. Effective fusion factor in fpn for tiny object detection. In *WACV*, 2021. 2

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 7

[11] Mingbo Hong, Shuiwang Li, Yuchao Yang, Feiyu Zhu, Qijun Zhao, and Li Lu. Sspnet: Scale selection pyramid network for tiny person detection from uav images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2021. 2

[12] Mengshun Hu, Jing Xiao, Liang Liao, Zheng Wang, Chia-Wen Lin, Mi Wang, and Shin'ichi Satoh. Capturing small, fast-moving objects: Frame interpolation via recurrent motion enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6):3390–3406, 2021. 2

[13] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *CVPR*, 2017. 2, 3

[14] Bo Huang, Junjie Chen, Tingfa Xu, Ying Wang, Shenwang Jiang, Yuncheng Wang, Lei Wang, and Jianan Li. Siamsta: Spatio-temporal attention based siamese tracker for tracking uavs. In *ICCVW*, 2021. 3

[15] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 3

[16] Nan Jiang, Kuiran Wang, Xiaoke Peng, Xuehui Yu, Qiang Wang, Junliang Xing, Guorong Li, Qixiang Ye, Jianbin Jiao, Zhenjun Han, et al. Anti-uav: a large-scale benchmark for vision-based uav tracking. *IEEE Transactions on Multimedia*, 2021. 2

[17] Mate Kisantal, Zbigniew Wojna, Jakub Murawski, Jacek Naruniec, and Kyunghyun Cho. Augmentation for small object detection. *arXiv preprint arXiv:1902.07296*, 2019. 3

[18] Chunggi Lee, Seonwook Park, Heon Song, Jeongun Ryu, Sanghoon Kim, Haejoon Kim, Sérgio Pereira, and Donggeun Yoo. Interactive multi-class tiny-object detection. In *CVPR*, 2022. 2, 3

[19] Jianan Li, Xiaodan Liang, Yunchao Wei, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Perceptual generative adversarial networks for small object detection. In *CVPR*, 2017. 2

[20] Jing Li, Dong Hye Ye, Timothy Chung, Mathias Kolsch, Juan Wachs, and Charles Bouman. Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs). In *IROS*. IEEE, 2016. 2, 3

[21] Xi Liang, Jing Zhang, Li Zhuo, Yuzhao Li, and Qi Tian. Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(6):1758–1770, 2019. 2, 3

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2, 6

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 6

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2, 6

[25] Murari Mandal, Lav Kush Kumar, and Santosh Kumar Vipparthi. Mor-uav: A benchmark dataset and baselines for moving object recognition in uav videos. In *ACMMM*, 2020. 2, 3

[26] Junhyug Noh, Wonho Bae, Wonhee Lee, Jinhwan Seo, and Gunhee Kim. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *ICCV*, 2019. 2

[27] Juntong Qi, Jinjin Guo, Mingming Wang, Chong Wu, and Zhenwei Ma. Formation tracking and obstacle avoidance for multiple quadrotors with static and dynamic obstacles. *IEEE Robotics and Automation Letters*, 7(2):1713–1720, 2022. 2

[28] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Detecting flying objects using a single moving camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(5):879–892, 2016. 1

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 3

[30] Xu Tang, Daniel K Du, Zeqiang He, and Jingtuo Liu. Pyramidbox: A context-assisted single shot face detector. In *ECCV*, 2018. 2, 3

[31] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*. Springer, 2020. 3, 4

[32] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 6, 7

[33] Muhammad Uzair, Russell SA Brinkworth, and Anthony Finn. Bio-inspired video enhancement for small moving target detection. *IEEE Transactions on Image Processing*, 30:1232–1244, 2020. 3

[34] Haiping Wu, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Sequence level semantics aggregation for video object detection. In *ICCV*, 2019. 7

[35] Xin Wu, Danfeng Hong, and Jocelyn Chanussot. Uiu-net: U-net in u-net for infrared small object detection. *IEEE Transactions on Image Processing*, 32:364–376, 2022. 2

[36] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *CVPR*, 2018. 2

[37] Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented r-cnn for object detection. In *ICCV*, 2021. 2

[38] Chenhongyi Yang, Zehao Huang, and Naiyan Wang. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. In *CVPR*, 2022. 2, 6

[39] X Yang, J Yang, J Yan, Y Zhang, T Zhang, Z Guo, X Sun, and K SCRDet Fu. Towards more robust detection for small, cluttered and rotated objects. In *ICCV*, volume 27, 2019. 7

[40] Hui Zhang, Kunfeng Wang, Yonglin Tian, Chao Gou, and Fei-Yue Wang. Mfr-cnn: Incorporating multi-scale features and global information for traffic object detection. *IEEE Transactions on Vehicular Technology*, 67(9):8019–8030, 2018. 2

[41] Anran Zhou, Weixin Xie, and Jihong Pei. Background modeling in the fourier domain for maritime infrared target detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2634–2649, 2019. 2

[42] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7380–7399, 2021. 1

[43] Pengfei Zhu, Jiayu Zheng, Dawei Du, Longyin Wen, Yiming Sun, and Qinghua Hu. Multi-drone-based single object tracking with agent sharing network. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):4058–4070, 2020. 2

[44] Yabin Zhu, Chenglong Li, Yao Liu, Xiao Wang, Jin Tang, Bin Luo, and Zhixiang Huang. Tiny object tracking: A large-scale dataset and a baseline. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–15, 2023. 2