remote sensing

MDPI

Article

One-Shot Multiple Object Tracking in UAV Videos Using Task-Specific Fine-Grained Features

Han Wu [1], Jiahao Nie [1], Zhiwei He [1,2,*], Ziming Zhu [1] and Mingyu Gao [1,2]

[1] The School of Electronic Information, Hangzhou Dianzi University, Hangzhou 310018, China
[2] Zhejiang Province Key Laboratory of Equipment Electronics, Hangzhou 310009, China
* Correspondence: zwhe@hdu.edu.cn

Abstract: Multiple object tracking (MOT) in unmanned aerial vehicle (UAV) videos is a fundamental task and can be applied in many fields. MOT consists of two critical procedures, i.e., object detection and re-identification (ReID). One-shot MOT, which incorporates detection and ReID in a unified network, has gained attention due to its fast inference speed. It significantly reduces the computational overhead by making two subtasks share features. However, most existing one-shot trackers struggle to achieve robust tracking in UAV videos. We observe that the essential difference between detection and ReID leads to an optimization contradiction within one-shot networks. To alleviate this contradiction, we propose a novel feature decoupling network (FDN) to convert shared features into detection-specific and ReID-specific representations. The FDN searches for characteristics and commonalities between the two tasks to synergize detection and ReID. In addition, existing one-shot trackers struggle to locate small targets in UAV videos. Therefore, we design a pyramid transformer encoder (PTE) to enrich the semantic information of the resulting detection-specific representations. By learning scale-aware fine-grained features, the PTE empowers our tracker to locate targets in UAV videos accurately. Extensive experiments on VisDrone2021 and UAVDT benchmarks demonstrate that our tracker achieves state-of-the-art tracking performance.

Keywords: multiple object tracking; unmanned aerial vehicle video; optimization contradiction; task-specific representation; transformer encoder; fine-grained feature

Citation: Wu, H.; Nie, J.; He, Z.; Zhu, Z.; Gao, M. One-Shot Multiple Object Tracking in UAV Videos Using Task-Specific Fine-Grained Features. *Remote Sens.* **2022**, *14*, 3853. https://doi.org/10.3390/rs14163853

Academic Editor: Eufemia Tarantino

Received: 8 July 2022
Accepted: 5 August 2022
Published: 9 August 2022

1. Introduction

Multiple object tracking (MOT) aims to predict the trajectories of all targets from a given video. With the development of computer vision, MOT is widely applied in many fields, such as intelligent video surveillance [1], human-computer interaction [2], and autonomous driving [3]. In addition, MOT is the foundation for advanced computer vision tasks such as video understanding [4], behavior recognition [5], and behavior analysis [6]. In recent years, due to the strong flexibility and high safety of unmanned aerial vehicles (UAVs), MOT from the UAV view has attracted the attention of many scholars. However, there exist various challenges for MOT based on airborne platforms, including occlusions, low resolution, and small targets.

Recently, benefiting from the rapid development of object detection techniques, the tracking-by-detection (TBD) paradigm has become the mainstream MOT framework. The TBD paradigm decomposes the MOT task into three main steps [7], i.e., object detection, feature extraction, and data association. In the tracking process, TBD-based trackers associate detected targets in different frames into complete trajectories based on their appearance, motion, and other features. To distinguish different targets under complex scenarios, most trackers consider re-identification (ReID) as a vital component and extract the identity embedding of targets through a ReID network. Based on the relationship between detection and ReID networks, TBD-based trackers can be classified into two-stage and one-shot approaches. As shown in Figure 1a, the two-stage trackers treat detection

and ReID as two completely independent subtasks. Two-stage methods first locate targets through a detector and then predict the identity embedding of targets using a ReID network. Although effective and robust, two-stage trackers suffer from the drawbacks of large computational overhead and high complexity. Since the two-stage approaches inferred two deep networks successively. This disadvantage of slow tracking speed makes it difficult to deploy in real applications. Therefore, one-shot trackers integrate detection and ReID into a unified network, as shown in Figure 1b. By making the two tasks share features, one-shot trackers predict the location and embedding of targets in a single network. One-shot trackers have a significant speed advantage because they avoid a lot of repeated calculations. However, in contrast, most existing one-shot methods struggle to achieve robust tracking in UAV videos.

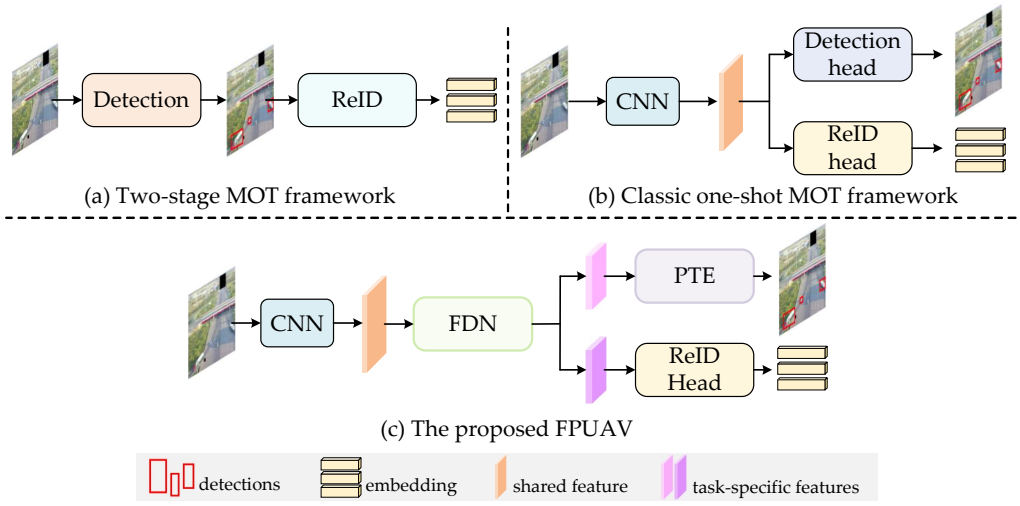


Figure 1. Comparison of the MOT frameworks.

We argue that the main reason for the sub-optimal tracking performance of one-shot trackers is the neglect of the conflicts within the network. Previous one-shot trackers mostly incorporate a ReID head directly into an off-the-shelf detector [8–10]. Then, the detection and ReID make predictions based on the shared feature map. This scheme of sharing features seems reasonable but ignores the essential difference between the detection and ReID tasks. Specifically, detection aims to find common information between targets of the same category, while ReID searches for differences between individual targets. Due to this contradiction, the shared features cannot simultaneously satisfy the required representations for both tasks. Therefore, sharing features directly leads to difficulties in optimizing the network. Optimization of either of the two tasks is likely to lead to significant performance decay of the other, while the performance degradation of detection or ReID results in many tracking failures, such as bounding box drift and missed targets. Moreover, existing trackers mainly focus on tracking regular-sized and slow-moving targets. In contrast, a large number of small targets and blurred targets caused by bidirectional motion in UAV videos exist. Therefore, previous trackers struggle to locate all targets in UAV videos.

To improve the tracking performance of one-shot trackers in UAV videos, we propose two modules to alleviate the above issues. As shown in Figure 1c, we first propose a feature decoupling network (FDN) to mitigate the contradiction between the two tasks. The FDN converts the extracted shared features into detection-specific and ReID-specific representations. To achieve this, we learn the commonalities and characteristics between the tasks through self-attention and cross-attention. Then, we design a pyramid transformer encoder (PTE) to enrich the semantic information of the detection-specific representations. Given the detection-specific features obtained by the FDN, the PTE performs multi-scale learning on them and captures fine-grained features. The PTE enables our tracker to locate targets of different sizes under complex scenarios accurately.

The main contributions of this paper are summarized as follows:

- We observe that the contradiction between detection and ReID constrains the tracking performance of one-shot trackers. To alleviate this contradiction, we propose a feature decoupling network (FDN) to convert shared features into detection-specific and ReID-specific representations.
- We design a pyramid transformer encoder (PTE) to transform the detection-specific features generated by the FDN into scare-aware fine-grained representations. The PTE enables the one-shot network to locate targets in UAV videos accurately.

2. Related Work

In this section, we first review the tracking-by-detection paradigm. Then, we divide the recent trackers into two-stage and one-shot methods and briefly describe each tracker.

2.1. Tracking-by-Detection

The tracking-by-detection (TBD) paradigm splits the MOT task into three phases: object detection, feature extraction, and data association [11]. Specifically, the TBD-based trackers associate detected targets in different video frames into complete tracklets based on their similarities.

Most early works focus attention on improving the data association performance. Milan et al. [12] considered data association as a global optimization problem and integrated data association and tracklets into energy functions. Finally, they constrain the tracklets by constructing a motion model. Many studies defined the MOT task as a graph model [13–16] in which each vertex denoted a detected target, and the edges represented the similarity among targets. Then, the matching relationship of each vertex was determined by the Hungarian algorithm [17] or the greedy algorithm [18]. Ren et al. [13] formulated MOT as a network flow, where a flow was defined as an indicator connecting two nodes, and a tracklet corresponded to a flow path in the graph. The network flow-based trackers can obtain the optimal global solution in polynomial time and improve tracking accuracy by simultaneously considering information from multiple frames. However, these methods have difficulty taking into account multidimensional information in the tracking process. Xiang et al. [14] defined MOT as a conditional random field (CRF). Given the tracklets as input, the CRF predicted the probabilistic relationship among the detections and each tracklet. CRF-based methods can effectively model the interaction among targets but are prone to fall into local optimality. Peng et al. [15] modeled MOT as a graph clustering problem based on the minimum cost subgraph multicut (MCSM). The MCSM measured the similarity among targets by edge-related costs and then united multiple high-confidence targets in temporal and spatial dimensions. The resulting cluster represented a tracked tracklet. Brendel et al. [16] formulated MOT as a maximum-weight independent set, which was the heaviest subset of non-adjacent nodes in the attribute graph. The nodes in the attribute graph denoted the track pairs in consecutive video frames, and the weights indicated the affinity of the pairs. If multiple tracklets share the same detection, the nodes are connected so that the global association results can be obtained through the attribute map. These methods utilize information from multiple frames in the tracking process and therefore usually achieve favorable tracking accuracy and robustness. However, due to the large computational overhead, these methods are not suitable for practical applications.

With the rapid development of deep learning, research on the TBD paradigm in recent years has focused mainly on object detection and feature extraction. Current TBD-based trackers typically utilize or design high-performance detectors to localize targets precisely. In addition, benefiting from the powerful feature extraction capability of neural networks, identity embedding-based trackers have attracted extensive research interest and achieved remarkable progress. According to the manner of locating targets and extracting the appearance features, existing TBD-based trackers can be classified into two-stage approaches and one-shot approaches. We review the representative two-stage and one-shot trackers of recent years below.

2.2. Two-Stage Trackers

Two-stage trackers treat object detection and identity embedding extraction as two separate tasks. Specifically, two-stage trackers first use a detection network to locate targets and then predict the identity embedding of targets through an independent ReID network.

Benefiting from the strong feature extraction of convolutional neural networks (CNN), many trackers obtain the identity embedding of targets through deep networks. In 2016, Yu et al. [19] localized targets by Faster R-CNN [20] and designed a ReID network based on GoogLeNet [21] to predict discriminative appearance features. Son et al. [22] learned multiple images containing different targets simultaneously to predict more discriminative features. Meanwhile, they proposed a quadruplet loss to strengthen the time constraint so that there is a greater affinity among targets with shorter time intervals. Lee et al. [23] designed a ReID network integrated with a feature pyramid network (FPN) to enrich the resulting information by fusing features at different levels. Subsequently, Sun et al. [24] proposed a deep affinity network for estimating the embedding of targets and predicting inter-target affinity. These trackers perform well in crowded scenes and are robust to scale transformation of targets. However, trackers that rely only on appearance features often suffer from errors such as bounding box drift in scenarios with similar target interference.

There are frequently a large number of similar targets in real tracking scenarios. Therefore, most recent two-stage methods combine the motion and appearance features of targets for robust tracking. Wojke et al. [25] predicted the location of targets at the next moment by a Kalman filter [26] and designed a deep CNN to extract the identity embedding of targets. To alleviate the misleading data association by noisy detections and redundant trajectories, Chen et al. [27] proposed a scoring mechanism to remove unreliable detections and candidate trajectories. Zhou et al. [28] used CNN to model the motion patterns and inter-target interactions of targets. Subsequently, Shan et al. [29] and Girbau et al. [30] predicted the motion state of targets based on a graph neural network and recurrent neural network, respectively. Li et al. [31] designed an auto-tuning Kalman filter to predict the location accurately and evaluate the inter-target similarity by a recurrent neural network.

Two-stage trackers achieve high tracking accuracy and are robust to various challenges under complex scenarios. However, these methods have high network complexity and high computational overhead. Since two deep networks are inferred successively, the two-stage approaches have a slow tracking speed, which makes it difficult to satisfy the needs of real applications. Therefore, the one-shot trackers with smaller computational efforts have attracted more research attention in recent years.

2.3. One-Shot Trackers

To make the MOT technique better suited for practical applications, one-shot trackers predict the location and identity embedding of targets in a unified network. By sharing features between the two tasks, one-shot trackers avoid a lot of repeated calculations and thus effectively improve the tracking speed.

Voigtlaender et al. [32] integrated a parallel ReID head in Mask R-CNN [8] and proposed TrackR-CNN. This ReID branch predicted the identity embedding of each candidate region through the fully connected layer. Wang et al. [33] incorporated a ReID head in YOLOv3 [34] and proposed a joint detection and embedding model (JDE). The proposed JDE treated network training as a multi-task learning problem and applied an automatic balance loss [35] to balance the importance of classification, regression, and ReID within the network. JDE is the first real-time MOT tracker. However, compared with the previous two-stage methods, the tracking accuracy of JDE does not show a significant advantage. Zhang et al. [36] designed an anchor-free network , which reduced the risk of overfitting by learning the low-dimensional identity embedding. Meng et al. [37] presented a spatio-temporal attention for updating the weights of identity embedding at each moment. Liu et al. [38] designed a deformable convolution-based region transformation module to

reduce the focus on irrelevant regions in the ReID branch. Yan et al. [39] employed FPN to aggregate multi-level features to enrich the information of targets.

Although previous one-shot trackers have achieved remarkable performance, we argue current one-shot trackers still have shortcomings. Firstly, most existing one-shot methods directly integrate a parallel ReID head in an off-the-shelf detection network. However, these approaches ignore the essential difference between detection and ReID. Specifically, object detection aims to find common information among targets of the same category, while ReID focuses on finding the differences among individual targets. This contradiction makes it difficult for the extracted features to satisfy the demands of both tasks, thus making the network hard to optimize. In addition, these trackers mainly focus on tracking regular-size targets. However, UAV videos include a large number of small targets. Therefore, most existing one-shot trackers struggle to accurately locate multi-size targets in UAV videos, resulting in a large number of lost targets. To robustly track multiple targets in UAV videos, this paper proposes the FDN and the PTE. Specifically, the FDN converts the shared features into task-specific representations.The PTE further enhances the effectiveness of detection-specific features, thus enabling our tracker to locate targets of different sizes more accurately.

3. Methodology

Our one-shot tracker, namely the FPUAV, mainly consists of two components, i.e., the feature decoupling network designed in Section 3.2 and the pyramid transformer encoder described in Section 3.3. Before describing these two crucial modules in detail, we first introduce the overall framework of the FPUAV in Section 3.1. Finally, we describe the inference and training details of the FPUAV in Sections 3.4 and 3.5, respectively.

3.1. Overall Framework

We follow the principle of a one-shot MOT, i.e., locating the targets and predicting their identity embedding in a unified network. The overall framework of the FPUAV is shown in Figure 2. Firstly, we transform the input image into a feature map through the backbone network. Then, unlike previous methods that directly feed the resulting feature map into the prediction head, we propose a feature decoupling network (FDN) to convert the obtained feature map into two task-specific representations. Subsequently, we design a pyramid transformer encoder (PTE) to generate fine-grained detection-specific features to localize targets under complex scenes accurately. Finally, based on the identity embedding predicted by the ReID head, the FPUAV associates the detections of different frames as complete tracklets.

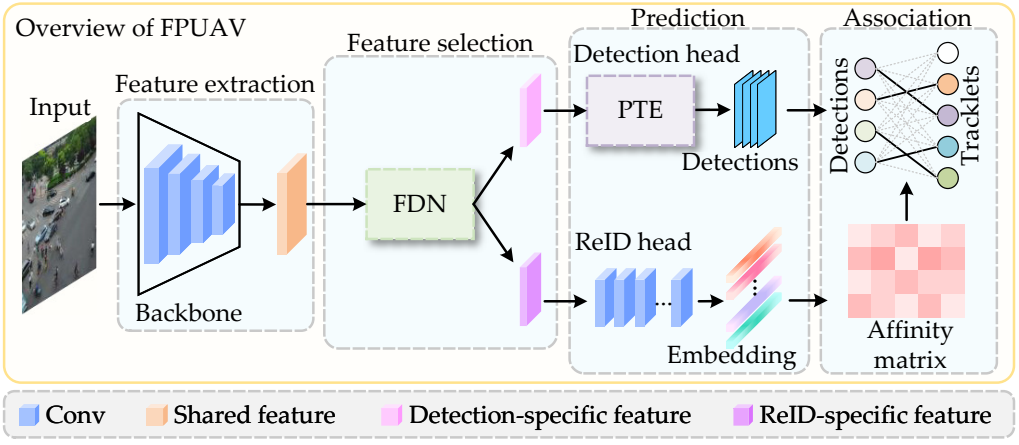


Figure 2. The architecture diagram of the FPUAV.

3.2. Feature Decoupling Network

The proposed FDN aims to decouple the shared features into two task-specific representations. The FDN utilizes self-attention to enhance the feature representations of each task and exchanges semantic information of different tasks by cross-attention. The network structure of the FDN is shown in Figure 3.

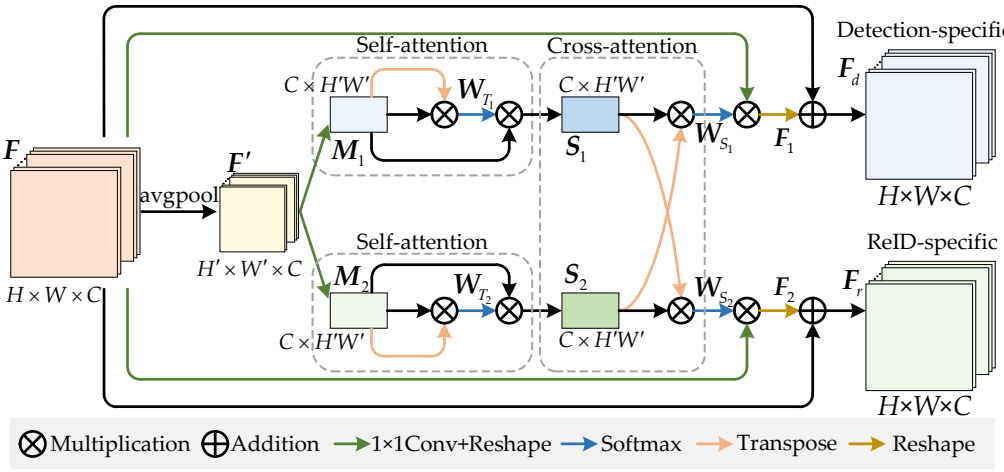


Figure 3. The architecture of the FDN. Given the original shared feature map $\boldsymbol{F}$, the FDN decouples $\boldsymbol{F}$ into two task-specific representations $\boldsymbol{F}_d$ and $\boldsymbol{F}_r$.

Given the shared feature $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$, we first capture the statistical information $\boldsymbol{F}' \in \mathbb{R}^{C\times H'\times W'}$ through an average pooling layer. Then, we pass $\boldsymbol{F}'$ through two parallel 1×1 convolution layers and reshape the resulting tensors into $\boldsymbol{M}_1, \boldsymbol{M}_2 \in \mathbb{R}^{C\times N'}$, where $N' = H' \times W'$. We multiply $\boldsymbol{M}_1$ with its corresponding transpose matrix. Meanwhile, we perform the same operation for $\boldsymbol{M}_2$. Then, we compute the self-relation weight maps $\boldsymbol{W}_{T_1}, \boldsymbol{W}_{T_2} \in \mathbb{R}^{C\times C}$ for each branch through the row softmax as follows:

$$\boldsymbol{W}_{T_k}^{ij} = \frac{\exp(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_k^j)}{\sum_{j=1}^{C} \exp(\boldsymbol{m}_k^i \cdot \boldsymbol{m}_k^j)}, k \in \{1,2\} \tag{1}$$

where $\cdot$ denotes dot product and $\boldsymbol{m}_k^i$ and $\boldsymbol{m}_k^j$ indicate the i-th and j-th rows of $\boldsymbol{M}_1$ or $\boldsymbol{M}_2$, respectively. $\boldsymbol{W}_{T_k}^{ij}$ denotes the element of $\boldsymbol{W}_{T_k}$ in the location (i,j), which represents the relationship between the i-th and j-th channels in the tensor. Then, we perform matrix multiplication on $\boldsymbol{M}_k$ and $\boldsymbol{W}_{T_k}$ to obtain $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathbb{R}^{C\times N'}$. Afterward, we seek to learn the relationship between the two tasks. To achieve this, we multiply $\boldsymbol{S}_1$ with the transpose of $\boldsymbol{S}_2$, while multiplying $\boldsymbol{S}_2$ with the transpose of $\boldsymbol{S}_1$ in the other branch. Then, we compute the cross-relation weight maps $\boldsymbol{W}_{S_1}, \boldsymbol{W}_{S_2} \in \mathbb{R}^{C\times C}$ through the row softmax as follows:

$$\boldsymbol{W}_{S_k}^{ij} = \frac{\exp(\boldsymbol{s}_k^i \cdot \boldsymbol{s}_h^j)}{\sum_{j=1}^{C} \exp(\boldsymbol{s}_k^i \cdot \boldsymbol{s}_h^j)}, (k,h) \in \{(1,2),(2,1)\} \tag{2}$$

where $\boldsymbol{s}_k^i$ and $\boldsymbol{s}_h^j$ represent the i-th and j-th channel of $\boldsymbol{S}_1$ or $\boldsymbol{S}_2$, respectively. $\boldsymbol{W}_{S_k}^{ij}$ denotes the element of $\boldsymbol{W}_{S_k}$ in the location (i,j), which expresses the effect of the i-th channel of $\boldsymbol{S}_k$ on the j-th channel of $\boldsymbol{S}_h$. We pass the input feature map $\boldsymbol{F}$ through two parallel 1×1 convolution layers and reshape them to $\mathbb{R}^{C\times N}$, where $N = H \times W$. The resulting feature maps are multiplied by $\boldsymbol{W}_{S_1}$ and $\boldsymbol{W}_{S_2}$, respectively. Then, we recover the dimensionality

of the resulting tensors to obtain $\boldsymbol{F}_1, \boldsymbol{F}_2 \in \mathbb{R}^{C\times H\times W}$. Finally, we fuse $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$ with the original feature $\boldsymbol{F}$ as follows:

$$\boldsymbol{F}_d = \lambda_d \times \boldsymbol{F}_1 + (1-\lambda_d) \times \boldsymbol{F} \tag{3}$$

$$\boldsymbol{F}_r = \lambda_r \times \boldsymbol{F}_2 + (1-\lambda_r) \times \boldsymbol{F} \tag{4}$$

where $\boldsymbol{F}_d$ and $\boldsymbol{F}_r$ are the obtained detection-specific and ReID-specific representations, respectively. λ_d and λ_r are trainable parameters.

After receiving the task-specific independent features, the FPUAV predicts the embedding of targets based on $\boldsymbol{F}_r$. Meanwhile, $\boldsymbol{F}_d$ is fed into the pyramid transformer encoder to learn fine-grained representations, allowing the tracker to predict the targets accurately.

3.3. Pyramid Transformer Encoder

In this section, we present the proposed PTE, which is shown in Figure 4. The PTE takes the detection-specific representation $\boldsymbol{F}_d$ obtained by the FDN as input. We first reshape $\boldsymbol{F}_d$ to $\boldsymbol{Q} \in \mathbb{R}^{C\times N}$. Meanwhile, we design a pyramid aggregation module (PAM) to obtain $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{C\times N}$ with rich scale information.

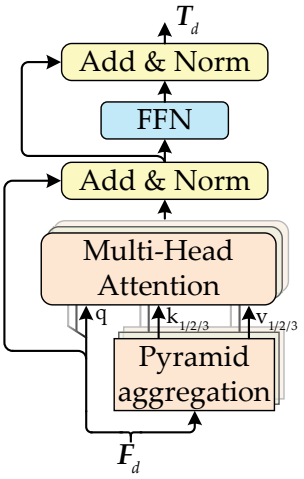


Figure 4. The pyramid transformer encoder.

As shown in Figure 5, PAM captures richer multi-scale information by making full use of both high-level and low-level information. PAM combines different convolutions and dilated convolutions (DConv) in three parallel branches to capture multi-scale features and aggregate features through a residual connection. Specifically, each branch consists of a convolution layer and a DConv layer, which can be expressed as follows:

$$\boldsymbol{M}_1 = Conv^1_{3\times3}(Conv_{1\times1}(\boldsymbol{F}_d)) \tag{5}$$

$$\boldsymbol{M}_2 = Conv^2_{3\times3}(Conv_{3\times3}(\boldsymbol{F}_d)) \tag{6}$$

$$\boldsymbol{M}_3 = Conv^3_{3\times3}(Conv_{5\times5}(\boldsymbol{F}_d)) \tag{7}$$

where $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, and $\boldsymbol{M}_3$ denote the output features of the three branches. $Conv^i_{3\times3}$ is the 3×3 DConv with dilation rate i. DConv can expand the receptive field of the feature while maintaining the resolution, as shown in Figure 6. Then, we obtain the fused feature through a residual connection as follows:

$$\boldsymbol{F}_s = Conv_{1\times1}(\boldsymbol{F}_d) + \boldsymbol{M}_1 \times \boldsymbol{M}_2 \times \boldsymbol{M}_3 \tag{8}$$

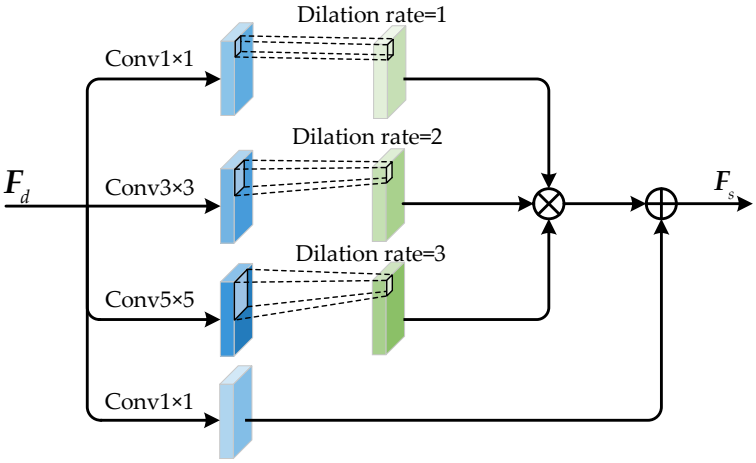


Figure 5. Illustration of the pyramid aggregation module.

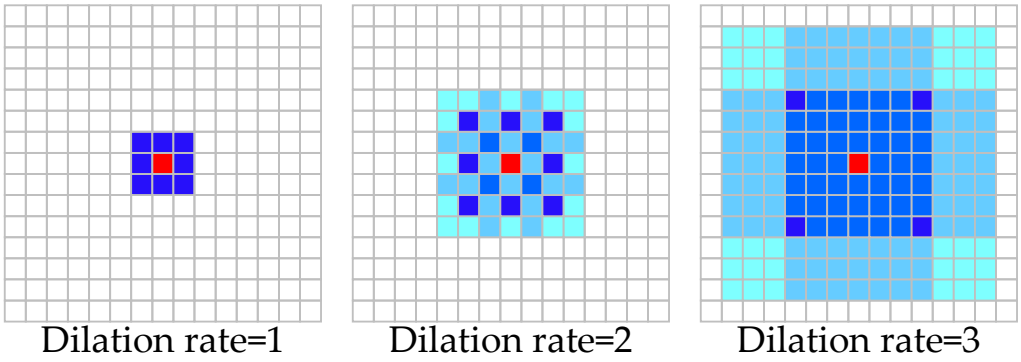


Figure 6. Illustration of the DConv layers.

All convolutions in PAM are followed by batch normalization (BN) and a ReLU activation function. Then, $\boldsymbol{F}_s$ is reshaped to $\boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{C \times N}$. Next, the PTE calculates the attention map as follows:

$$\boldsymbol{O} = \text{Softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^{\text{T}}}{\sqrt{d_k}})\boldsymbol{V} \tag{9}$$

where d_k is the dimension of $\boldsymbol{K}$. In this way, the PTE effectively captures the visual dependence of the feature map from spatial information. Subsequently, we capture the dependence information between multi-scale receptive field regions. The enhanced features $\boldsymbol{T}_{pte} \in \mathbb{R}^{C \times N}$ can be obtained as follows:

$$\begin{aligned} \boldsymbol{T}_{pte} &= \text{MultiHead}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) \\ &= \text{Norm}(\text{Concat}(\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_n)\boldsymbol{W}_o + \boldsymbol{Q}) \end{aligned} \tag{10}$$

where $\text{Norm}(\cdot)$ denotes layer normalization, and $\boldsymbol{W}_o$ is a trainable weight matrix. We recover the dimensionality of $\boldsymbol{T}_{pte}$ to obtain $\boldsymbol{T} \in \mathbb{R}^{C \times H \times W}$ and pass $\boldsymbol{T}$ through a feed-forward network (FFN) represented by 3×3 Conv-BN-ReLU. Finally, we obtain the fine-grained feature $\boldsymbol{T}_d \in \mathbb{R}^{C \times H \times W}$ through a residual connection and layer normalization.

By using the PTE, our tracker suppresses irrelevant information in the feature map, thus enabling the network to focus on multi-scale targets. The FPUAV accurately locates targets based on the fine-grained features $\boldsymbol{T}_d$ obtained by the PTE.

3.4. Online Tracking

In this section, we describe our online inference in detail. We first design a network similar to YOLOv5 for predicting the location and identity embedding of targets. Notably, the network contains the proposed FDN and PTE. After obtaining the position and em-

bedding of targets, we perform data association to associate targets at different frames. As shown in Figure 7, we employ the same cascade matching strategy as the representative one-shot tracker JDE [33]. Specifically, we first compute the affinity in identity embedding among the detections and existing tracklets by cosine distance. Then, we utilize the Kalman filter to provide positional constraints to reduce the ambiguity caused by multiple similar targets. Subsequently, we obtain the matching results of detections and tracklets through the Hungarian algorithm. In the second stage, unmatched tracklets and unmatched detections are second matched by the IOU and Hungarian algorithm. Afterward, we integrate the matching results of the two stages. We update the identity embedding of successfully matched tracklets to cope with the variations in appearance as follows:

$$\boldsymbol{a}_i^t = \varepsilon \boldsymbol{a}_i^{t-1} + (1-\varepsilon)\boldsymbol{e}_i^t \tag{11}$$

where $\boldsymbol{e}_i^t$ is the embedding of the detected target, and i is the ID index. Here, $\boldsymbol{a}_i^t$ and $\boldsymbol{a}_i^{t-1}$ denote the identity embedding of tracklets; ε is a hyperparameter and is set to 0.9 in our experiments. The unmatched detections are initialized as new tracklets and their embedding would be used as the initial embedding of the new tracklets. The unmatched tracklets would be set to inactive status. Tracklets that are inactive for 30 consecutive frames would be removed from candidate tracklets. Subsequently, matched tracklets, new tracklets, and inactive tracklets are used together as candidate tracklets for future moments.

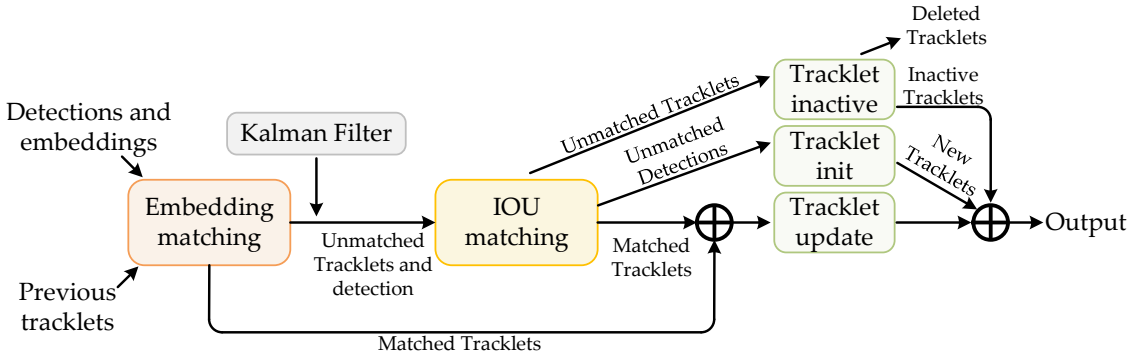


Figure 7. The details of our data association process.

3.5. Optimization Objectives

In this section, we introduce the optimization objectives of the FPUAV. Since the FPUAV contains multiple subtasks, we apply multiple training losses to optimize our network.

For the detection branch, we utilize binary cross-entropy as the classification loss L_{cls} and employ GIOU loss as the regression loss L_{reg} as follows:

$$L_{reg} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{12}$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{13}$$

$$\alpha = \frac{v}{(1-IOU)+v} \tag{14}$$

where b and b_{gt} are the centers of the predicted bounding box (PBbox) and ground truth bounding box (GBbox), $\rho(\cdot)$ denotes the Euclidean distance, c is the diagonal length of the smallest outer rectangle of the PBbox and GBbox, and w, h, w^{gt}, and h^{gt} represent the width and height of PBbox and GBbox, respectively. Then, the loss function of the detection branch can be described as:

$$L_{det} = L_{cls} + \beta L_{reg} \tag{15}$$

where β is a hyperparameter, which is set to 0.05 in our experiments. For the ReID branch, we first transform the predicted identity embedding into a class distribution vector $p = \{p_i\}_{i=1}^{K}$. K is the total number of targets. We express the one-hot representation of the ReID task as $q = \{q_j\}_{j=1}^{K}$. The ReID loss L_{id} can be formulated as:

$$L_{id} = -\sum_{j=1}^{K}\sum_{i=1}^{K} q_j \log(p_i) \tag{16}$$

Finally, we join L_{det} and L_{id} to optimize our network. The overall loss function can be described as:

$$L_{total} = \frac{1}{2}\left(\frac{1}{e^{w_1}}L_{det} + \frac{1}{e^{w_2}}L_{id} + w_1 + w_2\right) \tag{17}$$

where w_1 and w_2 are trainable parameters used to balance the importance of tasks within the network dynamically.

4. Experiments

To demonstrate the superiority of the FPUAV, we compare the FPUAV with other state-of-the-art (SOTA) trackers in Section 4.4. Then, we visualize the qualitative results of the FPUAV in Section 4.5. Finally, we construct ablation experiments in Section 4.6 to verify the effectiveness of the FDN and the PTE.

4.1. Dataset

We train and evaluate the FPUAV on VisDrone2021 [40] and UAVDT [41] benchmarks. VisDrone2021 and UAVDT are challenging large-scale datasets captured by drones. VisDrone2021 contains 56 training videos, 7 validation videos, and 33 test videos. These videos include various scenes from sports fields, neighborhoods, city roads, highways, and suburbs. UAVDT consists of 100 videos containing a total of 80k images. These videos are mainly shot in squares, highways, intersections, etc.

VisDrone2021 and UAVDT have numerous issues arising from UAVs flying under complex scenarios. Compared to other MOT benchmarks [42–44], VisDrone2021 and UAVDT have a significantly higher percentage of small targets. an extensive number of targets in the UAV video occupy less than 32×32 pixels, as shown in Figure 8a. Due to the bidirectional motion of the UAV and targets, the videos have extensive motion blur, as shown in Figure 8b. In addition, VisDrone2021 and UAVDT contain heavily crowded scenes with occlusions, as shown in Figure 8c,d. Each of these challenges may lead to undesirable tracking failures.

4.2. Evaluation Metrics

To comprehensively compare the FPUAV with other SOTA methods, we apply multiple metrics [45–47] to measure the tracking performance of our tracker. Multiple object tracking accuracy (MOTA↑) is considered the most important metric. MOTA takes various errors in the tracking process into account and is defined as follows:

$$\text{MOTA} = 1 - \frac{\text{FN} + \text{FP} + \text{IDS}}{\text{GT}} \tag{18}$$

where FN↓, FP↓, and IDS↓ indicate false negatives, false positives, and ID switches, respectively. GT is the total number of ground truth bounding boxes. Identification F1 score (IDF1↑) is a critical metric for tracking robustness and is defined as follows:

$$\text{IDF1} = \frac{2\text{IDTP}}{2\text{IDTP} + \text{IDFP} + \text{IDFN}} \tag{19}$$

where IDTP↑, IDFP↓, and IDFN↓ indicate ID true positives, ID false positives and ID false negatives, respectively. Different from MOTA, IDF1 focuses on the correctness of the ID of each trajectory, as shown in Figure 9.

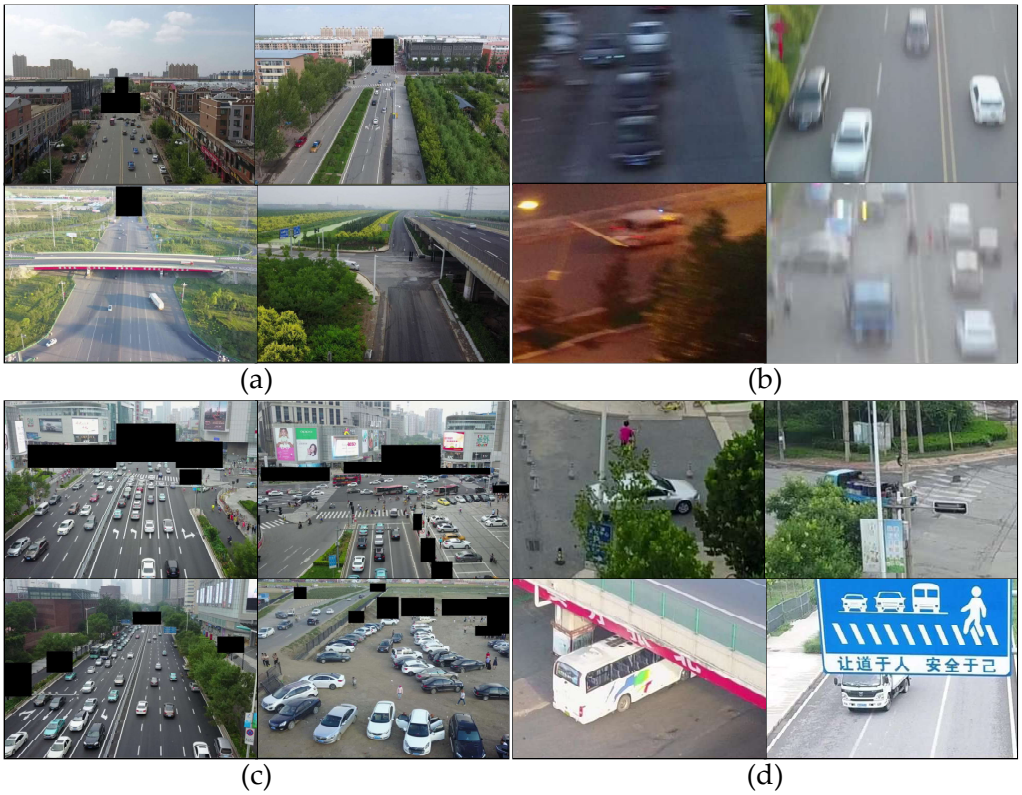


Figure 8. Hard cases of tracking in UAV videos: (**a**) small targets; (**b**) motion blur; (**c**) crowded scenarios; (**d**) occlusion.

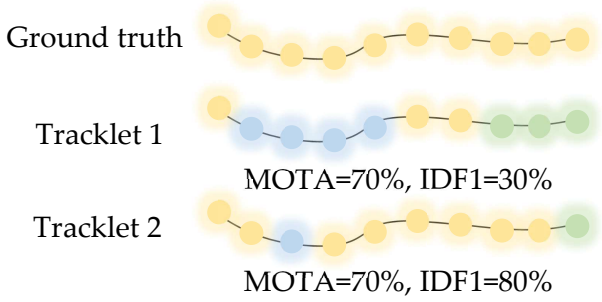


Figure 9. The importance of joint MOTA and IDF1. Different colors indicate different IDs. The MOTA is the same for both tracklets, but tracklet 2 with a higher IDF1 is obviously the superior result.

Multiple object tracking precision (MOTP↑) mainly considers the overlap degree of the tracking and ground truth boxes. Mostly tracked targets (MT↑) indicates targets with more than 80% of their ground truth boxes successfully tracked. Mostly lost targets (ML↓) indicates targets with less than 20% of their ground truth boxes successfully tracked. Fragmentation (Frag↓) represents the number of interruptions for all tracklets. In addition, we utilize overall parameters (Params↓), calculations (number of multi-adds↓), and frames per second (FPS↑) to evaluate the efficiency of our tracker. In the above metrics, ↑ means the higher score is better, while ↓ is the opposite.

4.3. Implementation Details

To further improve the robustness of the FPUAV under complex scenarios, we employ various data augmentations, such as photometric distortions, color jittering, cropping, and

random scale. We use the YOLOv5L backbone pre-trained on COCO [48] as the feature extraction network. We train the FPUAV for 30 epochs on training sets of VisDrone2021 and UAVDT using the standard SGD optimizer. The batch size is set to 10. The initial learning rate is set to 5×10^{-4} and decays to 5×10^{-5} at the 20th epoch. The resolution of each input image is resized to 1088×608. Our code is implemented using Python 3.7 and PyTorch 1.7. We train the network on 2 NVIDIA GTX1080Ti GPUs and evaluate the FPUAV on a single GPU.

4.4. Comparison with Preceding SOTAs

We compare the FPUAV with preceding state-of-the-art trackers on VisDrone2021 and UAVDT benchmarks. The experimental results are reported in Tables 1 and 2, with the **best** and the second-best marked. The experimental results show that the FPUAV achieves state-of-the-art tracking performance.

Table 1. Comparison with state-of-the-art methods on VisDrone2021.

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	IDS↓	Frag↓
MOTDT [27]	−0.8	21.6	68.5	87	1196	1437	3609
SORT [25]	14.0	38.0	73.2	506	**545**	3629	4838
IOUT [49]	28.1	38.9	74.7	467	670	2393	3829
GOG [50]	28.7	36.4	**76.1**	346	836	1387	**2237**
DAN [51]	28.9	37.7	74.8	535	602	1952	5634
JDE [33]	26.6	34.9	74.1	516	751	3200	3176
FairMOT [36]	30.8	41.9	74.3	577	697	3007	2996
MOTR [52]	22.8	41.4	72.8	272	825	**959**	3980
TrackFormer [53]	25.0	30.5	73.9	385	770	4840	4855
FPUAV	**34.3**	**45.0**	74.2	**585**	688	2138	2577

Table 2. Comparison with state-of-the-art methods on UAVDT.

Method	MOTA↑	IDF1↑	MOTP↑	MT↑	ML↓	IDS↓	Frag↓
CEM [54]	−6.8	10.1	70.4	94	1062	1530	**2835**
SMOT [55]	33.9	45.0	72.2	524	367	1752	9577
GOG [50]	35.7	0.3	72	627	374	3104	5130
IOUT [49]	36.6	23.7	72.1	534	357	9938	10463
CMOT [56]	36.9	57.5	**74.7**	664	351	1111	3656
SORT [25]	39.0	43.7	74.3	484	400	2350	5787
DSORT [57]	40.7	58.2	73.2	595	358	2061	6432
DAN [51]	41.6	29.7	72.5	648	367	12902	13610
JDE [33]	39.5	55.3	73.6	624	442	3124	8536
FairMOT [36]	44.9	60.9	73.8	672	365	2279	7163
MDP [58]	43.0	61.5	73.5	647	**324**	**541**	4299
FPUAV	**48.6**	**66.2**	73.7	**692**	349	1999	5387

As shown in Table 1, the FPUAV achieves the highest scores on MOTA, IDF1, and MT. The FPUAV obtains 34.3% on MOTA and 45.0% on IDF1. It outperforms the recently proposed two-stage tracker DAN [51] by 5.4% (34.3–28.9%) and 7.3% (45.0–37.7%), respectively. The FPUAV surpasses the representative one-shot tracker FairMOT [36] by 3.5% (34.3–30.8%) on MOTA and 3.1% (45.0–41.9%) on IDF1. Compared to TrackFormer [53], which also uses transformer, our tracker outperforms it by 9.3% (34.3–25.0%) on MOTA and 14.5% (45.0–30.5%) on IDF1. Meanwhile, the FPUAV surpasses other comparison trackers on MT, indicating the high integrity of our output trajectories. In addition, the FPUAV achieves second place on Frag, showing that the FPUAV is not prone to lose targets.

To further measure our tracker, we evaluate the FPUAV on UAVDT. On the UAVDT benchmark, the FPUAV obtains 48.6% on MOTA and 66.2% on IDF1. the FPUAV exceeds DAN [51] by 7.9% (48.6–41.6%) on MOTA and 36.5% (66.2–29.7%) on IDF1. DSORT [57] extracts the identity embedding of targets through a deep network. However, the FPUAV

still surpasses DSORT by 8.0% (66.2–58.2%) on IDF1. This is the advantage that the proposed FDN provides by learning ReID-specific representations. Meanwhile, the FPUAV outperforms recently released one-shot tracker FairMOT [36] by 3.7% (48.6–44.9%) on MOTA and 5.3% (66.2–60.9%) on IDF1. Furthermore, the FPUAV achieves first and second place on MT and ML, respectively.

The superiority of the FPUAV on MOTA and MT mainly comes from the proposed FDN and PTE. Specifically, the FDN enables our tracker to learn detection-specific representations, while the PTE further enhances the effectiveness of the detection-specific features. The synergy of the FDN and the PTE improves the accuracy of the FPUAV in locating targets, thus reducing failure cases such as false positives and false negatives. The lead of the FPUAV on IDF1 benefits from the FDN, which enables our tracker to learn ReID-specific representations to predict more discriminative identity embedding.

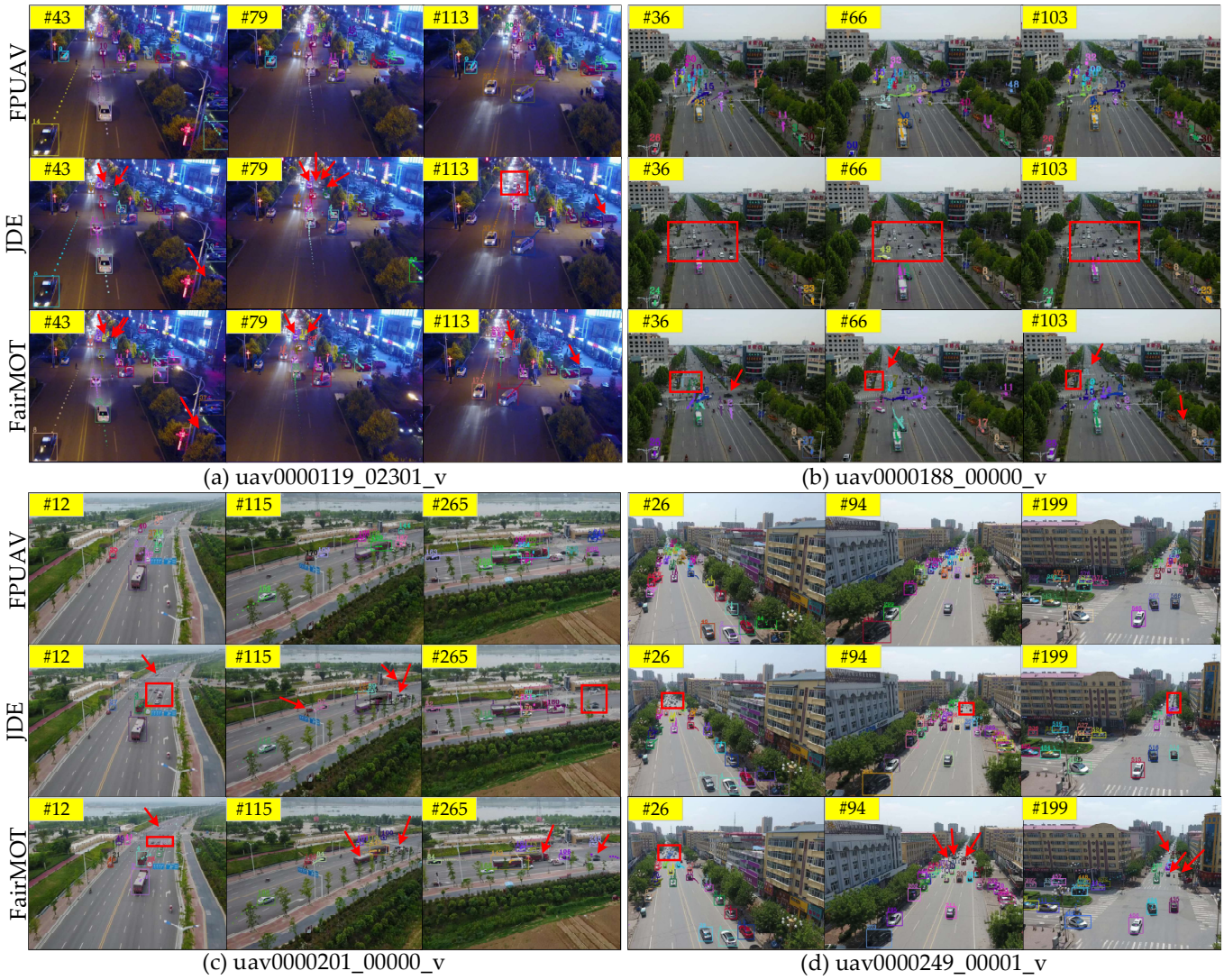
To verify the efficiency of the proposed network, we test the number of parameters and computations and the tracking speed of the FPUAV on Visdrone2021. Moreover, we measure the efficiency of two-stage trackers DSORT and DAN and one-shot trackers JDE and FairMOT. The comparison results are reported in Table 3. Since two deep networks are inferred successively, the computational overhead of DSORT and DAN is much higher than that of one-shot trackers. Likewise, the tracking speed of DSORT and DAN is difficult to satisfy the needs of practical applications. By making detection and ReID share features, one-shot trackers have considerable computational overhead and tracking speed. Compared to FairMOT and JDE, there is a slight increase in the calculations of the FPUAV. Although the tracking speed of our tracker decreases slightly, the speed of 17.6 FPS is still competitive. In summary, the FPUAV is an accurate and efficient tracker for real applications.

Table 3. Comparison with state-of-the-art methods on UAVDT.

Method	Params ($\times 10^6$)↓	Multi-Adds ($\times 10^9$)↓	FPS ↑
DSORT [57]	96.2	153.2	5.3
DAN [51]	103.9	164.7	4.9
JDE [33]	<u>67.2</u>	**67.7**	<u>17.8</u>
FairMOT [36]	**19.8**	<u>72.7</u>	**18.0**
FPUAV	69.3	73.6	17.6

4.5. Visualization

In this section, we visualize the tracking results of the FPUAV and analyze the qualitative results. We present some tracking results for the FPUAV and representative one-shot trackers JDE [33] and FairMOT [36] in Figure 10. As shown in Figure 10a, both JDE and FairMOT miss many targets in a scene with poor lighting conditions and a cluttered background. In this challenging scenario, the FPUAV successfully tracks multiple targets that are lost by JDE and FairMOT. There are a large number of extremely small targets that are difficult to locate in Figure 10c,d. In these videos, JDE has difficulty tracking most targets. Although FairMOT outperforms JDE, it still loses many distant targets. In these challenging scenarios, the FPUAV locates most small targets successfully. The tracking robustness for multi-scale targets benefits from the the FDN and the PTE. Specifically, the the FDN enables the network to learn representations suitable for efficient detection. The PTE enhances the robustness of the FPUAV for targets of different sizes by capturing fine-grained features.



(a) uav0000119_02301_v (b) uav0000188_00000_v

(c) uav0000201_00000_v (d) uav0000249_00001_v

Figure 10. Qualitative analysis of the FPUAV compared with JDE and FairMOT. Red arrows mark missing targets, and red boxes are areas where many targets are lost.

To illustrate that the FDN enables the FPUAV to predict more discriminative identity embedding, we visualize the embedding predicted by the FPUAV and JDE in Figure 11. Figure 11a shows the detections. The identity embeddings of the detections and existing tracklets are shown in Figure 11b,c. Figure 11d shows the predicted similarity among the detections. Figure 11e presents the similarity among existing tracklets. Figure 11f shows the similarity among the detections and the candidate tracklets. These similarities are obtained by calculating the cosine distances among the feature vectors. In Figure 11d–f, the red color indicates high similarity, while blue is the opposite. We observe many ambiguous matches in JDE. The high similarity among different targets misleads the subsequent data association. Compared to JDE, our tracker suppresses more ambiguous matches, which indicates that the FPUAV predicts more discriminative embedding.

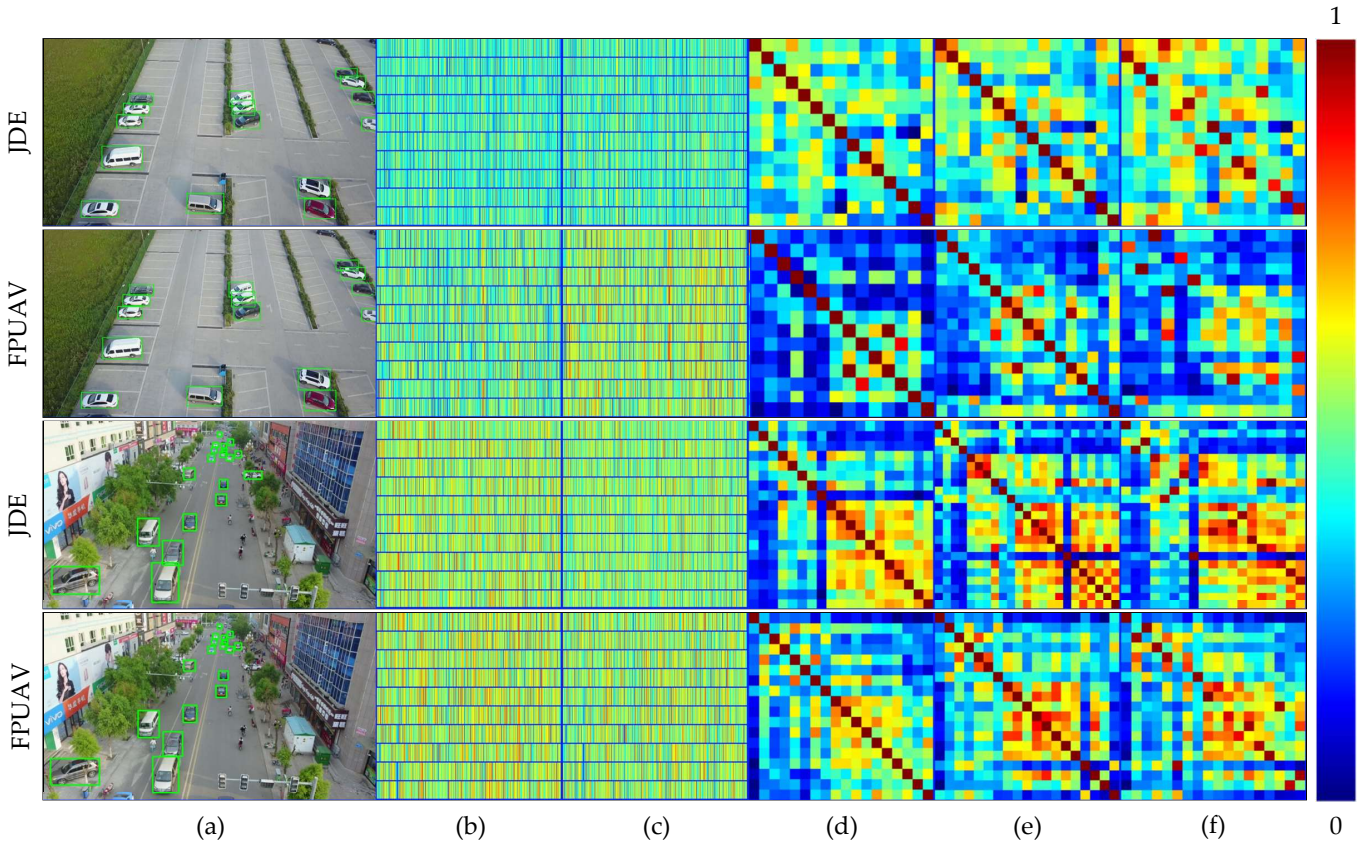


Figure 11. Identity embedding visualization for the FPUAV and JDE (**a**–**f**).

We visualize some tracking results of the FPUAV on VisDrone2021 and UAVDT benchmarks in Figures 12 and 13, respectively. These videos include various challenges in real applications, including crowded scenes, nighttime, viewpoint changes, distance changes, and multi-size targets. The FPUAV outputs the tracklets accurately under these challenging conditions. The tracking accuracy and robustness of the FPUAV satisfy the needs of real applications.

4.6. Ablation Study

In this section, we further verify the superiority of the FDN and the PTE by performing ablation experiments. For a fair comparison, all experiments are trained on VisDrone2021 training sets and tested on test sets. We consider the FPUAV with the FDN and the PTE removed as the baseline tracker. The experimental results are summarized in Table 4.

Table 4. Ablation analysis of the FPUAV on VisDrone2021.

Num	Baseline	FDN	PTE	MOTA ↑	IDF1 ↑	IDS ↓
①	√			27.6	39.4	3029
②	√	√		31.5	<u>43.9</u>	<u>2370</u>
③	√		√	<u>32.4</u>	41.8	2435
④	√	√	√	**34.3**	**45.0**	**2138**

The FDN aims to improve the ability to locate and identify targets by learning task-specific representations. When the baseline tracker is equipped with the FDN (② vs. ①), MOTA and IDF1 increase by 3.9% (31.5–27.6%) and 4.5% (43.9–39.4%), respectively. Meanwhile, IDS drops from 3029 to 2370. The tracker with the FDN removed (③ vs. ④) has a 1.9% (32.4–34.3%) decrease on MOTA and a 3.2% (41.8–45.0%) decline on IDF1.

Figure 12. Tracking results of the FPUAV on VisDrone2021.

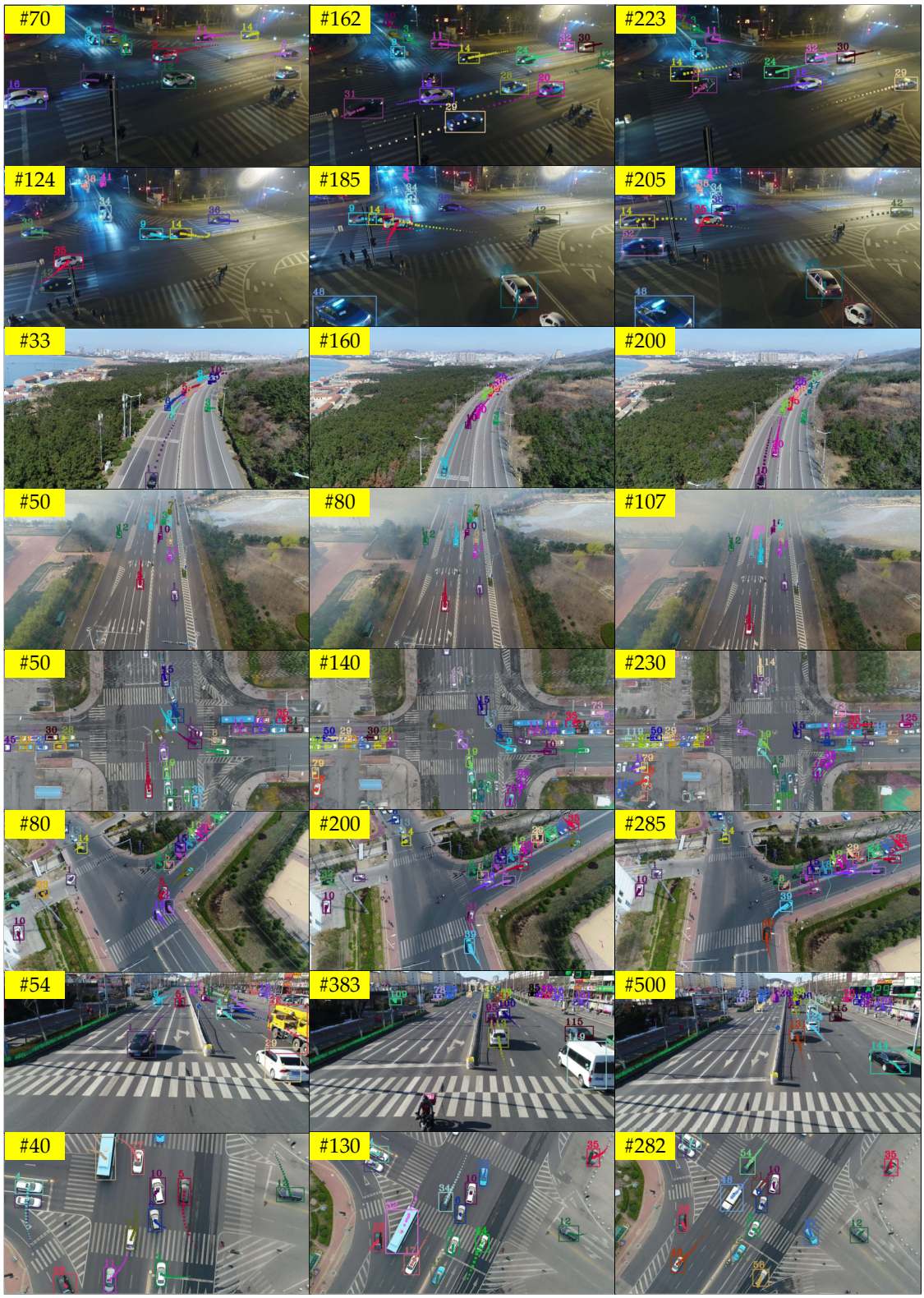


Figure 13. Tracking results of the FPUAV on UAVDT.

PTE is designed to capture fine-grained features for accurate localization. The PTE brings gains of 4.8% (32.4–27.6%) on MOTA and 2.4% (41.8–39.4%) on IDF1 for the baseline tracker (③ vs. ①). Meanwhile, IDS drops from 3029 to 2435. After removing the PTE from FPUAV (④ vs. ②), MOTA and IDF1 decrease by 2.8% (31.5–34.3%) and 1.1% (43.9–45.0%).

When we combine both the FDN and the PTE into the baseline tracker (④ vs. ①), FPUAV outperforms the baseline tracker by 6.7% (34.3–27.6%) on MOTA and 5.6% (45.0–39.4%) on IDF1. Above extensive ablation experiments demonstrate the effectiveness of the FDN and the PTE.

5. Conclusions

In this paper, we propose a novel FPUAV network for multiple object tracking in UAV videos. We observe that the optimization contradiction between detection and ReID limits the tracking performance of one-shot trackers. Therefore, we propose a feature decoupling network (FDN) to convert the original features into detection-specific and ReID-specific representations. The FDN effectively alleviates the conflict of multiple tasks within the network. In addition, existing one-shot trackers struggle to locate small targets in UAV videos accurately. Therefore, we propose a pyramid transformer encoder (PTE) to enrich the semantic information of detection-specific features. By learning scale-aware fine-grained features, the PTE enables the FPUAV to locate targets in UAV videos accurately. Extensive experiments on VisDrone2021 and UAVDT demonstrate that the FPUAV achieves state-of-the-art performance. In addition, we believe that the proposed FDN and PTE can be easily integrated into other one-shot trackers.

Author Contributions: Conceptualization, H.W.; methodology, H.W.; software, H.W.; validation, J.N. and Z.Z.; formal analysis, H.W. and J.N.; investigation, J.N.; resources, H.W.; data curation, H.W.; writing—original draft preparation, H.W.; writing—review and editing, H.W. and J.N.; visualization, H.W.; supervision, Z.Z.; project administration, Z.H.; funding acquisition, Z.H. and M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No.61571394 and No.62001149) and the Key Research and Development Program of Zhejiang Province (Grant 2020C03098).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Tian, Y.; Dehghan, A.; Shah, M. On Detection, Data Association and Segmentation for Multi-Target Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2146–2160. [CrossRef] [PubMed]
2. Fernandez-Sanjurjo, M.; Mucientes, M.; Brea, V.M. Real-Time Multiple Object Visual Tracking for Embedded GPU Systems. *IEEE Internet Things J.* **2021**, *8*, 9177–9188. [CrossRef]
3. Chen, T.; Pennisi, A.; Li, Z.; Zhang, Y.N.; Sahli, H. A Hierarchical Association Framework for Multi-Object Tracking in Airborne Videos. *Remote Sens.* **2018**, *10*, 1347. [CrossRef]
4. Wu, H.; Du, C.J.; Ji, Z.P.; Gao, M.Y.; He, Z.W. SORT-YM: An Algorithm of Multi-Object Tracking with YOLOv4-Tiny and Motion Prediction. *Electronics* **2021,** *10*, 2319. [CrossRef]
5. Wang, C.Y.; Su, Y.; Wang, J.J.; Wang, T.; Gao, Q. UAVSwarm Dataset: An Unmanned Aerial Vehicle Swarm Dataset for Multiple Object Tracking. *Remote Sens.* **2022**, *14*, 2601. [CrossRef]
6. Wan, X.Y.; Cao, J.K.; Zhou, S.P.; Wang, J.J.; Zheng, N.N. Tracking Beyond Detection: Learning a Global Response Map for End-to-End Multi-Object Tracking. *IEEE Trans. Image Process.* **2021**, *30*, 8222–8235. [CrossRef]
7. Sun, Z.H.; Chen, J.; Chao, L.; Ruan, W.J.; Mukherjee, M. A Survey of Multiple Pedestrian Tracking Based on Tracking-by-Detection Framework. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 1819–1833. [CrossRef]
8. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 386–397. [CrossRef]
9. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
10. Duan, K.W.; Bai, S.; Xie, L.X.; Qi, H.G.; Huang, M.M.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–21 November 2019.
11. Ciaparrone, G.; Sanchez, F.L.; Tabik, S.; Troiano, L.; Tagliaferri, R.; Herrera, F. Deep learning in video multi-object tracking: A survey. *Neurocomputing* **2020**, *381*, 61–88. [CrossRef]

12. Milan, A.; Schindler, K.; Roth, S. Multi-Target Tracking by Discrete-Continuous Energy Minimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2054–2068. [CrossRef]
13. Ren, W.H.; Wang, X.C.; Tian, J.D.; Tang, Y.D.; Chan, A.B. Tracking-by-Counting: Using Network Flows on Crowd Density Maps for Tracking Multiple Targets. *IEEE Trans. Image Process.* **2021,** *30*, 1439–1452. [CrossRef] [PubMed]
14. Xiang, J.; Xu, G.H.; Ma, C.; Hou, J.H. End-to-End Learning Deep CRF Models for Multi-Object Tracking Deep CRF Models. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 828. [CrossRef]
15. Peng, J.L.; Wang, T.; Lin, W.Y.; Wang, J.; See, J.; Wen, S.L.; Ding, E.R. TPM: Multiple object tracking with tracklet-plane matching. *Pattern Recognit.* **2020**, *107*, 107480. [CrossRef]
16. Brendel, W.; Amer, M.; Todorovic, S. Multiobject tracking as maximum weight independent set. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1273–1280.
17. Huang, C.; Wu, B.; Nevatia, R. Robust Object Tracking by Hierarchical Association of Detection Responses. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 788–801.
18. Shu, G.; Dehghan, A.; Oreifej, O.; Hand, E.; Shah, M. Part-based Multiple-Person Tracking with Partial Occlusion Handling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1815–1821.
19. Yu, F.W.; Li, W.B.; Li, Q.Q.; Liu, Y.; Shi, X.H.; Yan, J.J. POI: Multiple Object Tracking with High Performance Detection and Appearance Feature. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 36–42.
20. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; p. 28.
21. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
22. Son, J.; Baek, M.; Cho, M.; Han, B. Multi-Object Tracking with Quadruplet Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3786–3795.
23. Lee, S.; Kim, E. Multiple Object Tracking via Feature Pyramid Siamese Networks. *IEEE Access* **2019**, *7*, 8181–8194. [CrossRef]
24. Lin, T.Y.; Dollar, M.; Girshick, M.; He, K.M.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3786–3795.
25. Bewley, A.; Ge, Z.Y.; Ott, L.; Ramov, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
26. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Fluids Eng.* **1960**, *82*,35–45. [CrossRef]
27. Chen, L.; Ai, H.Z.; Zhuang, Z.J.; Shang, C. Real-Time Multiple People Tracking with Deeply Learned Candidate Selection and Person Re-Identification. In Proceedings of the IEEE International Conference on Multimedia and Expo, San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
28. Zhou, H.; Ouyang, W.L.; Cheng, J.; Wang, X.G.; Li, H.S. Deep Continuous Conditional Random Fields With Asymmetric Inter-Object Constraints for Online Multi-Object Tracking. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 1011–1022. [CrossRef]
29. Shan, C.B.; Wei, C.B.; Deng, B.; Huang, J.Q.; Hua, X.S.; Cheng, X.L.; Liang, K.W. Tracklets Predicting Based Adaptive Graph Tracking. *arXiv* **2020**, arXiv:2010.09015.
30. Girbau, A.; Giró-i-Nieto, X.; Rius, I.; Marqués, F. Multiple Object Tracking with Mixture Density Networks for Trajectory Estimation. *arXiv* **2021**, arXiv:2106.10950.
31. Lit, Z.; Cai, S.Z.; Wang, X.Y.; Shao, H.Y.; Niu, L.; Xue, N. Multiple Object Tracking with GRU Association and Kalman Prediction. In Proceedings of the International Joint Conference on Neural Networks, Shenzhen, China, 18–22 July 2021; pp. 1–8.
32. Voigtlaender, P.; Krause, M.; Osep, A.; Luiten, J.; Sekar, B.B.G.; Geiger, A.; Leibe, B. MOTS: Multi-Object Tracking and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7934–7943.
33. Wang, Z.D.; Zheng, L.; Liu, Y.X.; Li, Y.L.; Wang, S.J. Towards Real-Time Multi-Object Tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
34. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2021**, arXiv:1804.02767.
35. Kendall, A.; Gal, Y.; Cipolla, R. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7482–7491.
36. Zhang, Y.F.; Wang, C.Y.; Wang, X.G.; Zeng, W.J.; Liu, W.Y. FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking. *Int. J. Comput. Vision* **2021**, *129*, 3069–3087. [CrossRef]
37. Meng, F.J.; Wang, X.Q.; Wang, D.; Shao, F.M.; Fu, L. Spatial-Semantic and Temporal Attention Mechanism-Based Online Multi-Object Tracking. *Sensors* **2020**, *20*, 1653. [CrossRef] [PubMed]
38. Liu, X.H.; Luo, Y.C.; Yan, K.D.; Chen, J.F.; Lei, Z.Y. Part-MOT: A multi-object tracking method with instance part-based embedding. *IET Image Proc.* **2021**, *15*, 2521–2531. [CrossRef]
39. Yan, Y.C.; Li, J.P.; Qin, J.; Liao, S.C.; Yang, X.K. Efficient Person Search: An Anchor-Free Approach. *arXiv* **2021**, arXiv:2109.00211.

40. Du, P.F.; Wen, L.Y.; Du, D.W.; Bian, X.; Fan, H.; Hu, Q.H.; Ling, H.B. Detection and Tracking Meet Drones Challenge. *arXiv* **2021**, arXiv:2001.06303v3.
41. Du, D.; Qi, Y.K.; Yu, H.Y.; Yang, Y.F.; Duan, K.W.; Li, G.R.; Zhang, W.G.; Huang, Q.M.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 1141–1159.
42. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
43. Wen, L.Y.; Du, D.W.; Cai, Z.W.; Lei, Z.; Chang, M.C.; Qi, H.G.; Qi, H.G.; Lim, J.; Yang, M.H.; Lyu, S.W. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Comput. Vision Image Underst.* **2020**, *193*, 102907. [CrossRef]
44. Dendorfer, P.; Osep, A.; Milan, A.; Schindler, K.; Cremers, D.; Reid, I.; Roth, S.; Leal-Taixe, L. MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking. *Int. J. Comput. Vis.* **2020**, *129*, 845–881. [CrossRef]
45. Wu, B.; Nevatia, R.T. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. *Int. J. Comput. Vis.* **2007**, *75*, 247–266. [CrossRef]
46. Bernardin, K.; Stiefelhagen, R. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP J. Image Video Process.* **2008**, *1*, 246309. [CrossRef]
47. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance Measures and a Data Set for Multi-target, Multi-camera Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 17–35.
48. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
49. Bochinski, E.; Eiselein, V.; Sikora, T. High-Speed Tracking-by-Detection Without Using Image Information. In Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance, Lecce, Italy, 29 August–1 September 2017.
50. Pirsiavash, H.; Ramanan, D.; Fowlkes, C. Globally-Optimal Greedy Algorithms for Tracking a Variable Number of Objects. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1201–1208.
51. Sun, S.J.; Akhtar, N.; Song, H.S.; Mian, A.; Shah, M. Deep Affinity Network for Multiple Object Trackin. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 104–119.
52. Zeng, F.A.; Dong, B.; Wang, T.C.; Chen, C.; Zhang, X.Y.; Wei, Y.C. Motr: End-to-end multiple-object tracking with transformer. *arXiv* **2021**, arXiv:2105.03247.
53. Meinhardt, T.; Kirillov, A.; Leal-Taixé, L.; Feichtenhofer, C. TrackFormer: Multi-Object Tracking With Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LO, USA, 19–23 June 2022; pp. 8844–8854.
54. Milan, A.; Roth, S.; Schindler, K. Continuous energy minimization for multitarget tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013** *36*, 58–72. [CrossRef]
55. Dicle, C.; Camps, O.I.; Sznaier, M. The Way They Move: Tracking Multiple Targets with Similar Appearance. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 2304–2311.
56. Bae, S.H.; Yoon, K.J. Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1218–1225.
57. Wojke, N.; Bewley, A.; Paulus, D. Simple Online and Realtime Tracking with a Deep Association Metric. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 3645–3649.
58. Xiang, Y.; Alahi, A.; Savarese, S. Learning to Track: Online Multi-Object Tracking by Decision Making. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4705–4713.