Multimedia Tools and Applications (2021) 80:13911–13929
https://doi.org/10.1007/s11042-020-10427-1

Multiple objects tracking in the UAV system based on hierarchical deep high-resolution network

Wei Huang [1,2] · Xiaoshu Zhou [3] · Mingchao Dong [3] · Huaiyu Xu [1,2,3]

Received: 14 June 2020 / Revised: 23 September 2020 / Accepted: 22 December 2020 /
Published online: 19 January 2021

Abstract
Robust and high-performance visual multi-object tracking is a big challenge in computer vision, especially in a drone scenario. In this paper, an online Multi-Object Tracking (MOT) approach in the UAV system is proposed to handle small target detections and class imbalance challenges, which integrates the merits of deep high-resolution representation network and data association method in a unified framework. Specifically, while applying tracking-by-detection architecture to our tracking framework, a Hierarchical Deep High-resolution network (HDHNet) is proposed, which encourages the model to handle different types and scales of targets, and extract more effective and comprehensive features during online learning. After that, the extracted features are fed into different prediction networks for interesting targets recognition. Besides, an adjustable fusion loss function is proposed by combining focal loss and GIoU loss to solve the problems of class imbalance and hard samples. During the tracking process, these detection results are applied to an improved DeepSORT MOT algorithm in each frame, which is available to make full use of the target appearance features to match one by one on a practical basis. The experimental results on the VisDrone2019 MOT benchmark show that the proposed UAV MOT system achieves the highest accuracy and the best robustness compared with state-of-the-art methods.

Keywords Multi-object tracking · UAV · HDHNet · Fusion loss

1 Introduction

With the increasing popularity of commercial unmanned aerial vehicles (UAV) and the rise of computer vision as well as artificial intelligence technology, the tracking algorithms based on

✉ Wei Huang
huangw@sari.ac.cn

[1] Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[3] ShanghaiTech University, Shanghai 201210, China

drones have become an active research area. Efficient video image processing algorithms and complex deep neural networks have made it easier to realize auto-navigation, campus security monitoring, and disaster relief.

Deep visual object tracking [14, 23, 29] can be divided into single-object tracking (SOT) and multi-object tracking (MOT). SOT algorithms need to initialize a box of one target in the first video frame, then keep tracking the object until it disappears from view. MOT is an important computer vision task designed to analyze videos to maintain the trajectories of all interesting targets without the need to know target appearances and numbers in advance [28]. These works try to explain the connections and differences in tracking algorithms based on deep learning and show the importance of that in visual tracking. Algorithms of MOT are mainly divided into online-tracking and offline-tracking; the former only uses current and past information to track targets about the current frame; the latter, on the contrary, is able to exploit global information while trying to determine the target identities in a specific frame. Compared with offline-tracking methods, online-tracking methods without using future information tend to get worse robustness and performance features. Following the tracking-by-detection paradigm in most cases [46, 47, 49], MOT receives the detection results of each frame before inputting and associating them as the final trajectories. Therefore, many MOT algorithms describe the task as a data- association [3, 47, 52, 57] issue.

However, due to the fact that the drone usually flies at high altitude, UAV tracking suffers the impact brought by platform motion and image instability, such as target aspect ratio change, viewpoint change, fast movement, scale change, target occlusion, target loss, etc. Some end-to-end and supervised advanced tracking algorithms have been proposed by many scholars, which can effectively alleviate the tracking problems for drones mentioned above. Notably, top trackers usually use Cascade R-CNN [5] to generate detections in individual frames and integrate the temporal information, such as IoU tracker [4] and FlowNet [40] to complete the assignment. Similarly, some researchers combine IoU tracker [4], CenterNet [56], and DaSiameseRPN [58] for multiple object tracking. Unfortunately, these methods are limited to the applications of larger pedestrian and car detection, etc., though long-term cues are well applied during the tracking. SORT [3] and DeepSORT [47] are referred as linear uniform velocity models, which is independent from camera motion and object categories [21], thus degrading its performance to some extent. In this paper, the proposed drones MOT framework is based on tracking-by-detection schema, which strongly depends on a precise and effective detector. However, current MOT methods generally adopt a deep convolution neural network, such as Faster RCNN [35], RetinaNet [27] and ResNet [18], etc., to detect targets in each frame, which has missed a lot of high-resolution features available with the network layer deepening, especially in a drone scenario. Some algorithms applied HRNet [41] to solve small target detection issues, which have high recognition precision, but low frame rate.

In view of the challenges above, this paper is dedicated to improving the UAV MOT system's efficiency. A network is investigated, which has not only high-resolution feature but also faster detection speed. The network is divided into two parts: a deep high-resolution feature extract module and the prediction network. The former is a combination of High-resolution Representation network (HRNet) [41] and Hierarchical Deep Aggregation network(HDA) [51], which is called Hierarchical Deep High-resolution network (HDHNet). The latter can be a variety of CNN networks, such as Faster RCNN [39], Cascade RCNN [5], or Hybrid Task Cascade(HTC) [7] based on Region Proposal Network(RPN) [39]. We will compare various prediction networks' performance and experiment with the effects of different resolution videos on the detector in the experimental section.

The contributions of our work are summarized as follows:

1) It is difficult for traditional MOT approaches based on the tracking-by-detection paradigm to handle small targets detection or tracking in UAV scenes. The Hierarchical Deep High-resolution network (HDHNet) and an end-to-end framework are proposed for the feature extraction of small targets. It is faster and comparably accurate as an online MOT system compared with other one-stage or two-stages state-of-the-art detection methods.
2) Due to the imbalance between positive samples and negative samples in the UAV scene, especially for some hard samples, existing methods are difficult to deal with these problems simultaneously, which usually cause the model's overfitting. So an adjustable loss function fusing on focal loss [27] and GIoU [37] loss is proposed to train our model in the UAV dataset, which is available to alleviate these problems.
3) We investigate a flexible MOT algorithm on UAV scenes, integrating the detection and tracking modules in a unified framework. The former consists of HDHNet, the backbone network, and different prediction networks, while the latter adopts an improved DeepSort algorithm. And we are the first to introduce the high-resolution network to UVA video with the objective to address the issue of small target detection and tracking, and apply different prediction networks to our detection framework. Experiments on the Visdrone2019 [21] MOT dataset show that our method has the advantages over state-of-the-art MOT systems in terms of training data volume, speed, and accuracy.

2 Related work

2.1 Single object tracking

Based on feature extraction and filtering search methods, such as KCF [20], CSK [19], SACK [10], etc., most traditional SOT algorithms adopt correlation filtering for tracking, which is fast but difficult to deal with the problem of occlusion. [9, 12] proposed the method of fusion Saliency Detection and correlation filtering to make multi-scale target Detection and tracking performance more robust. With the development of deep neural networks, computer vision enjoys a rapid growth of deep learning. [54] apply cascade R-CNN based on multi-scale attention and imbalanced samples to improve detector performance. [11] employ dual-channel CNN to solve image super-resolution. These methods are very beneficial in handling the problem of object detection and tracking. Besides, SiamFC [2] with its stable overtime tracking rate and accuracy for target-tracking algorithms based on the siamese network, is characterized by wide attention and application. SiamRPN [24] is joined to the RPN [39] in the siamese network, therefore, the original similarity calculation problem is converted to the issues of classification and regression, thus further improving the tracking accuracy. SiamRPN++ [25] has applied networks, such as ResNet [18] and Inception [42], as the first deep benchmark networks, to tracking networks based on siamese networks, thus significantly improving network feature extraction and tracking performance. Drones need a small number of algorithm parameters, less memory, and short inference time to realize real-time target detection due to the weak hardware computing power. Being often difficult to be applied to standard algorithms, SlimYOLOv3 [53] prunes the improved version of YOLOv3. It achieves a better detection accuracy than the original algorithm on UAV target detection data set under the condition in which the number of parameters, memory consumption, and inference time are significantly reduced.

2.2 Multiple object tracking

The standard method used in the MOT algorithm is based on the tracking-by-detection paradigm [3, 13, 47, 57]. A single object tracker (SOT) is applied by Chu et al. [13] to the MOT framework, which utilizes the advantages of SOT in adapting appearance features and detecting targets. Xu et al. [49] propose a deep Hungarian network to solve optimal soft-assignment by recurrent neural networks. Zhu et al. [57] propose a spatial attention network, namely DMAN, for target occlusion and noisy detections, [16, 45, 55] introduce the image-based and video-based attention mechanism to saliency object detection, which can to some extent be used for object tracking as well. [44] presents a baseline method, namely TrackR-CNN, to implement detection, segmentation, and tracking jointly. In [46], a nearly real-time MOT system is proposed to facilitate learning object detection and embedding simultaneously based on a shared network. Recently, [6, 36] adopt the reinforcement learning method to predict the position of targets in the next frame, while [36] each target is regarded as an agent detected by the prediction network; after that, the decision network is used to search the best tracking result according to the association between multiple target agents and detection results. A universal pipeline of the UAV MOT system is illustrated in Fig. 1, which consists of four parts: (1) Input videos, which can be drone videos of different resolutions; (2) Feature extractor, which extracts image features through a deep neural network; (3) Object detection, which connects a prediction network to detect all interesting targets; (4) Object tracking, which will be used to match all targets to form trajectories on all video frames.

2.3 High-resolution representations

As a crucial step in computer vision tasks, extracting strong feature representation is available to determine the quality of the results directly. Different strided convolutions are used to lower the later layer size by exploiting high-to-low resolution networks, such as VGGNet [39], ResNet [18], and DenseNet [22]. Encoder-decoder [33], U-Net [38], and Hourglass network [15, 32, 50] apply high-to-low and low-to-high resolution representations during their feature extraction process, which leverage down-sample and up-sample subnets to output the features of different layers. Recently, high-resolution representation has achieved great success in pose estimation, semantic segmentation, and object detection, etc. The HRNet [41] can maintain a high-resolution representation throughout the process, whose four stages are illustrated in

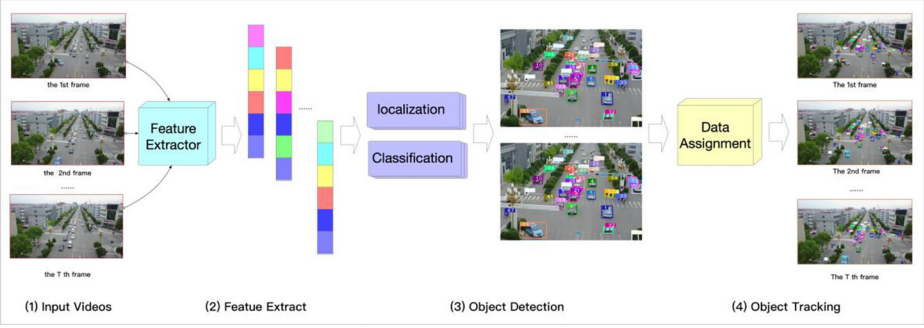


Fig. 1 An illustration of the MOT algorithm process. The algorithm steps are: (1) Input a video; (2) Extract all object features; (3) Detect all objects' localization and categories; (4) Tracking by data assignment

Fig. 2. The HRNet [41] can gradually add high-to-low resolution subnets composed by multi-scale group convolutions to build more stages and connect multiple resolution subnets in parallel. Besides, MMDetection [8] has integrated different versions of HRNet [41] to address various visual tasks, including small target detection.

2.4 Data association

Data association is a significant step for all MOT algorithms based on tracking-by-detection schema, and [34, 43, 52] formulate the process of data association as various optimization problems. Through modeling, the MOT problem is converted to a problem of bipartite graph assignment. [57] focuses on the use of target positions and movement. While Zhang et al. [52] regard the MOT problem as a MAP- data-association one and implement optimization using the global min-cost network flow. In the online tracking algorithm, the current trajectory and detection target are taken as vertex sets, respectively, and then the nodes can be connected between the two vertex sets. The weights of adjacent edges can be calculated and obtained based on appearance models, motion models, or other measurement models representing the similarity or connection cost between nodes. For the problem of bipartite graph matching, the Hungarian algorithm [30] is almost the core algorithm of bipartite graph matching, except for bipartite graph multiple matching.

3 Methods

3.1 Pipeline

Our entire MOT framework is illustrated in Fig. 3. The algorithm runs in the following procedure. Firstly, all detections are obtained from the proposed feature extractor HDHNet (Sec. 3.2). After that, different prediction networks are performed on the HDHNet (Sec. 3.3), which generates an enlarged detection set $\{D_1', D_2', ..., D_N'\}$. Thirdly, a fusion lost function is proposed to train our model (Sec. 3.4). Finally, the MOT algorithm is applied in the new detection set (Sec. 3.5), which will output these targets' trajectories.

3.2 Hierarchical deep high-resolution network

The HRNet [41] maintains high-resolution representations by concatenating high-to-low resolution convolutions in parallel with repeated multi-scale fusions across parallel

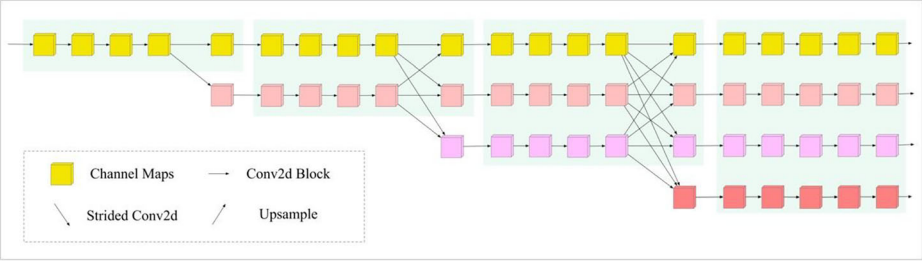


Fig. 2 The architecture of a high-resolution network, which has four stages. In each stage, there are convolution blocks with different resolutions that constitute multi-scale group convolution

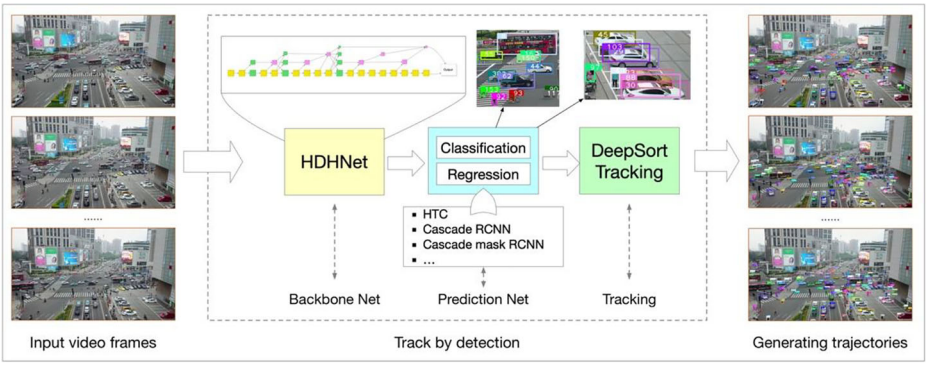


Fig. 3 A pipeline of our tracking framework which refers to track-by-detection architecture. The steps are as follows: (a) Extract features by proposed HDHNet and detect bounding boxes by HTC, Cascade RCNN, etc.; (b) Use improved Deepsort [47] tracking algorithm to track all the targets; (c) Generate trajectories for all targets

convolutions. However, every stage of HRNet has many parameters, which results in a slow extraction process. In this paper, an original and comprehensible feature extraction network HDHNet is proposed, and the inspiration is from HRNet [41] and DLA network [51]. The former gets robust feature representation on input images, and the latter reduces the network parameters through the sparse multi-scale network structure and improves the network speed. Aggregation and transition are defined as the combination of different layers throughout a network, and ResNet [18] block is introduced to each layer of HDHNet. Next, our proposed feature extraction network HDHNet in detail will be presented in detail.

Architecture The HDHNet architecture is illustrated in Fig. 4. There are four stages, the 1st stage contains no aggregation blocks but only two convolution blocks. The 2nd stage to the 4th stage contains 1, 2, 4 aggregation blocks and 2, 4, 8 convolution blocks, respectively. The 1st stage consists of two ResNet blocks with high-resolution, and the 2nd, 3rd, and 4th stages consist of repeated modular multi-resolution blocks and aggregation blocks of different scales.

At each stage, aggregation blocks are integrated with high-resolution blocks, intermediate aggregation blocks and the previous stage's transition blocks. A multi-resolution group

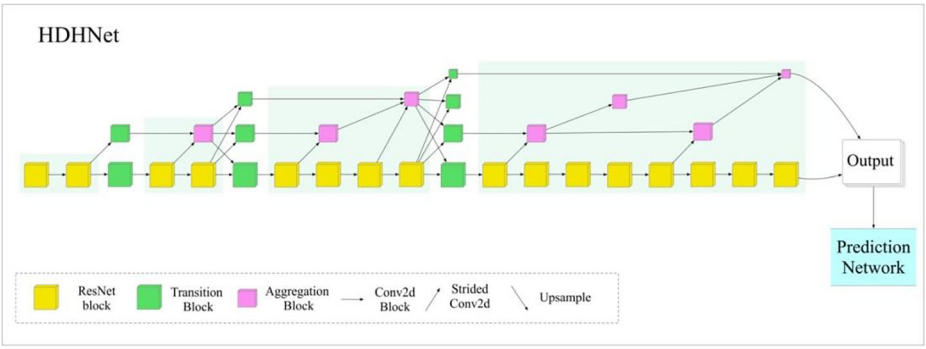


Fig. 4 The proposed feature extractor Hierarchical Deep High-resolution network (HDHNet). There are four stages. The 1st stage contains high-resolution convolution (as the yellow blocks), and the 2nd (3rd, 4th) stage keeps the same resolution as the 1st stage, which is connected by a group of transition modules (green blocks) and adding aggregation blocks (pink block) to different layers of different stages

convolution with aggregation is illustrated in Fig. 5(a) where the yellow and pink blocks denote the high-resolution features and aggregations with different scales respectively. The transition module is shown in Fig. 5(b) composed of ResNet blocks (the green blocks) of different scales. Different blocks in various stages are aggregated to obtain richer deep semantic information. The multi-resolution group convolution is a simple extension of group convolution, which divides the input channel into multiple channel subsets. It performs regular 3*3 convolution on each subset under different spatial resolutions. The multi-branch fully connected mode is similar to regular convolution, as shown in Fig. 5(c). The input and output channels are divided into multiple subsets connected in a fully connected manner.

Instantiation The network HDHNet is instantiated by composing a High-resolution network (HRNet) [41] and a Deep Layer Aggregation network (DLA) [51]. Before the network, two 3*3 strided convolutions are used to reduce the resolution to 1/4, thus getting 64-channel feature maps. After that, the dimension of the feature map is reduced to 48 channels with 1×1 convolution. HDHNet has four stages, as described in Fig. 4, the 1st stage has two residual blocks with 48 input channels, and the 2nd, 3rd, and 4th stages have 2, 4, and 8 high-resolution residual units, respectively. Besides, the aggregation block's resolution decreases by two times while the number of channels upsampled (two times) at different levels of stages, whose channels are 48, 96, 192, and 384, respectively.

Advantages HDHNet is characterized by three advantages: firstly, HDHNet connects multi-resolution sub-networks from high to low in parallel instead of connecting in series. Therefore, high resolution can be maintained throughout the entire process, rather than reverting from high resolution to low resolution. Secondly, the existing fusion schemes leverage low-resolution and high-resolution feature information in most cases, while the repeated multi-scale fusion method in HDHNet network employs the same depth and similar level of low-resolution feature representation to improve the high-resolution feature representation, which makes the high-resolution feature representation much stronger. Thirdly, HDHNet learns from hierarchical deep aggregation blocks, which effectively improve the network speed.

3.3 Prediction network

Taking advantage of the multi-scale high-resolution feature representations extracted in HDHNet (Sec 3.2), a prediction network is established to predict targets' location and category

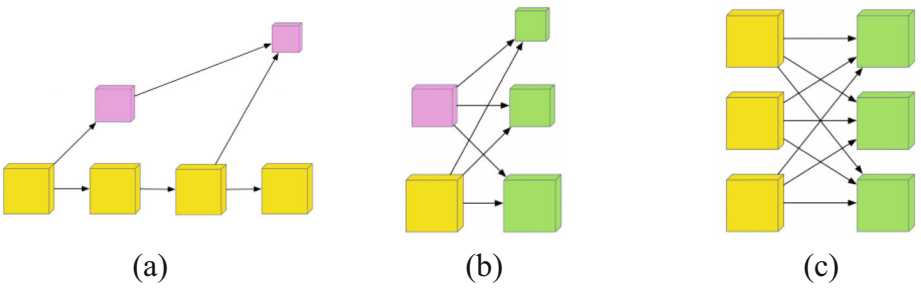


Fig. 5 Block of different resolution: (**a**) multi-resolution group convolution with aggregation and (**b**) multi-resolution transition convolution (**c**) the multi-branch fully connected model

in drone videos. In the recent past, significant progress has been made in multiple-object tracking by utilizing different effective prediction networks. Many research works show that an expressive performance of prediction networks strongly relies on the comprehensive, practical, and deep features extracted by the backbone network. Implementing the end-to-end trainable HDHNet is followed by a prediction network, which can effectively identify the video's targets. In later experiments, the performances of the backbone network will be compared and evaluated empirically with various prediction networks, after that, the specific prediction networks applied in this paper will be introduced in detail.

Convolutional neural network (CNN) CNN is adopted as the first prediction network of our framework, whose architecture is depicted in Fig. 6. It has two branches, consisting of five 3*3 convolutional layers and adopt Relu as the activation function, while being trained simultaneously without sharing weights. The classification model is used to output the positive sample target category, whose last layer output channel is 81 * A, where A represents the number of anchors of different scales. In our experiment based on the pre-training weights of the COCO [26] dataset, nine types of the anchor with three kinds of scales are employed respectively, i.e., 1: 2, 1: 1, and 2: 1. There are 81 categories, including 80 categories from the COCO dataset and an extra background class. Differently, the regression model outputs the position and bounding box of each positive sample target with its last convolutional layer output channel of 4 * A, where 4 represents the prediction of x, y, w, and h.

Cascade RCNN [5] As a powerful and classic architecture, the cascade can greatly improve the performance of multi-tasks, as shown in Fig. 7(a), which is a derivation of the R-CNN [17] and consists of a series of trained detectors with increasing intersection-over-union (IoU) thresholds. Sequential training is performed on the cascade R-CNN stages, and the output of one stage is used to train the next stage. This resampling gradually improves the hypothesis's quality, thus ensuring that the size of the positive training set for all detectors is equal. The motivation is to

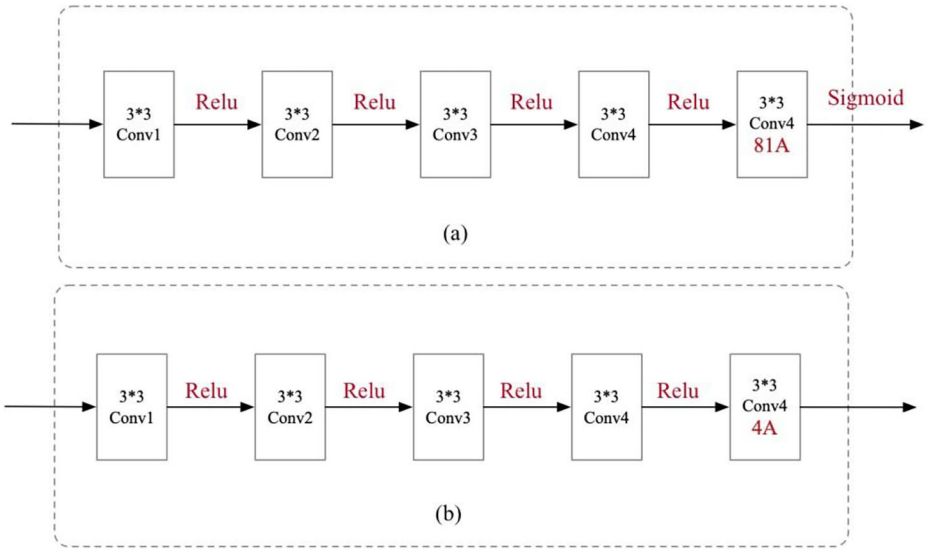
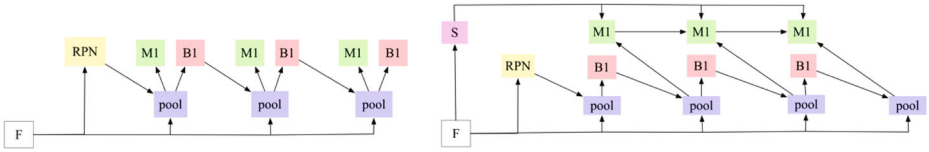


Fig. 6 The CNN prediction network contains (**a**) Classification and (**b**) Regression model



(a) (b)

Fig. 7 The architecture of (a) Cascade R-CNN and (b) HTC

observe that the output IoU of a regression variable is almost always better than the input IoU. In our experiments, the input IoU of the three stages is set as [0.5, 0.6, 0.7], which has been proven to effectively improve the robustness and accuracy of network detection.

Hybrid task Cascade (HTC) [7] HTC is mainly used in instance segmentation tasks, and the box and mask branches of each stage run alternately during the training process by designing a multi-task and multi-stage hybrid cascade structure. Information flow is directly added between the mask branches of different stages, and a branch of semantic segmentation is merged to enhance the context information. The architecture of HTC is demonstrated in Fig. 7(b), while in our experiments, HTC is also used as the prediction module for the detection network. Compared with ordinary CNN prediction networks, though the detection speed is relatively slow, the performance has been dramatically improved. Besides, the results of different prediction networks will be shown in the experimental part later.

3.4 Learning

Our proposed HDHNet is trained in an end-to-end manner, and in Sec. 3.2 and Sec. 3.3, a target detection network is introduced based on the HDHNet, which needs to output the position and category of the target in the picture separately. It is a multi-task prediction model trained with a multi-task fusion loss function combined by classification and localization loss. Furthermore, the adjustable hyper-parameter λ is applied for balancing the two loss functions, where λ is generally set as 0.5. After that, the process of establishing our loss function will be introduced.

Classification of loss Cross Entropy loss (CE loss) is generally used in the classification loss calculation in the detection task, as shown in Formula (1). However, for a single detection network, candidate frames are not generated in advance. Instead, a single detection network generates many simple negative samples (mainly belong to the background), which will mainly contribute to the loss value and dominate the gradient update direction. There are many such simple negative samples, especially in drone data sets. If CE loss is used to train the entire classification network, the network performance will be inferior. Therefore, to address the class imbalance problem, CE loss is replaced with the focal loss [27], which is defined as follows:

$$CE(\hat{y}) = -\frac{1}{n}\sum_{k=1}^{n}\left[I\{y = y^{'}\}\log(p_t)\right] \tag{1}$$

$$p_t = \{p \quad if \;\; y = 1 1-p \; otherwise \tag{2}$$

The formula is simplified as:

$$CE(p_t) = -\alpha_t \log(p_t) \tag{3}$$

$$L_{cls} = FL(p_t) = -\alpha_t (1-p_t)^{\gamma} \log(p_t) \tag{4}$$

Where the parameter n denotes the number of samples, p_t denotes the probability of the category prediction, and I represents the indicator function. Here are calculations of the loss of a single category. The calculation of losses of different categories is independent of each other and satisfies independent distribution. Multiplying the coefficient $(1 - p_t)^{\gamma}$ can effectively alleviate the impact of class imbalance to some extent.

Localization of loss Another detection task is to locate these targets in drone videos, and bounding boxes are usually used to represent them. The main indicator for measuring the performance of target detection is cross-ratio IoU, as shown in Formula (5). However, this type of loss function does not always reflect the positioning accuracy. Although IoU is furnished with scale invariance, it also has two problems, that is, firstly, in the case that the two boxes of A and B do not intersect, in other words, when IoU = 0, the distance (or similarity) of the two boxes cannot be reflected at this time, the loss function does not have a gradient at this time, and it cannot be trained with gradient descent method.

Secondly, even if A and B share the same IoU, it doesn't mean that the detection frame's positioning effect is the same. As shown in Fig. 8, the IoU sizes in a, b, and c are the same, while the detection results obviously show different overlapping relationships between A and B, the position offset, especially in C, is large. This type of target detection in the drone scene is more obvious since the drones' shooting angle generally changes during their flight. If IoU is calculated directly, it will be easy to bring an impact on the positioning accuracy of the target.

$$\mathrm{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

GIoU (Generalized Intersection over Union) [37] is used to calculate the similarity between two bounding boxes, which overcomes the shortcomings of IoU, and the calculation is expressed as follows:

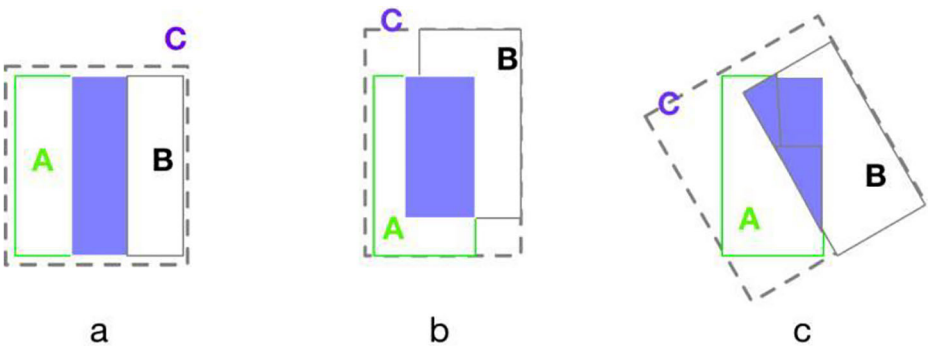


Fig. 8 The same IoU values with different detection bounding boxes

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|} \tag{6}$$

The GIoU loss function is defined as follows:

$$L_{reg} = L_{GIoU} = 1 - GIoU \tag{7}$$

Fusion loss During the multi-task training process, the loss function of different tasks on the overall performance is adjusted by the weight coefficient λ. The following fusion loss functions are utilized in this paper:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, p_i^*\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\left(t_i, t_i^*\right) \tag{8}$$

Where L_{cls} and L_{reg} refer to the classification loss and GIoU loss respectively, i denotes an anchor index of the mini-batch, p_i represents the prediction probability of the target, and p_i* represents the ground truth box. In the case that it is a positive sample, p_i* is equal to 1; otherwise, it's equivalent to 0. t_i and t_i* represent the prediction box's location and the ground truth box's location, respectively.

3.5 Multiple object tracking

Our multi-object tracking framework follows the tracking-by-detection schema based on online-tracking, whose algorithm process is illustrated in Fig. 9. Firstly, as the aforementioned proposed extractor HDHNet, input UAV videos frames $\{F_1, F_2, \ldots, F_n\}$, which extracts features $\{f_1, f_2, \ldots, f_n\}, f_n \in R^{W*W*D}$, then generate object bounding boxes by different prediction net. Secondly, initialize parameters, including the maximum number of unmatched frames A_max=90 and the minimum number of matched frames n_init = 3. Bounding box detection confidence is set to 0.65, Non-Maximum Suppression (NMS) [31] threshold =0.85. Thirdly, the detection boxes are filtered according to the confidence degree, which are determined by the confidence of detection during tracking and initialized bounding box detection confidence. If the former is lower, the bounding box is deleted. And then NMS [31] was carried out on all Bounding boxes, and the borders whose coincidence degree was higher than the NMS

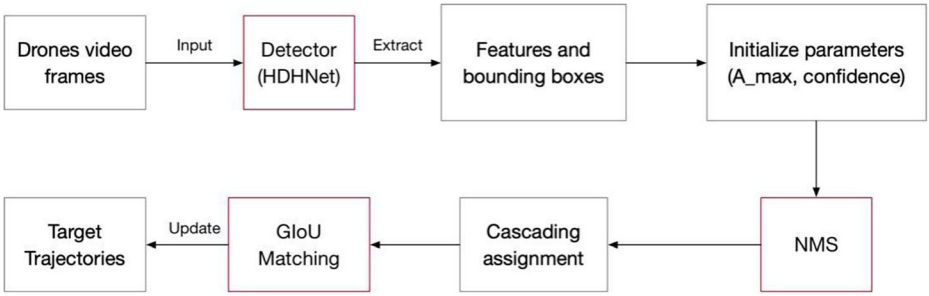


Fig. 9 The process of our tracking model based on DeepSORT [47]

threshold were deleted, that is, the situation of eliminating multiple boxes on a target was eliminated. Next, cascade matching of all trajectories and detection results. Finally, GIOU matching for unmatched track and detection, then generating the final trajectories of UAV videos. And the proposed algorithm based on DeepSORT [47] is described as follows:

Algorithm 1: The improved Deepsort algorithm

TrackerUAV $(F, Model_{HDHNet})$;
Input : UAV video contains N frames $F = \{F_1, F_2, F_3, ..., F_N\}$
the trained model HDHNet.
Output: All trajectories of UAV video $T = \{T_1, T_2, T_3, ..., T_M\}$
initialize:
The maximum number of unmatched frames $A_{max} = 90$
The minimum number of matched frames $n_{init} = 3$
Bounding box detection confidence $cof = 0.65$
Non-Maximum Suppression threshold $T_{nms} = 0.85$
Trajectories $T = \emptyset$
for $i = 1, 2, ..., N$ **do**
  detection:object detection by HDHNet
  $BB_i = Model_{HdhNet}(F_i)$,
  tracking:
  $BB_i \leftarrow \{bb_j | bb_j \in BB_i, Cof_{bb_j} \geq cof\}$ $BB_i \leftarrow NMS(BB_i)$
  $T \leftarrow T \cup MatchingCascade(T, BB_i)$
  $T \leftarrow T \cup GIoUMatching(T, BB_i)$
end

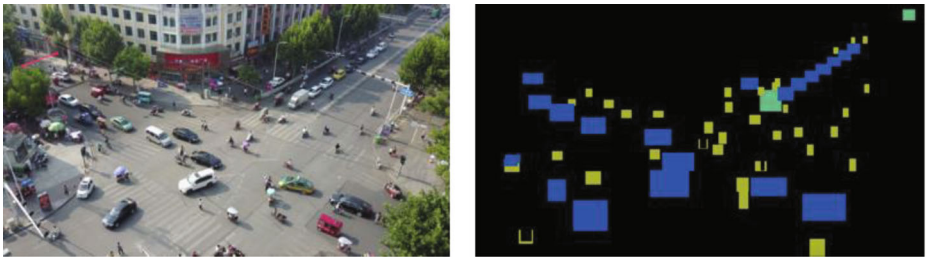
4 Experiments

In this part, an extensive evaluation of the proposed MOT framework based on HDHNet is demonstrated. The hardware environment of this experiment is characterized by Inter Core i7-6500 k CPU 3.4GHz, and two GPUs of TITAN RTX 24G memory and the software environment is configured by Python3.6, Pytorch1.1.0 and MMDetection [8] framework. Besides, our tracking algorithm is tested on the VisDrone2019 [21] MOT dataset.

4.1 Implementation details

Data preprocessing Our whole experiments are based on the MMDetection [8] framework, which requires the training dataset to meet the requirements of COCO format. Firstly, we need to convert the Visdrone2019 MOT dataset to the COCO format. Besides, Cascade mask RCNN will be used in our prediction network; therefore, it is also necessary to extract the mask of the target in the video dataset. An example of the masked picture is shown in Fig. 10.

Training process In our experiment, two methods are trained for training, one is pre-training weights based on the COCO dataset, and the other is random initialization trained from the beginning to the end. However, the latter is challenging to converge during the experiment; therefore the former training method is chosen. Besides, based on the HDHNet backbone network proposed, experiments are conducted with different prediction networks and different



a. Original picture b. Masked picture

Fig. 10 Example of extracting mask: (**a**) Denote the original picture; (**b**) Show the corresponding masked picture

resolutions to train 50 epochs on the VisDrone2019 MOT dataset. The loss function denoted in Sec. 3.3 is used, and the SGD is chosen for the optimizer. It is found that when training to 30 epochs, the convergence is close to 0.2, and the detection accuracy reaches more than 98%.

4.2 Performance evaluation

4.2.1 Evaluation metrics

Two authoritative MOT metrics are used to evaluate our MOT system performance, which are defined as [48] and CLEAR MOT metrics [1]. These metrics are designed to assess the overall performance, and indicate the potential shortcomings in each model. These metrics are denoted as follows:

1) FP (↓): false positives of the entire video;
2) FN (↓): false negatives of the entire video;
3) IDSW (↓): ID switches of the entire video.
4) Frag (↓): fragmentations where a track is interrupted by miss detections;
5) FM (↓): number of a ground-truth trajectory interrupted during the tracking process
6) IDF1(↑): Ratio of correctly identified detection to the number of computed detections and ground truth
7) MOTA (↑): combining false positives, false negatives, and IDSW, the score is then defined as follow:

$$MOTA = 1 - \frac{(FN + FP + IDSW)}{GT} \in (-\infty, 1] \tag{9}$$

8) MOTP (↑): the mismatch between the ground truth and the predicted results is calculated as follows:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_{t} c_t} \tag{10}$$

4.2.2 Results

Table 1 shows a comparison between the proposed algorithm and other state-of-the-art methods on the VisDrone2019 benchmark validation set, and the benchmark is followed to evaluate our work with the CLEAR MOT Metrics [1]. Among which IDF1 and MOTA are considered to be most important. To build the target detectors, HDHNet is connected with different prediction networks, including CNN, Cascade RCNN, Cascade mask RCNN, and HTC, respectively. As the performance of trackers depends greatly on the performance of detectors, it is observed that the proposed HDHNet has obviously improved our tracker performance. Besides, MOTA, MOTP, IDS, and IDF1 scores are in the leading position in the VisDrone2019 benchmark among the other online two-stages algorithms; especially, MOTA has achieved the improvement of over 4%. Based on our HDHNet, it is found that FP and IDS scores of Cascade mask RCNN are much better, while HTC performs better as the remaining indicators. GOG [34] benefits from global information of whole sequences and spatial overlap between frame detections, thus achieving the best FN scores in terms of our MOT metrics. In the evaluation of MOT trackers, the MOTA is closely related to the detector's accuracy and recall, while IDF1 can express identity consistency. A powerful multi-target tracking model should be characterized by higher IDF1 and MOTA scores. In this paper, the experiment results show that our detector which combines HDHNet and HTC outperforms other methods a lot, although some indicators are not the best.

In Table 2, different detectors in the VisDrone2019 benchmark are compared with varying resolutions of input, and the recognized targets are usually very small in the UAV scene. Traditional detectors are generally used to detect relatively large targets, which are not ideal for small-target recognition. Both RetinaNet and HRNet are detectors used for multi-scale targets, and in this paper, it is observed that RetinaNet is available for the maximum tracking speed of 23 FPS with 720P resolution, which can be deemed as real-time tracking. However, the recall and precision are much lower than those of HRNet and HDHNet. The GFLOPs is calculated on the input size 1920*1080. Compared with HRNet, our model has reduced the number of parameters by 15 M and the FLOPs has dropped by more than 200. As a reliably excellent feature extractor for small-target recognition using high-resolution features, HRNet has improved the recall by about 10% and the accuracy by over 20% respectively compared with RetinaNet. However, its tracking speed is really slow, which has only 2.9 FPS for 720P resolution. Our proposed HDHNet is available to improve FPS by about 1.5 PFS with the close recall and precision of HRNet, whereas it still fails to achieve a real-time tracking with high precision.

Table 1 Comparisons results of the algorithms on the VisDrone MOT dataset using the CLEAR-MOT evaluation protocol

Method	MOTA↑	MOTP↑	IDF1↑	FN↓	FP↓	IDS↓	FM↓
GOG [34]	28.7	76.1	36.4	17,706	144,657	1387	2237
IOUT [4]	28.1	74.7	38.9	36,158	126,549	2393	3829
SORT [3]	18.1	65.1	32.2	78,467	104,453	3342	4304
HDHNet (CNN)	27.6	74.3	35.2	48,908	153,487	2489	3928
HDHNet (Cascade RCNN)	31.8	75.4	41.5	36,788	93,909	1134	1387
HDHNet (Cascade mask RCNN)	32.5	75.2	40.9	39,743	79,788	1042	1425
HDHNet (HTC)	32.9	76.9	42.3	35,686	80,454	1056	1242

Table 2 Comparison results of different detectors based on improved DeepSORT algorithm in terms of four different resolutions

Detectors	Resolution	Rcll (%)	Prcn (%)	FPS	#Params	GFLOPs
RetinaNet [27]	1280*720	52.6	50.2	23	25.5 M	170.2
	1920*1080	54.2	56.4	18		
	2880*1620	59.5	58.7	10		
	3840*2160	55.3	56.3	7		
HRNet [41]	1280*720	62.4	76.3	2.9	59.2 M	653.2
	1920*1080	67.8	79.5	2.2		
	2880*1620	72.5	80.5	1.3		
	3840*2160	69.5	76.2	0.9		
HDHNet (ours)	1280*720	62.5	75.4	**4.3**	38.4 M	397.2
	1920*1080	70.2	78.2	3.7		
	2880*1620	**73.4**	**82.5**	2.1		
	3840*2160	71.3	79.3	1.9		

Figure 11 shows the results when applying GIoU and NMS to our tracking framework, whose metrics of MOTA and precision have been improved by about 1% and 3%, respectively. When both GIoU and NMS are applied to our tracking model, the tracking performance will be the best. In our experiment, different GIoU threshold L_1 and NMS threshold L_2 are tested, and it is found that it is available for the greatest MOTA and precision when setting parameters as L_1= 0.65 and L_2 = 0.85.

Some qualitative results based on our tracking framework are shown in Fig. 12, which consists of videos from three different scenes, namely, day, night, and dense scenes. Different targets are represented with different colors in these frames, and they are furnished with unique identification and a unique number. As we can see, most of the targets are detected correctly and keep the same identity in different frames.

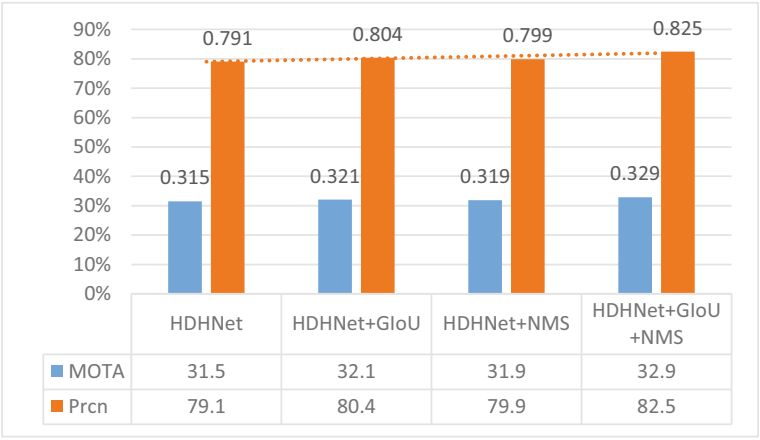


Fig. 11 The results when using the tricks GIoU and NMS to HDHNet tracking framework

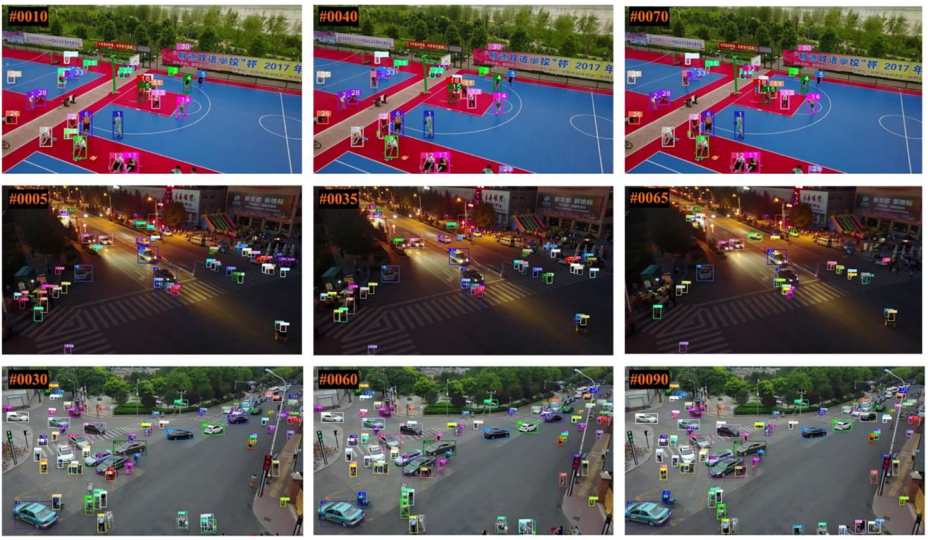


Fig. 12 Some results of our tracking algorithm on the VisDrone 2019 MOT dataset, which are obtained every 30 frames

5 Conclusion

In this paper, an online MOT framework based on the UAV system is proposed, which includes a novel and high-performance target detector named HDHNet. Besides, an improved MOT algorithm base on DeepSORT is proposed. The HDHNet combines the merits of hierarchical aggregation network and those of high-resolution representation network. It is available to extract high resolution and multi-scale features which are then applied to different prediction networks. Moreover, an adjustable loss function is proposed to train our model, which can further solve the problems of class imbalance and hard samples. The proposed model has much lower FLOPs and fewer parameters compared with HRNet to achieve lower computational complexity. The experiments and results show that our method is available for the highest MOTA and precision compared with state-of-the-art methods on the VisDrone2019 MOT benchmark. Though it still fails to be real-time at present, we will strive to achieve that goal in the future. What's more, introducing a video-based attention mechanism or some new efficient embedding CNN module to our MOT framework will be regarded as our future work.

Acknowledgments This work is supported by the Scale Test Verification Assessment and Demonstration Application for SEANET Program of the Chinese Academy of Sciences. Grant No. is XDC02070800.

References

1. Bernardin K, Stiefelhagen R (2008) Evaluating multiple object tracking performance: the CLEAR MOT metrics. *EURASIP Journal on Image and Video Processing*, *2008*, 1–10
2. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH (2016) Fully-convolutional siamese networks for object tracking. In European conference on computer vision (pp. 850–865). Springer, Cham
3. Bewley A, Ge Z, Ott L, Ramos F, Upcroft B (2016) Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3464–3468). IEEE
4. Bochinski E, Eiselein V, Sikora T (2017) High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1–6). IEEE
5. Cai Z, Vasconcelos N (2018) Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6154–6162)
6. Chen B, Wang D, Li P, Wang S, Lu H (2018) Real-time'Actor-Critic'Tracking. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 318–334)
7. Chen K, Pang J, Wang J, Xiong Y, Li X, Sun S ... Loy CC (2019) Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4974–4983)
8. Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, ... Zhang Z (2019) MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155
9. Chen Y, Wang J, Liu S, Chen X, Xiong J, Xie J, Yang K (2019) Multiscale fast correlation filtering tracking algorithm based on a feature fusion model. Concurrency and Computation: Practice and Experience, e5533
10. Chen Y, Wang J, Xia R, Zhang Q, Cao Z, Yang K (2019) The visual object tracking algorithm research based on adaptive combination kernel. J Ambient Intell Humanized Comput 10(12):4855–4867
11. Chen Y, Wang J, Chen X, Sangaiah AK, Yang K, Cao Z (2019) Image super-resolution algorithm based on dual-channel convolutional neural networks. Appl Sci 9(11):2316
12. Chen Y, Tao J, Zhang Q, Yang K, Chen X, Xiong J, ... Xie J (2020) Saliency Detection via the Improved Hierarchical Principal Component Analysis Method. Wireless Communications and Mobile Computing, 2020
13. Chu Q, Ouyang W, Li H, Wang X, Liu B, Yu N (2017) Online multi-object tracking using CNN-based single object tracker with spatial-temporal attention mechanism. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4836–4845)
14. Ciaparrone G, Sánchez FL, Tabik S, Troiano L, Tagliaferri R, Herrera F (2020) Deep learning in video multi-object tracking: A survey. Neurocomputing 381:61–88
15. Deng J, Trigeorgis G, Zhou Y, Zafeiriou S (2019) Joint multi-view face alignment in the wild. IEEE Transactions on Image Processing 28(7):3636–3648
16. Fan, D. P., Wang, W., Cheng, M. M., & Shen, J. (2019). Shifting more attention to video salient object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8554–8564).
17. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580–587)
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778)
19. Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In European conference on computer vision (pp. 702–715). Springer, Berlin, Heidelberg
20. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. IEEE trans pattern analysis machine intell 37(3):583–596
21. Hu P, Wen L, Du D, Bian X, Hu Q, Ling H (2020) Vision Meets Drones: Past, Present and Future. *arXiv preprint arXiv:2001.06303*
22. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700–4708)
23. Li P, Wang D, Wang L, Lu H (2018) Deep visual tracking: Review and experimental comparison. Pattern Recognition 76:323–338

24. Li B, Yan J, Wu W, Zhu Z, Hu X (2018) High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 8971–8980)
25. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J (2019) Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4282–4291)
26. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, ... Zitnick CL (2014) Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740–755). Springer, Cham
27. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980–2988)
28. Luo W, Xing J, Milan A, Zhang X, Liu W, Zhao X, Kim TK (2014) Multiple object tracking: A literature review. arXiv preprint arXiv:1409.7618
29. Marvasti-Zadeh SM, Cheng L, Ghanei-Yakhdan H, Kasaei S (2019) Deep learning for visual tracking: A comprehensive survey. arXiv preprint arXiv:1912.00535
30. Mills-Tettey GA, Stentz A, Dias MB (2007) The dynamic hungarian algorithm for the assignment problem with changing costs
31. Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 3, pp. 850–855). IEEE
32. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In European conference on computer vision (pp. 483–499). Springer, Cham
33. Peng X, Feris RS, Wang X, Metaxas DN (2016) A recurrent encoder-decoder network for sequential face alignment. In European conference on computer vision (pp. 38–56). Springer, Cham
34. Pirsiavash H, Ramanan D, Fowlkes CC (2011) Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011* (pp. 1201–1208). IEEE
35. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91–99)
36. Ren L, Lu J, Wang Z, Tian Q, Zhou J (2018) Collaborative deep reinforcement learning for multi-object tracking. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 586–602)
37. Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 658–666)
38. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234–241). Springer, Cham
39. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
40. Sun D, Yang X, Liu MY, Kautz J (2018) Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8934–8943)
41. Sun K, Zhao Y, Jiang B, Cheng T, Xiao B, Liu D, ... Wang J (2019) High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*
42. Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2016) Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint arXiv:1602.07261
43. Tang S, Andres B, Andriluka M, Schiele B (2016) Multi-person tracking by multicut and deep matching. In *European Conference on Computer Vision* (pp. 100–111). Springer, Cham
44. Voigtlaender P, Krause M, Osep A, Luiten J, Sekar BBG, Geiger A, Leibe B (2019) MOTS: Multi-object tracking and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7942–7951)
45. Wang W, Shen J (2017) Deep visual attention prediction. IEEE Trans Image Process 27(5):2368–2378
46. Wang Z, Zheng L, Liu Y, Wang S (2019) Towards Real-Time Multi-Object Tracking. arXiv preprint arXiv:1909.12605
47. Wojke N, Bewley A, Paulus D (2017) Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)* (pp. 3645–3649). IEEE
48. Wu B, Nevatia R (2006) Tracking of multiple, partially occluded humans based on static body part detection. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 1, pp. 951–958). IEEE
49. Xu Y, Ban Y, Alameda-Pineda X, Horaud R (2019) DeepMOT: A Differentiable Framework for Training Multiple Object Trackers. arXiv preprint arXiv:1906.06618
50. Yang J, Liu Q, Zhang K (2017) Stacked hourglass network for robust facial landmark localisation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 79–87)

51. Yu F, Wang D, Shelhamer E, Darrell T (2018) Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2403–2412)
52. Zhang L, Li Y, Nevatia R (2008) Global data association for multi-object tracking using network flows. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–8). IEEE
53. Zhang P, Zhong Y, Li X (2019) SlimYOLOv3: Narrower, faster and better for real-time UAV applications. In Proceedings of the IEEE International Conference on Computer Vision Workshops (pp. 0–0)
54. Zhang J, Xie Z, Sun J, Zou X, Wang J (2020) A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. IEEE Access 8:29742–29754
55. Zhao JX, Liu JJ, Fan DP, Cao Y, Yang J, Cheng MM (2019) EGNet: Edge guidance network for salient object detection. In Proceedings of the IEEE International Conference on Computer Vision (pp. 8779–8788)
56. Zhou X, Wang D, Krähenbühl P (2019) Objects as points. *arXiv preprint arXiv:1904.07850*
57. Zhu J, Yang H, Liu N, Kim M, Zhang W, Yang MH (2018) Online multi-object tracking with dual matching attention networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 366–382)
58. Zhu Z, Wang Q, Li B, Wu W, Yan J, Hu W (2018) Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 101–117)