remote sensing

MDPI

Article

A Real-Time Tracking Algorithm for Multi-Target UAV Based on Deep Learning

Tao Hong [1,2], Hongming Liang [2,*], Qiye Yang [3], Linquan Fang [1], Michel Kadoch [4] and Mohamed Cheriet [4]

[1] Yunnan Innovation Institute·BUAA, Kunming 650233, China
[2] School of Electronic and Information Engineering, Beihang University, Beijing 100191, China
[3] AVIC Chengdu Aircraft Design and Research Institute, Chengdu 610041, China
[4] École de Technologie Supérieure (ETS), University of Quebec, Montreal, QC H2L 2C4, Canada
* Correspondence: lianghongming@buaa.edu.cn

Abstract: UAV technology is a basic technology aiming to help realize smart living and the construction of smart cities. Its vigorous development in recent years has also increased the presence of unmanned aerial vehicles (UAVs) in people's lives, and it has been increasingly used in logistics, transportation, photography and other fields. However, the rise in the number of drones has also put pressure on city regulation. Using traditional methods to monitor small objects flying slowly at low altitudes would be costly and ineffective. This study proposed a real-time UAV tracking scheme that uses the 5G network to transmit UAV monitoring images to the cloud and adopted a machine learning algorithm to detect and track multiple targets. Aiming at the difficulties in UAV detection and tracking, we optimized the network structure of the target detector yolo4 (You Only Look Once V4) and improved the target tracker DeepSORT, adopting the detection-tracking mode. In order to verify the reliability of the algorithm, we built a data set containing 3200 pictures of four UAVs in different environments, conducted training and testing on the model, and achieved 94.35% tracking accuracy and 69FPS detection speed under the GPU environment. The model was then deployed on ZCU104 to prove the feasibility of the scheme.

Keywords: UAV; 5G; multi-target detection and tracking; YOLOv4; DeepSORT

Citation: Hong, T.; Liang, H.; Yang, Q.; Fang, L.; Kadoch, M.; Cheriet, M. A Real-Time Tracking Algorithm for Multi-Target UAV Based on Deep Learning. *Remote Sens.* **2023**, *15*, 2. https://doi.org/10.3390/rs15010002

Academic Editor: Andrzej Stateczny

Received: 3 November 2022
Revised: 15 December 2022
Accepted: 16 December 2022
Published: 20 December 2022

1. Introduction

The unmanned aerial vehicle (UAV) sector has advanced significantly as industry and technology have progressed. Aerial photography, agriculture, surveying and mapping, traffic supervision and other civilian areas are now using UAV technology that was previously only employed for military and scientific study. Aerial video, or all-around shooting of a target from a high altitude for film and television production and news reporting, achieves high-quality footage at a low cost [1]. Drones can be used in agriculture to establish the planting area, planting plan and risk assessment, as well as daily testing of agricultural crop growth and estimating the degree of damage under periods of disease and pest infestation, allowing for efficient and modern agricultural supervision. Aerial surveying and mapping using UAVs complements traditional aerial photogrammetry technology, with high precision, cheap operational costs, a quick production cycle and excellent data analysis capabilities. UAVs have potential in surveying and mapping work for national projects, as well as emergency response and other areas. In today's urban intelligent traffic network management, UAVs may perform live monitoring and traffic flow regulation in real time, reducing traffic congestion [2].

In recent years, the increasing number of drones has put pressure on the regulation of the low-altitude airspace. Such low-altitude, low-speed and small aircraft are difficult to detect in real time, bringing significant security threats to all countries. In the civil field [3], UAV illegal flight disturbance near the airport has caused the delay and cancellation of civil aviation flights and threatened flight safety in civil aviation. In April 2017, nine

incidents of UAV disturbance occurred in Chengdu, causing more than 100 flights to divert or turn back. UAV flight operations that are permitted without holder requirements, which most of the time is due to user error or product design flaws, represent a significant threat, and led to a series of drone attacks in May 2018 [4]. The son of one of the German national staff in a Beijing no-fly zone operating UAVs had his face cut by a hit-and-run UAV. In military airspaces, because most UAV flights are slow and the flight airspace is usually below 600 m, air defense warning radar and other air defense weapons often confuse these vehicles with insects and birds in flight, causing false positives. Additionally, UAV target drones' visual features are not very clear and so it is difficult to detect drones based on images and sound from ground surveillance, threatening military security. In 2019, Russian air defense systems in Syria detected small aerial targets suspected of terrorist attacks in Syria [5]. Therefore, for the safety of civil and military fields, accurate detection, tracking and interception of UAVs in designated areas is of practical significance.

The earlier proposed scheme for the detection of UAV targets is based on the recognition of audio signals generated by UAV flights and real-time tracking and monitoring of UAV remote control information and communication signals based on radio frequency scanning technology [6]. The detection accuracy of these two methods is low and so only targets within a short distance can be identified, which does not meet the requirements for UAV detection.

In recent years, the methods widely used in UAV target detection have included radar detection and image processing. For unmanned aerial vehicles such as "low slow small" targets with weak electromagnetic reflection, the detection ability of traditional radar detection systems is low in a complex, cluttered environment. Improving detection performance would be costly [7]. Image processing technology has been applied in the field of UAV detection for a long time. On the established UAV optical dataset, image processing algorithms such as SIFT and HOF operators are used to extract UAV target information. Finally, SVM, Adaboost and other classifiers are used to achieve target classification and recognition. Although such traditional methods are easy to understand and operate, they only make use of low-level information in images. Their detection accuracy and real-time performance in identifying UAV targets are low and their robustness inadequate; furthermore, they struggle to recognize multiple UAV targets in a complex background.

We propose a UAV detection scheme based on deep learning. The deep convolutional network is used as the target detector to extract high-level information from UAV images through supervised learning, so as to achieve high-precision and real-time detection of UAV targets, and then realize UAV detection. However, most depth learning UAV detection schemes remain in the detection stage and do not involve UAV target tracking. The development of the 5G network has made low-cost detection and tracking drones possible in the urban context and its characteristics of low latency and fast transmission speed are promising for real-time monitoring [8]. In addition, in recent years, China has vigorously advocated for the construction of smart cities and the urban supervision system has become increasingly intelligent, making it easier to obtain UAV monitoring images. Therefore, this study considers a combination of deep learning methods to put forward effective solutions to UAV monitoring problems.

2. Model and Methods

2.1. System Model

With the development and application of deep learning, new avenues in UAV detection are opening up. The deep convolutional network is used as the target detector to extract high-level information of UAV images through supervised learning [9], so as to achieve high-precision and real-time UAV target detection, and then realize UAV detection. Some scholars use transfer learning to classify UAV targets, or combine UAV liDAR images and use the convolutional network to extract features to detect UAVs. Currently, most deep learning UAV monitoring schemes involve only a single target detection or target tracking. This project believes that the "detection-tracking" mode formed by the combination of

target detection means and target tracking means may be a more perfect method to solve the UAV monitoring problem, which not only solves the problem of the limitations of target detection application, but also makes the tracking algorithm more accurate. The flow chart of UAV tracking is shown in Figure 1.

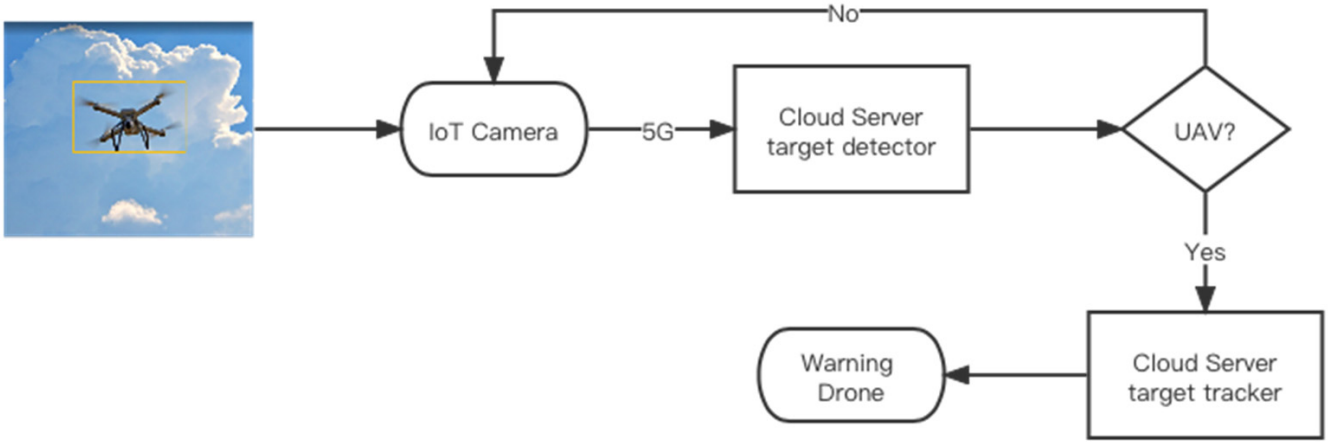

Figure 1. The UAV tracking flow chart.

After the video of the drone is captured by the camera, the target detector is used to determine whether it is the drone target. Then the target tracker is used to predict the trajectory of the drone target, track and warn it, and complete the monitoring of the drone in a specific area [10]. This scheme greatly improves the monitoring accuracy and avoids the waste of unnecessary computing power in tracking other targets.

2.1.1. Target Detection Algorithms

Thanks to breakthroughs in computer computing, deep learning has ushered in new developments [11], including convolutional neural networks, which have made great progress in the field of image detection in recent years. Compared with traditional image-processing algorithms, convolutional neural networks make use of advanced features, have higher recognition accuracy and better robustness.

The object detector usually consists of two parts: the head, which is used to predict object classes and bounding boxes, and the backbone, which is pre-trained on ImageNet. Target detection tasks can be divided into target classification and target location. Target detection methods can be divided into two-stage algorithm and one-stage algorithm.

The two-stage algorithm divides the target detection task into two steps. The single-stage algorithm simultaneously classifies and regresses candidate anchor frame targets to complete the detection task [12]. Table 1 shows the main convolutional network models sorted by category.

Table 1. Object detection algorithm classification.

The Network Structure	Classification of Network	Light Weight Network
backbone network	VGG, ResNet, ResNeXt, DenseNet	SqueezeNet, MobileNet, ShuffleNet
neck network	additional layer SPP, ASPP, RFB, SAM	Characteristics of the fusion FPN, BiFPN, NAS-FPN, ASFF
one-stage algorithm	RPN, SSD, YOLO, RetinaNet	CornetNet, CenterNet, CentripetalNet
two-stage algorithm	Mask R-CNN, Fast R-CNN, Faster R-CNN	Reppoints

The target detection task of the two-stage algorithm is divided into two parts: extracting the region of interest (RoI), and then classifying and regressing the RoI. At present, existing networks include R-CNN, SPP-Net, Fast-RCNN [13], Faster-RCNN, Mask-RCNN [14], Cascade R-CNN and so on. For example, Faster-RCNN, for the two-stage method, extracts the target region from the RPN (region proposal network) regional candidate network and then inputs the RCNN part of the convolutional neural network for target classification and border regression. For Fast-RCNN, although the RPN [15] improves the accuracy of the algorithm, it is also the biggest factor affecting real-time performance. In contrast, the traditional region initialization algorithm Selective-search is simple in principle and fast at generating candidate boxes, but it contains a significant number of redundant frames, which increases the amount of network computation, and the accuracy is thus inferior to that of the RPN [16].

The single-stage algorithm directly classifies and regresses the candidate anchor frames without preselecting regions. As the name suggests, it uses another strategy: applying a single neural network to the image. Typical models include the YOLO series, RefineNet, SSD, etc. The YOLO algorithm combines the target classification task with the border regression task and adopts the same loss function for training. Compared with the two-stage method, YOLO has a huge advantage in computing speed. However, because the candidate boxes of YOLO only have a fixed number and position, its target box regression performance is weak. Therefore, the main goal of the subsequent versions of YOLO (YOLOv2, YOLOv3, YOLOv4) became balancing the detection rate and accuracy of the model [17].

The above-mentioned methods can be classified as algorithms based on anchors, through which candidate boxes of different sizes and proportions are generated to solve the multi-scale detection problem. The anchor mechanism separates the classification and regression tasks of target detection. Firstly, the extraction network of the preselected frame is trained through the preset anchor frame to realize the binary classification of the target and background, and then the classification and regression tasks are carried out on the basis of the preset anchor frame, which significantly improves the target detection accuracy. However, due to the large number of anchors, the computation is aggravated, and many anchor-related parameters need to be set, so the detection speed of this kind of network is slow and the training process is difficult [18].

In general, a detector without an anchor is a one-stage algorithm. The Corner-Net network is a typical example. Corner-net uses the idea of keypoint detection to solve the problem of target detection. The network detects the keypoint information of the target in the figure, namely the upper-left corner and lower-right corner of the target box. By turning object detection into paired keypoint detection, the network eliminates the need to design a region extraction network for generating anchor boxes. However, because Corner-Net has a weak reference to the global information, the detection accuracy is not very high. The subsequent upgraded network, Center-Net, learned from this, representing each object with triplets [19].

In the object detection task, we can obtain the following four state quantities to evaluate the model performance, and their state changes are shown in Figure 2:

- True positives (YP): correctly predicted by the model as positive samples;
- True negatives (TN): correctly predicted as negative samples by the model;
- False positives (FP): negative samples are wrongly predicted as positive samples by the model;
- False negatives (FN): a positive sample is incorrectly predicted as negative by the model.

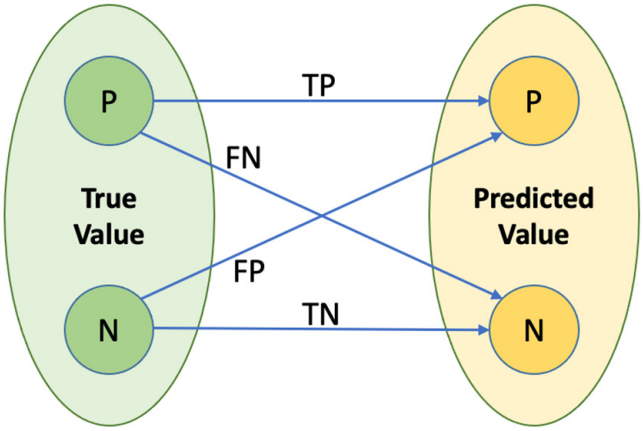

Figure 2. Change in state quantity.

In the field of target detection, it is usually necessary to box out the detected objects, which requires the evaluation of the quality of the detection boxes, as measured by intersection over union. The following equation is expressed in terms of observed state quantities:

$$IoU = \frac{TP}{TP+FP+FN} \tag{1}$$

For object detection tasks, the main indicators involved are precision, recall, average precision (AP) and mean average precision (mAP):

$$\text{precision} = \frac{TP}{TP+FP} \tag{2}$$

$$\text{recall} = \frac{TP}{TP+FN} \tag{3}$$

In the process of observation, multiple groups of accuracy and recall can be obtained for different confidence thresholds. A P-R (precision–recall) curve can be obtained by taking recall as the abscissa and accuracy as the ordinate, and then integrating the curve to obtain the area, which is the average accuracy AP of a single category. The mAP of the target detector can be obtained by averaging the APs of multiple categories.

2.1.2. Target Tracking Algorithms:

Traditional tracking models use points of interest in time and space for tracking, but they rely too much on low-level features such as intensity spikes and corners. Although traditional algorithms can also achieve high-precision single-target tracking, they struggle in real-time multi-target tracking. Due to the rapid development of flying-target detection technology in recent years, the first test of tracking performance in multi-target detection research was carried out, and owing to its success it became the leading detection technology. The idea of using an existing target detector and matching optimization algorithm to realize multiple-target tracking is formed on the basic principle of utilizing the target detector to realize classification and orientation. Then, the target frame of the previous frame is matched by the matching algorithm, and thus the tracking is realized [20].

The target tracking performance is mainly measured by the stability and accuracy of target tracking and the main indicators are as follows:

1. ID Switch (*IDSW):* indicates the number of times that the tracking ID of the same target changes in a tracking task;
2. Tracing fragmentation: the number of times the status of the same tracing target changes from tracing to fragmentation to tracing in a tracing task;

Multiple-object tracking accuracy (*MOTA*):

$$MOTA = 1 - \frac{\sum_t (FN+FP+IDSW)}{\sum_1 GT} \tag{4}$$

Here, *GT* refers to the total number of truth boxes in a frame.

2.2. Method

The core purpose of our study was to achieve an effective and low-cost UAV tracking method. Among multiple detection and tracking models, YOLOv4 and DeepSORT best suited the accuracy and speed requirements of the multi-target detection and tracking network. The network structure of YOLOv4 is relatively simple and can be conveniently applied to industrial landing. DeepSORT can obtain better experimental results at a lower cost. In order to solve the problems of large target scale change, mutual occlusion and fast speed in UAV detection and tracking, we modified the network structure and loss function of YOLOv4 to some extent, optimized the DeepSORT network matching strategy and improved the data set training strategy. Firstly, multi-target real-time tracking was realized on the GPU platform.

2.2.1. Target Detector

YOLOv4 is the latest official sequel of the YOLO series [21]. On the basis of YOLOv3, the author has adopted various optimization strategies in the field of object detection in recent years to improve the backbone network, activation function, network model training, data augmentation and loss function to different degrees. Although it does not contain too many theoretical innovations, the model ingeniously combines all kinds of target detection tricks to achieve new heights of detection speed and accuracy. Compared with previous generations of models, the improvement of YOLOv4 mainly includes the following parts:

The backbone network part in YOLOv3, the Leaky ReLU activation function, was replaced by Mish, whose expression is:

$$y = x \times \tanh(\ln(1 + ex)) \tag{5}$$

Mish considers the regularization of nonmonotone neural activation function. First of all, Mish is forward-unbounded and therefore can avoid vanishing gradients, and Mish functions are smooth everywhere, which helps the network to obtain better information in training, thus improving accuracy and generalization ability.

Cross-stage partial (CSP) was used to replace the residual module, which mainly aims to reduce the computational burden when enhancing gradient training. These optimizations resulted in a better backbone network, CSPDarknet53.

The neck part of the object detection algorithm is mainly used to integrate feature maps of different scales, so that the network can not only learn the deep classification features, but also attain accurate detection and positioning. This part of YOLOv4 selects the combination of SPP + PAN [22]. The SPP network can accelerate the reasoning calculation of the model, which is beneficial in solving the problem of large size differences of the target. PANet is a target segmentation model based on Mask-RCNN that was developed by Tencent Youtu's team. YOLOv4 refers to the network's neck structure. In YOLOv4, the output of the last three layers of the backbone network is mainly fused with its characteristics. Compared with FPN, PAN adds a step called bottom-up fusion, and in the PAN of YOLOv4 the addition operation combined with a feature map is changed to a multiplication operation. FPN's structure pays more attention to the top-down strong classification features, while the PAN structure in YOLO pays more attention to the bottom-up strong localization features. Such strong combination further improves the network capability of YOLOv4 [23]. The network structure of FPN and PAN is shown in Figure 3.

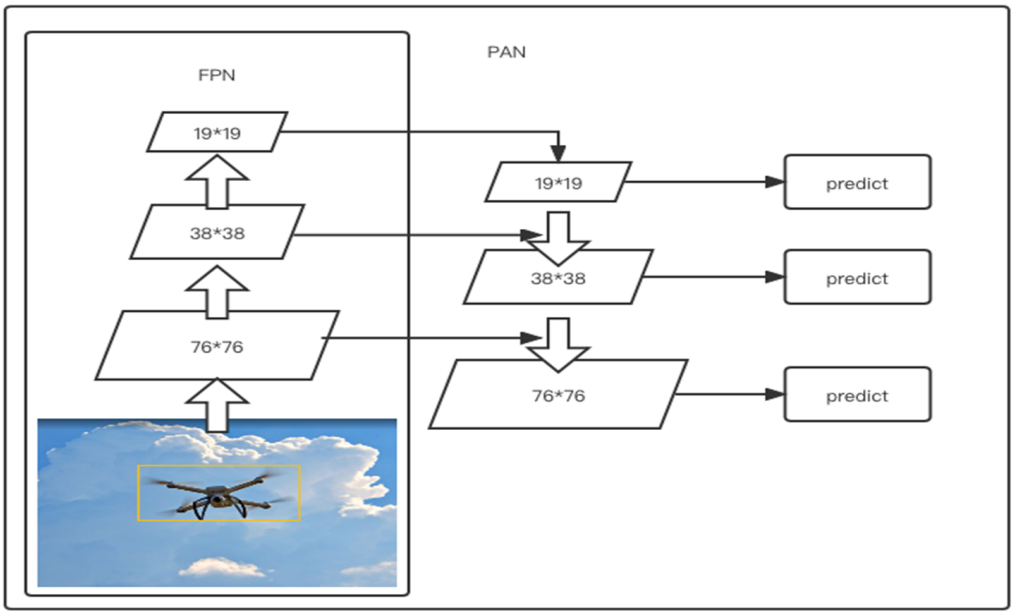

Figure 3. Structure of FPN and PAN.

Data processing: Compared with previous data enhancement operations, such as cutting, rotation and mirroring, the authors implemented efficient data enhancement strategies in YOLOv4. For example, image mixing (Mixup), in which two images are proportionally mixed, that is, image fusion, is conducive to the model's learning of deep features. CutMix cuts objects in an image and then combines them with other images. CutMix operators artificially add blocks to increase the training scene and stimulate the model to learn local features. Mosaic enhancement (Mosaic), which combines four training shapes of a certain proportion into one image, enables the model to recognize objects smaller than the normal size, and blur adds a blur effect to an image. The data enhancement operation enables YOLOv4 to increase its detection capability by 1.6% with almost no reduction in network detection speed.

The loss function, or *CIOU* loss, was introduced in YOLOv4 to replace MSE. The formula of *CIOU* is as follows:

$$L_{CIOU} = 1 - IoU(A, B) + \frac{\rho^2(A_{ctr}, B_{ctr})}{c^2} + a \cdot v \tag{6}$$

$$v = \frac{4}{\pi^2}(\tan^{-1}\frac{w^{gt}}{h^{gt}} - \tan^{-1}\frac{w}{h}) \tag{7}$$

$$a = \frac{v}{(1-IoU)+v} \tag{8}$$

Included intersection over union (*IoU*) is calculated as follows:

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \tag{9}$$

where $\rho^2(A_{ctr}, B_{ctr})$ refers to the Euclidean distance between the center of two rectangular boxes, c^2 is the diagonal length of the smallest rectangle containing A and B, w^{gt} and h^{gt} are the true frame's width and high, w and h are the width and height of the predicted box and ν is the loss function of the punishment. When the real box's and predicted box's width are highly consistent, this does not work.

In the UAV target detection task targeted by this topic, we adjusted the model to a certain extent according to the characteristics of the UAV image. The improved model is shown in Figure 4 below.

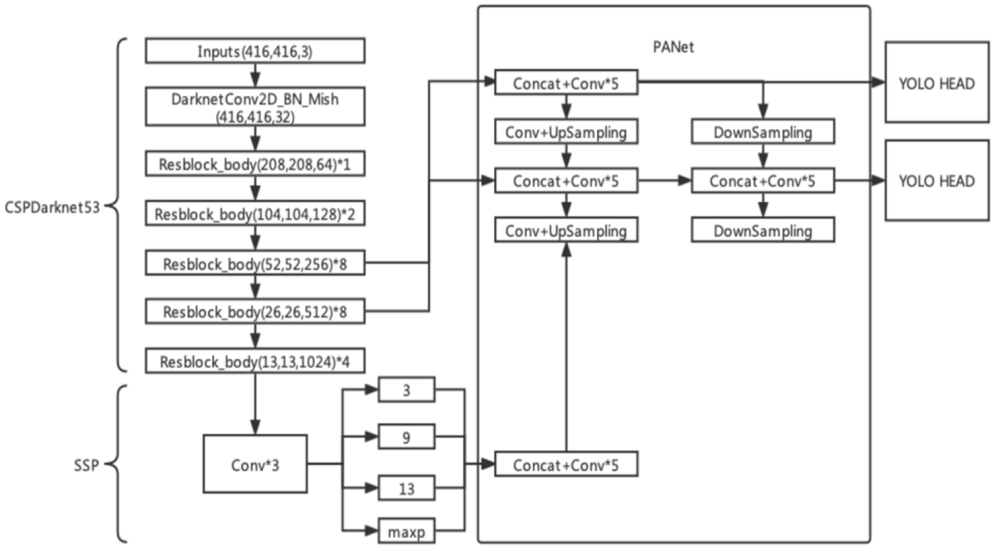

Figure 4. Improved YOLOv4 structure.

The YOLOv4 network modification, as shown in Figure 5, removes the third predicted output of YOLOv4 to satisfy the requirement of the multi-scale target detection task. The three groups were used to detect the branch and three-layer structure of PAN, but no single human–machine objective was used. The output of the last layer of the target detection output field is too large and so the image is not as difficult to process, does little to promote model performance and also increases the operation pressure; therefore, it can be deleted [24].

Figure 5. Random image erasure operation.

The data preprocessing augmentation operation is shown in Figure 5. Sometimes the whole fuselage cannot be displayed because of the occlusion during the flight of UAV. To solve this problem, we carry out random erasure operation on the images in the training set, randomly select a rectangular area in the image area and replace pixels with random values. Such operation can improve the generalization ability of the model and strengthen the learning of local feature information of network targets in the training. At the same time, the robustness of the model to noise and shielding is enhanced.

The loss function modification, the YOLOv4 loss function, consists of three parts: positioning loss, classification loss and confidence loss. Since UAV targets are mainly small- and medium-sized targets, in order to improve the training emphasis of the model on small target objects, the weight coefficient $(2 - w_i \times h_i)$ was added before the positioning loss. The smaller the target, the greater the corresponding positioning loss [14]. The revised positioning loss is as follows:

$$Loss_{coord} = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} 1_{ij}^{obj} (2 - w_i \times h_i)[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \hat{w}_i)^2 + \left(h_i - \hat{h}_i\right)^2] \quad (10)$$

2.2.2. Target Tracker

Simple Online and Real-Time Tracking, published in 2016, proposed adding traditional algorithms, such as the Hungarian algorithm and Kalman filter, on the basis of the original detector [25]. However, it achieved the best target-tracking performance at that time. The SORT algorithm flow is shown in Figure 6 below.

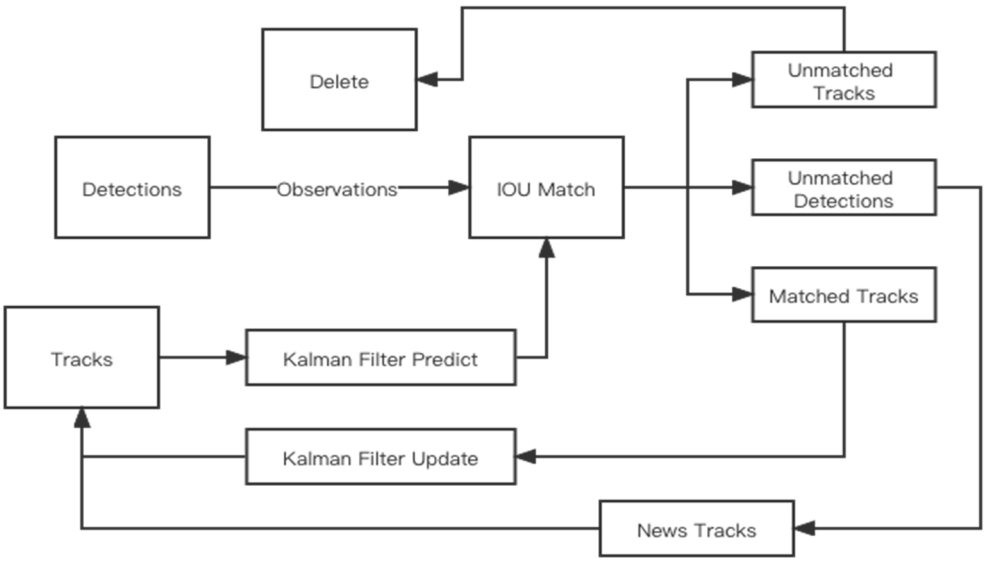

Figure 6. Improved DeepSORT Structure.

Compared with the SORT algorithm, the biggest difference of DeepSORT is that it adds a convolutional network to learn the surface features of the target, which is conducive to improving the stability of the tracking process and greatly reducing the ID switching frequency of the tracking target. The basic structure of DeepSORT is similar to that of SORT, which is divided into two parts: estimation model and data association. DeepSORT can be divided into three steps. In the first step, a Kalman filter is used to predict the current frame trajectory; the second step is to use the Hungarian algorithm to match trajectories and detection boxes, including cascade matching and intersection ratio matching. In the third step, Kalman filter updates the tracking information [26].

Kalman filtering is an algorithm that uses the state equation of a linear system to optimally estimate the state of a linear system through the input and output observations of the system. State estimation is an essential part of the Kalman filter. In general, quantitative inference of a random quantity based on observed data is an estimation problem, especially the state estimation of dynamic behavior, which can realize the estimation and prediction of the real-time operation state. In order to express the relationship between two state quantities, we introduced Mahalanobis distance and cosine distance.

Mahalanobis distance represents the distance between a point and a distribution and the covariance matrix is usually used to measure the similarity of two random variables from the same distribution. The formula is as follows:

$$d^{(1)}(i,j) = (d_j - y_j)^T S_i^{-1} (d_j - y_j) \tag{11}$$

When the moving target is regular, the matching effect of Mahalanobis distance is excellent. For a target with high mobility, such as a UAV, it is difficult to make a match, invalidating the Mahalanobis distance association and resulting in ID switching. In this case, we need to introduce a new metric, cosine distance:

$$d^{(2)}(i,j) = min\left\{1 - r_j^T r_k^{(i)} \middle| r_k^{(i)} \in R_i\right\} \tag{12}$$

When the Kalman filter prediction value is uncertain, DeepSORT will preferentially select IoU matching.

DeepSORT is designed to realize the re-identification function of pedestrians. In UAV monitoring, we summarize the possible problems:

1. The trajectory matches the detection box. For slow-moving objects between the front and back frames, the detector can successfully detect them and then the tracking can be realized;
2. The detection box does not exist, or the detector is missed. There is a trace, but the detection box cannot be matched; the detector performance thus needs to be improved to reduce the rate of missed detection;
3. The trajectory does not match the detection box and the UAV target moves too fast, so it flies out of the field of view, causing matching failure;
4. The two detection boxes overlap and there is occlusion between the targets, but the minimum cosine distance of the special diagnosis map can be calculated by cascading matching in DeepSORT to achieve re-recognition.

We also adapted the DeepSORT model to address the above issues. In order to avoid the failure of cascade matching and IoU matching in DeepSORT, the Hungarian algorithm was supplemented with traditional Euclidean distance to enhance the persistence of UAV target tracking. The formula is as follows:

$$d(i,j) = \rho(x_j - x_i, y_j - y_i) \tag{13}$$

Set the threshold as $d_{max} = 100$, and match when $d(i,j) \leq d_{max}$; otherwise, delete the predicted value, treat the observation box as a new target and create a new trajectory. The structure of the improved DeepSORT algorithm is shown in Figure 7.

2.3. Dataset Creation

Since there is no publicly available UAV dataset at present, we wrote a crawler program to download 3200 UAV images with rich backgrounds and in different categories on the Internet and annotated them, exporting the annotated data in the VOC2007 format [27] and obtained JPG image data and XML annotated data. The final UAV monitoring dataset is shown in Table 2.

Table 2. UAV detection dataset.

Black Four Rotor	White Four Rotor	Yellow Single Rotor	Red Single Rotor	Total
873	828	650	849	3200

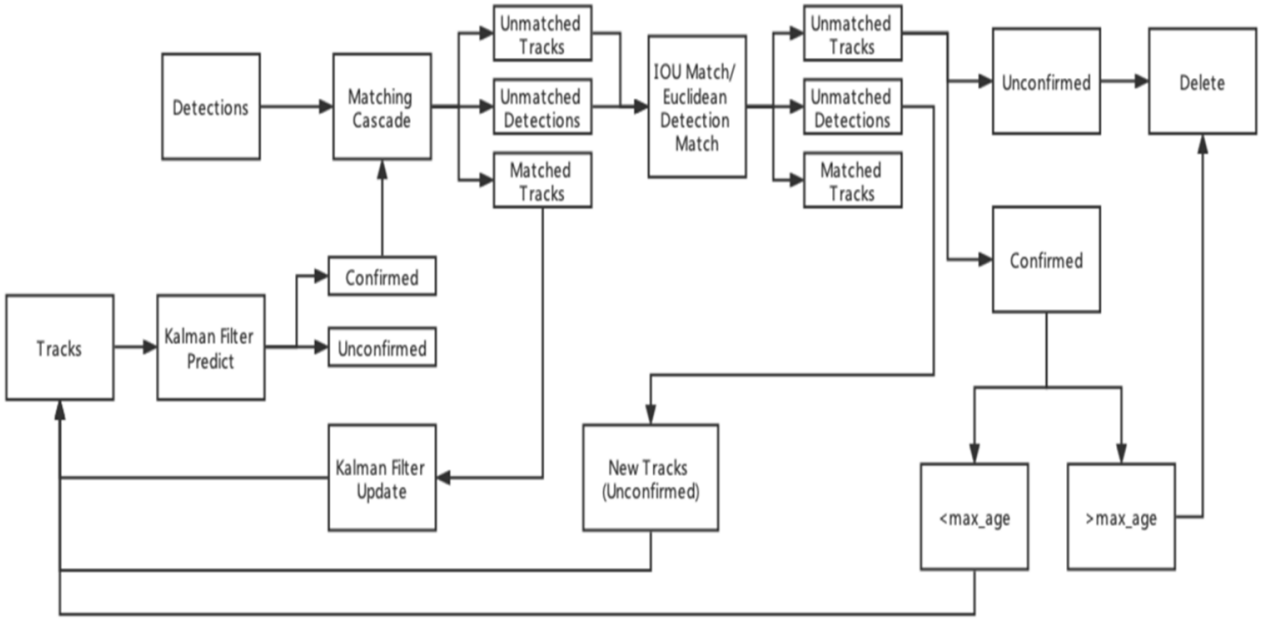

Figure 7. Improved DeepSORT structure.

The obtained UAV dataset cannot be directly used for the training of the YOLOv4 model, because the reading interface of the YOLOv4 model for annotation information is a TXT file, in which each behavior contains annotation box information, including category ID, the normalized value of center point coordinates of an annotation box, the normalized value of the annotation box width and the normalized value of the annotation box height [28]. Therefore, XML information needs to be transformed and the conversion formula is as follows:

$$x' = \frac{(x_{min}+x_{max})}{2W} \tag{14}$$

$$y' = \frac{(y_{min}+y_{max})}{2H} \tag{15}$$

$$w' = \frac{w}{W} \tag{16}$$

$$h' = \frac{h}{H} \tag{17}$$

where W and H are the width and height of the picture, respectively, and w and h are the width and height of the target box, respectively.

Finally, the number of converted TXT texts and pictures reached 3200, and the training set and test set were divided according to the ratio of 9:1 to obtain the training set and test set.

DeepSORT's convolutional network is originally designed for pedestrian re-recognition, so it contains the surface feature information of pedestrians. It is thus necessary to make a relevant UAV re-recognition dataset for re-training. Based on the converted target detection dataset, a program was written to individually cut and save the target box in the image, imitating the Market-1501 dataset, as shown in Figure 8.

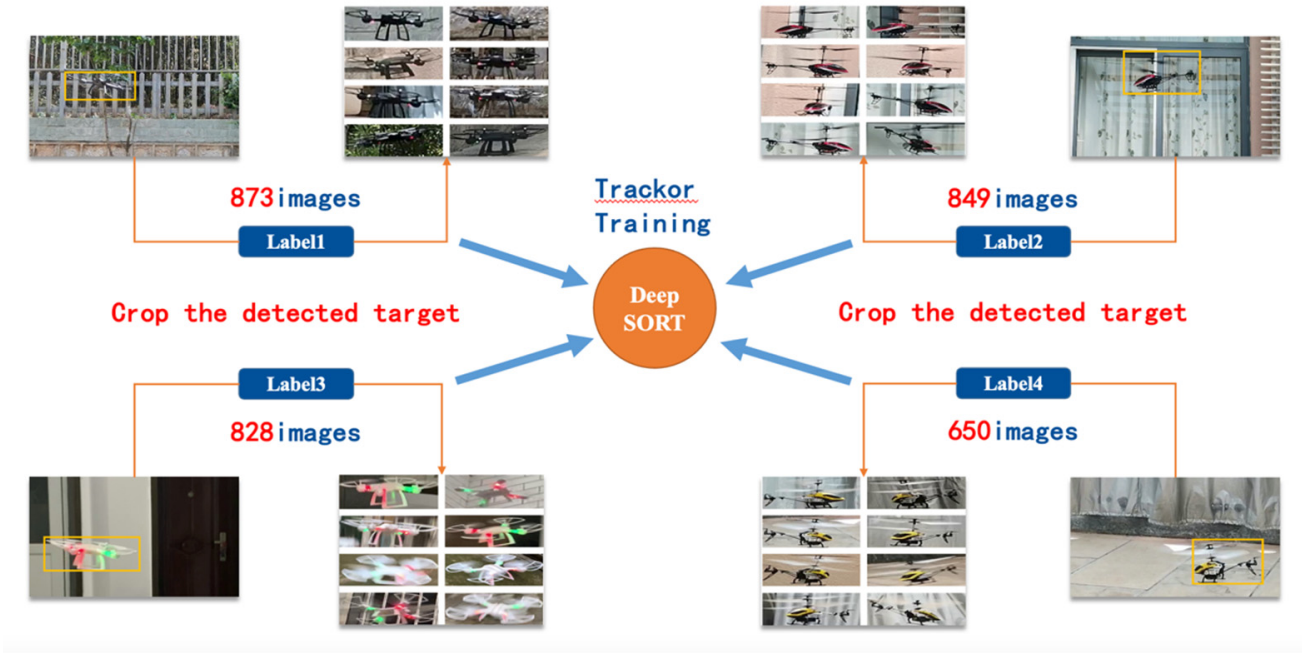

Figure 8. Construction method of UAV tracking dataset.

The cropped image is named according to the category plus ID and the target of the same category is placed in the same folder to obtain the UAV tracking dataset.

3. Experiments

3.1. Training and Analysis of Model

The environment required by target detector training was set up on the Ubuntu18.04 server platform. The server was equipped with a 3090 graphics card and the Pytorch1.8 framework was used for model training. The training parameters were set as follows: there were 3200 images in the UAV target detection dataset, 2980 of which were classified as the training set and 320 as the verification test set. The size of the input image for network training was 720 × 720 and the size of the test image was also 720 × 720 for control variable verification. The batch size of each training round for the UAV detection model was 16 [29], and a total of 300 iterations of training were carried out. The training/validation loss function curves are shown in Figure 9 below. (The horizontal axis represents the number of training sessions and the vertical axis represents the losses).

The 300 iterations of training for the UAV detection model took 10.5 h in total. Comparing the training loss function curve with the verification loss function curve, the border loss, classification loss and confidence loss of the two curves decreased with the increase in the number of iterations, indicating that the training strategy of the model was correct and no fitting phenomenon occurred. When the number of iterations reached 270, both curves tended to be stable and finally reached the minimum value at 300 iterations. The UAV target detection model was successfully trained.

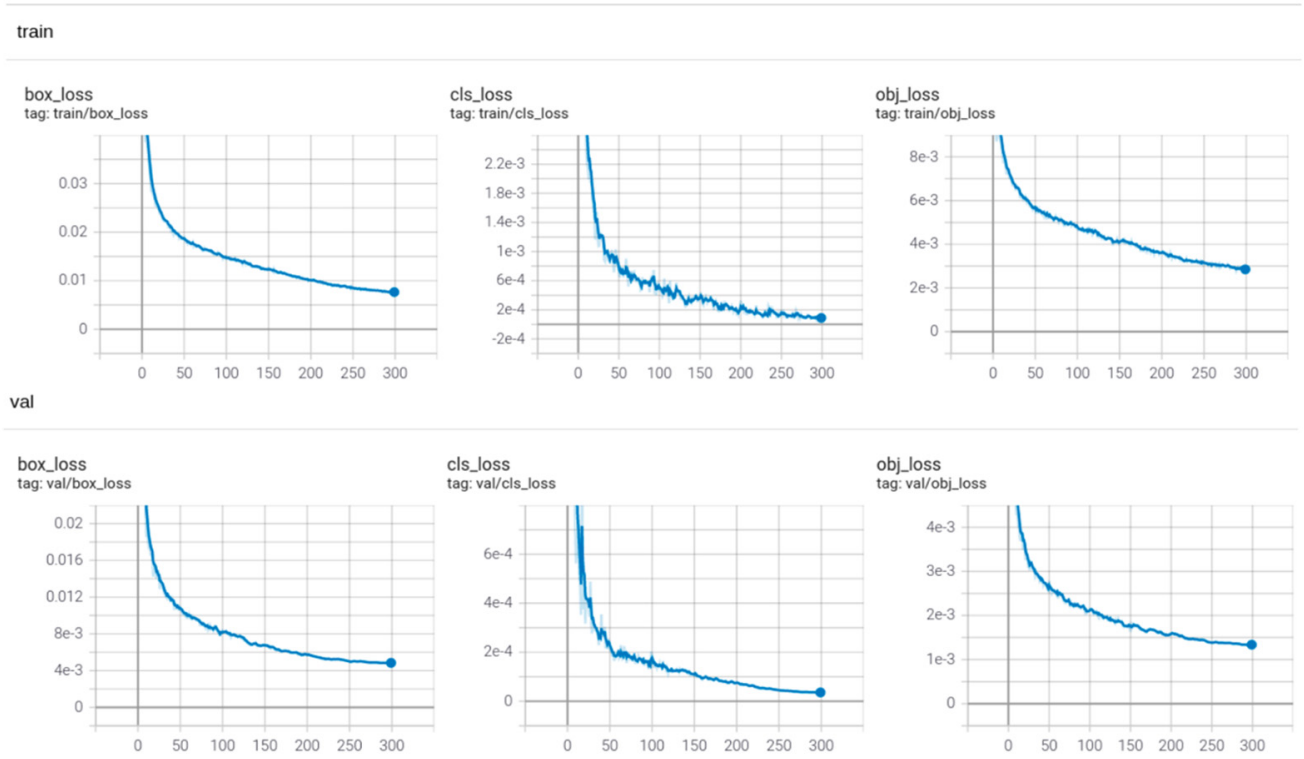

Figure 9. YOLOv4 training loss function curves.

According to the same strategy as that described above, the UAV detection model was trained and the results are shown in Table 3:

Table 3. Model checking performance comparison.

Model	YOLOv4	YOLOv4 Double Branch Detection	YOLOv4 Data Augmented	UAV Target Detector
Training time	10.5 h	10.4 h	10.6 h	10.5 h
mPA (IoU = 0.5)	0.968	0.959	0.990	0.988
Speed	64FPS	69FPS	64FPS	70FPS

From the above table, it can be seen that, for UAV targets, reducing the target detection branch helps to improve the speed of model inference, which increases the speed by 5FPS, and the performance loss caused by it is very small, less than one percentage point. The new data augmentation strategy can effectively improve the mAP of UAV target detection without affecting the reasoning speed. Our modification strategy allows the detection accuracy and detection speed of the model reach new heights, indicating that the modification direction is correct. The detection effect in GPU environment is shown in Figure 10.

We then input the UAV tracking dataset obtained through previous processing into the UAV target tracking model for training. The original network training input was 64 × 128, because most pedestrian images are 1:2 in size, while the image size in the previously generated UAV dataset was closer to 1:1, so the network training parameter was set as 128 × 128. The training batch size was set to 12 and 80 iterations to obtain the training curve shown in Figure 11. (The horizontal axis represents the number of training sessions and the vertical axis represents the losses.)

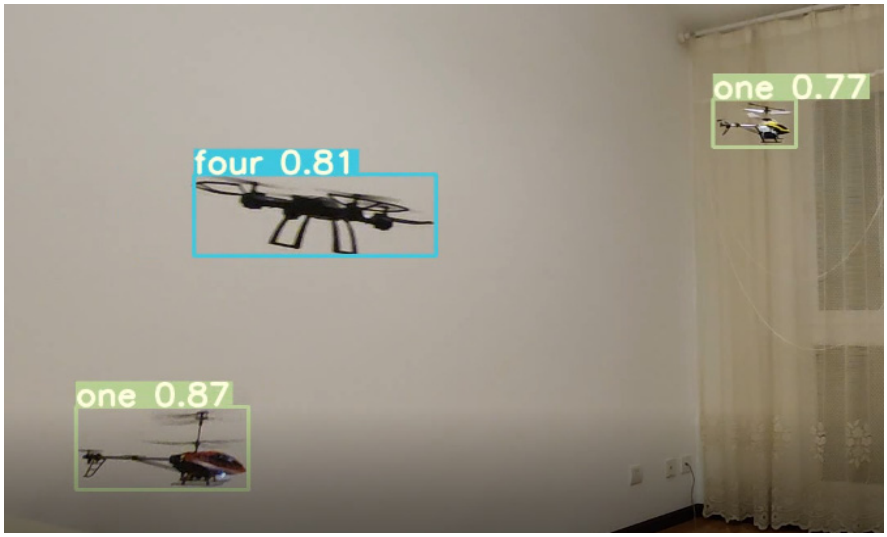

Figure 10. Detection effect in GPU environment.

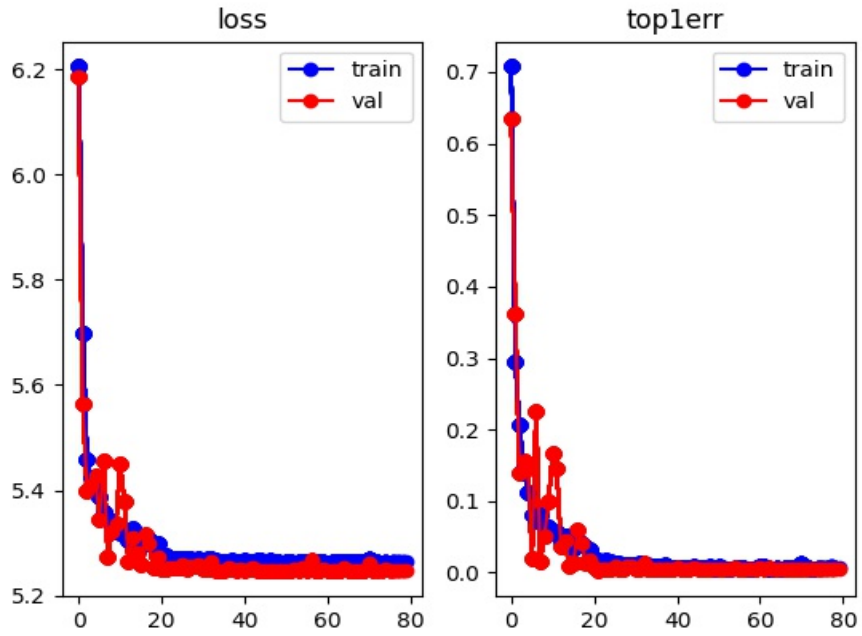

Figure 11. DeepSORT training loss function and TOP1 error rate.

As can be seen from the above figure, in the training process, the training loss was generally consistent with the verification loss, indicating that the data distribution was reasonable, the network did not overfit, the loss function had a correct downward trend and the curve region was smooth at 20 iterations. The TOP1 error rate decreased gradually in the training iteration, and was almost zero at approximately 20 iterations. In conclusion, the target tracking model was successfully trained.

3.2. DPU Deployment of Target Detector

In order to verify the effectiveness of the improved algorithm, we deployed the trained target detection model on the DPU for verification. The DPU selected in this subject is the ZCU104 developed by Xilinx Company for AI algorithm deployment. The model is quantified, pruned, compiled and deployed with the help of Vitis-AI development tool, and the real-time detection system of UAV target is finally built. Figure 12 shows the algorithm deployment process.

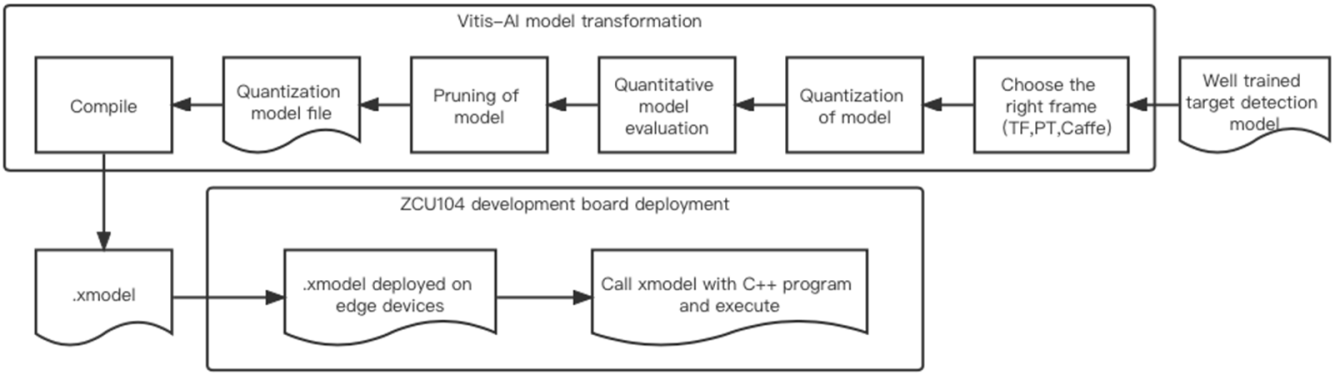

Figure 12. Flowchart of algorithm DPU deployment.

We connected the ZCU104, which deployed the target detection model, to the camera, and carried out the UAV detection in the outdoor complex environment, evaluated the target detection accuracy and speed of the development board and obtained that the detection accuracy of the algorithm reached 87.0% in the development board environment, and could run at 38FPS. Considering the calculation power of ZCU development board, the real-time detection speed can still be barely achieved under the premise of maintaining high detection accuracy, which proves the feasibility of the algorithm in this subject. The DPU detection effect of the algorithm is shown in Figure 13.

Figure 13. DPU detection effect diagram..

3.3. Experimental Results

The trained target detection model and target re-recognition model were combined for UAV target tracking. The setup of this experiment, which did not use a network training video to test the tracking performance of the algorithm, is shown in Figure 14. For the video studied, using the three target drones, complex phenomena occurred: the unmanned aerial vehicle (UAV) flight pattern disappeared, the goals overlapped, and the target flew out of sight, several significant problems already seen in UAV tracking.

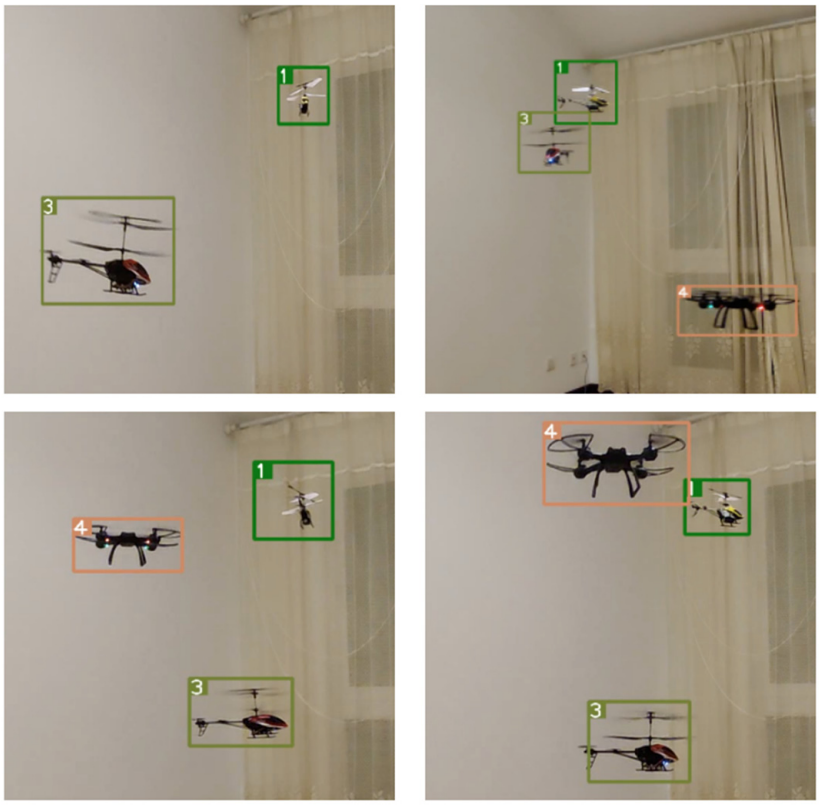

Figure 14. Setup of UAV tracking dataset.

In the previous chapters, Euclidean distance was added into the target tracking model to assist tracking according to the characteristics of UAV targets. In order to test the improvement effect of the target tracker, we combined the improved target detection model with the target tracking model before and after the improvement to track the UAV in the same video. In addition, two models, YOLOv3 + DeepSORT and CenterNet + DeepSORT, were added as comparison experiments in order to verify the performance superiority of our proposed model. The performance pairs of each model are shown in Table 4. (FM is the number of target tracking interrupts; GT is the total number of theoretical frames; IDSW is the number of target ID switching; MOTA is the precision of multi-target tracking.)

Table 4. Comparison of model performance before and after improvement.

Model	FPS	FP	FN	FM	GT	IDSW	MOTA
Target Detector + DeepSORT	69	0	90	13	1591	11	0.9365
Target Detector + Target Tracker	69	0	85	10	1591	6	0.9435
YOLOv3 + Target Tracker	54	0	87	28	1591	8	0.9215
CenterNet + Target Tracker	25	0	503	13	1591	31	0.66436

4. Discussion

It can be seen from the test results of the target detector shown in Table 2 that, for UAV targets, reducing the target detection branch helps to improve the speed of model inference, which increases the speed by 5FPS, and thus, the performance loss is very small, less than one percentage point. The new data augmentation strategy can effectively improve the mAP of UAV target detection without affecting the reasoning speed. Our modification strategy allows the detection speed and accuracy of the model to reach new heights, indicating that the modification is effective.

As can be seen from the test results of the overall model shown in Table 3, adding Euclidean distance to the target tracking model as a supplement to cascading matching

effectively improves the tracking performance of the target tracker by 0.9% and improves the ID switching problem. By comparing with the performance of YOLOv3+DeepSORT and CenterNet + DeepSORT, we can also see that the tracking speed and accuracy of our model have been greatly improved, which is enough to prove that the direction of our model selection and modification is correct. Overall, the improved detection-tracking model achieves the target tracking performance of 94.35% accuracy while achieving the real-time speed of 69FPS.

5. Conclusions

Aiming at solving the monitoring problem for small UAVs, a feasible and effective solution is proposed in this study. With the high-speed data transmission of 5G cameras, an improved "detection-tracking" model composed of a target detector and a target tracker was developed, which could monitor the target accurately and at a high speed. Additionally, the modification of the corresponding module of the model makes the model more suitable for monitoring small aircraft with a slow flight speed, a small size and a complex flight environment compared with the traditional tracking model. It can also be seen from the experimental results that the speed of this model was significantly improved without reducing the detection accuracy and it can effectively deal with the common problems of multi-target tracking, such as target loss and mutual occlusion. In order to verify the implementability of the model, we deployed the algorithm on a DPU connected with a camera for testing and achieved high accuracy and identification speed. In addition, the DPU with the detection model deployed can be applied in more diverse scenarios. For example, it can be used as a module of UAV to realize monitoring in more complex environments. To sum up, the model proposed in this study can effectively realize the real-time recognition and tracking of multiple small UAV targets simultaneously.

Author Contributions: Investigation, L.F.; Data curation, Q.Y.; Writing–original draft, H.L.; Writing—review & editing, T.H. and M.K.; Supervision, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: Tao Hong is supported by the National Natural Science Foundation of China under Grant No.61827901. Hongming Liang is supported by the Central Guidance on Local Science and Technology Development Special Fund of Shenzhen City under Project No.2021Szvup079.

Data Availability Statement: Not applicable.

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

References

1. Chen, K.; Li, H.; Li, C.; Zhao, X.; Wu, S.; Duan, Y.; Wang, J. An Automatic Defect Detection System for Petrochemical Pipeline Based on Cycle-GAN and YOLO v5. *Sensors* **2022**, *22*, 7907. [CrossRef] [PubMed]
2. Chohan, U.W.; van Kerckhoven, S. *Activist Retail Investors and the Future of Financial Markets: Understanding YOLO Capitalism*; Taylor and Francis: Oxfordshire, UK, 2023.
3. Li, Y.; Fu, C.; Ding, F.; Huang, Z.; Pan, J. Augmented Memory for Correlation Filters in Real-Time UAV Tracking. In Proceedings of the International Conference on Intelligent Robots and Systems, Las Vegas, NV, USA, 24 October 2020–24 January 2021.
4. Soft Computing. Researchers from Shanghai Jiao-Tong University Detail New Studies and Findings in the Area of Soft Computing (Collaborative model based UAV tracking via local kernel feature). *Comput. Wkly. News* **2018**.
5. Aglyamutdinova, D.B.; Mazgutov, R.R.; Vishnyakov, B.V. Object Localization for Subsequent UAV Tracking. In Proceedings of the ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Riva del Garda, Italy, 4–7 June 2018; Volume XLII-2.
6. Sun, Z.; Wang, Y.; Gong, C.; Laganiére, R. Study of UAV tracking based on CNN in noisy environment. *Multimed. Tools Appl.* **2020**, *80*. [CrossRef]
7. Xie, J.; Huang, S.; Wei, D.; Zhang, Z. Multisensor Dynamic Alliance Control Problem Based on Fuzzy Set Theory in the Mission of Target Detecting and Tracking. *J. Sens.* **2022**, *2022*, 7919808. [CrossRef]
8. Kwok, D.; Nejo, T.; Costello, J.; Okada, H. IMMU-31. Tumor-Specific Alternative Splicing Generates Spatially-Conserved Hla-Binding Neoantigen Targets Detected Through Integrative Transcriptomic and Proteomic Analyses. *Neuro Oncol.* **2021**, *23* (Suppl. 6), vi99. [CrossRef]

9. DENSO TEN Limited. Patent Issued for Radar Device and Target Detecting Method (USPTO 10,712,428). *Comput. Netw. Commun.* **2020**.
10. Chen, J.; Wang, H.; Zhang, H.; Luo, T.; Wei, D.; Long, T.; Wang, Z. Weed detection in sesame fields using a YOLO model with an enhanced attention mechanism and feature fusion. *Comput. Electron. Agric.* **2022**, *202*, 107412. [CrossRef]
11. Yu, Z.; Zhongyin, G.; Jianqing, W.; Yuan, T.; Haotian, T.; Xinming, G. Real-Time Vehicle Detection Based on Improved YOLO v5. *Sustainability* **2022**, *14*, 12274.
12. Tan, L.; Lv, X.; Lian, X.; Wang, G. YOLOv4_Drone: UAV image target detection based on an improved YOLOv4 algorithm. *Comput. Electr. Eng.* **2021**, *93*, 107261. [CrossRef]
13. Hansen, L.; Kuangang, F.; Qinghua, O.; Na, L. Real-Time Small Drones Detection Based on Pruned YOLOv4. *Sensors* **2021**, *21*, 3374.
14. Jinhui, L. Multi-target detection method based on YOLOv4 convolutional neural network. *J. Phys. Conf. Ser.* **2021**, *1883*, 012075.
15. Fei, C.; Huanxin, Z.; Xu, C.; Runlin, L.; Shitian, H.; Juan, W.; Li, S. *Remote Sensing Aircraft Detection Method Based on LIGHTWEIGHT YOLOv4*; National University of Defense Technology: Changsha, China, 2021.
16. Li, F.; Gao, D.; Yang, Y.; Zhu, J. Small target deep convolution recognition algorithm based on improved YOLOv4. *Int. J. Mach. Learn. Cybern.* **2022**, prepublish. [CrossRef]
17. Jun, W.S.; Fan, P.Y.; Gang, C.; Li, Y.; Wei, W.; Zhi, X.C.; Zhao, S.Y. Target Detection of Remote Sensing Images Based on Deep Learning Method and System. In Proceedings of the 2021 3rd International Conference on Advanced Information Science and System, Sanya, China, 26–28 November 2021; pp. 370–376. [CrossRef]
18. Li, X.; Luo, H. An Improved SSD for Small TARGET detection. In Proceedings of the 2021 6th International Conference on Multimedia and Image Processing (ICMIP 2021), Zhuhai, China, 8–10 January 2021; pp. 15–19. [CrossRef]
19. Sun, W.; Yan, D.; Huang, J.; Sun, C. Small-scale moving target detection in aerial image by deep inverse reinforcement learning. *Soft Comput.* **2020**, *24*, 5897–5908. [CrossRef]
20. Andrade, R.O.; Yoo, S.G.; Ortiz-Garces, I.; Barriga, J. Security Risk Analysis in IoT Systems through Factor Identification over IoT Devices. *Appl. Sci.* **2022**, *12*, 2976. [CrossRef]
21. Zhimin, G.; Yangyang, T.; Wandeng, M. A Robust Faster R-CNN Model with Feature Enhancement for Rust Detection of Transmission Line Fitting. *Sensors* **2022**, *22*, 7961.
22. Mian, Z.; Peixin, S.; Xunqian, X.; Xiangyang, X.; Wei, L.; Hao, Y. Improving the Accuracy of an R-CNN-Based Crack Identification System Using Different Preprocessing Algorithms. *Sensors* **2022**, *22*, 7089.
23. Ren, J.; Jiang, X. A three-step classification framework to handle complex data distribution for radar UAV detection. *Pattern Recognit.* **2021**, *111*, prepublish. [CrossRef]
24. Guo, X.; Zuo, M.; Yan, W.; Zhang, Q.; Xie, S.; Zhong, I. Behavior monitoring model of kitchen staff based on YOLOv5l and DeepSort techniques. *MATEC Web Conf.* **2022**, *355*, 03024. [CrossRef]
25. Qiu, X.; Sun, X.; Chen, Y.; Wang, X. *Pedestrian Detection and Counting Method Based on YOLOv5+DeepSORT*; Tibet University: Lhasa, China, 2021.
26. He, Y.; Li, X.; Nie, H. A Moving Object Detection and Predictive Control Algorithm Based on Deep Learning. *J. Phys. Conf. Ser.* **2021**, *2002*, 012070. [CrossRef]
27. Rueda, M.G.V.; Hahn, F. Target detect system in 3D using vision apply on plant reproduction by tissue culture. In Proceedings of the Aerospace/Defense Sensing, Simulation, and Controls, Orlando, FL, USA, 21 March 2001; Volume 4390.
28. Li, W.; Feng, X.S.; Zha, K.; Li, S.; Zhu, H.S. Summary of Target Detection Algorithms. *J. Phys. Conf. Ser.* **2021**, *1757*, 012003. [CrossRef]
29. Durga, B.K.; Rajesh, V. A ResNet deep learning based facial recognition design for future multimedia applications. *Comput. Electr. Eng.* **2022**, *104*, 108384. [CrossRef]