electronics

MDPI

Article

A Video-Based Real-Time Tracking Method for Multiple UAVs in Foggy Weather

Jiashuai Dai *, Ling Wu and Pikun Wang

College of Weapon Engineering, Naval University of Engineering, Wuhan 430000, China
* Correspondence: dai_js@163.com; Tel.: +86-15-725211341

Abstract: Aiming at the real-time tracking problem of multiple unmanned aerial vehicles (UAVs) based on video under fog conditions, we propose a multitarget real-time tracking method that combines the Deepsort algorithm with detection based on improved dark channel defogging and improved You Only Look Once version 5 (YOLOv5) algorithm. The contributions of this paper are as follows: 1. For the multitarget tracking problem under fog interference, a multialgorithm combination method is proposed. 2. By optimizing dark channel defogging, the complexity of the original algorithm is reduced from $O(n^2)$ to $O(n)$, which simplifies the processing time of the defogging algorithm. 3. The YOLOv5 network structure is optimized so that the network can synchronously reduce the detection time while maintaining high-precision detection. 4. The amount of algorithm processing through image size compression is reduced, and the real-time performance under high-precision tracking is improved. In the experiments conducted, the proposed method improved tracking precision by 36.1% and tracking speed by 39%. The average time of tracking per image frame was 0.036s, satisfying the real-time tracking of multiple UAVs in foggy weather.

Keywords: real-time tracking of multiple UAVs; dark channel defogging; YOLOv5; Deepsort; deep learning

check for updates

Citation: Dai, J.; Wu, L.; Wang, P. A Video-Based Real-Time Tracking Method for Multiple UAVs in Foggy Weather. *Electronics* **2022**, *11*, 3576. https://doi.org/10.3390/electronics11213576

Academic Editor: Carlos Tavares Calafate

Received: 20 September 2022
Accepted: 28 October 2022
Published: 1 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

UAVs have been widely used in diverse missions. Because UAVs have the characteristics of high flexibility and good concealment, they play an important role in the military field. The use of UAVs at sea to detect and attack naval vessels poses a threat to the safety of navigation. Therefore, effective detection and tracking of unmanned aerial vehicle targets are of great significance. In foggy weather, the characteristics of UAVs in the video are weakened, which reducing the tracking accuracy.

Presently, video-based multiobject tracking mainly uses the following methods: labeled multi-Bernoulli multiobject tracking algorithm based on detection optimization [1], multiobject tracking algorithm for position prediction by combining deep neural network SiamCNN with contextual information [2], and pedestrian tracking algorithm with a combination of YOLO detection algorithm and Kalman filter [3]. These methods focus on problems such as tracking loss caused by occlusion, target ID matching error in the tracking process, and missing tracking targets. In the process of detection and tracking, there is a negative correlation between tracking accuracy and algorithm processing time. The realization of high-precision tracking often leads to an increase in algorithm time, which makes it difficult to meet real-time performance.

Recently, scholars have carried out research on image defogging. Salazar-Colores et al., (2020) proposed a novel methodology based on depth approximations through DCP, local Shannon entropy, and fast guided filter for reducing artifacts and improving image recovery in sky regions with low computation time [4]. Liu et al., (2019) combined dark channel defogging with a KCF tracking algorithm for multitarget tracking under fog conditions [5]. However, this method needs to manually label multiple target frames in the first image.

Once the tracking target is lost, it will not be able to complete the follow-up tracking task. Defogging is mainly achieved by image enhancement [6–9] and using a physical model [10–14]. Using image enhancement for image defogging does not consider the influence of fog on the image but achieves the defogging effect by adjusting the image contrast. Meanwhile, the physical model for image defogging takes into account the foggy image generation process and other factors, with dark channel defogging used as a typical algorithm.

In order to achieve real-time tracking for multiple UAV targets, we selected the YOLOv5 algorithm, which has excellent speed performance in the current target detection field, to carry out the UAV target detection task. In the process of matching and tracking, considering the UAV motion has the characteristics of direction and speed mutation, conventional tracking methods such as Kalman filtering and SORT algorithm are prone to cause tracking and matching errors. Therefore, we chose the Deepsort algorithm to perform track correlation for the detected UAV target. The Deepsort algorithm can combine the motion state and appearance characteristics of the target to perform matching and correlation in the tracking process. It has good tracking ability for moving direction and speed mutation targets.

We implemented and tested a "detection-based tracking" algorithm for multiple UAVs in foggy weather by combining an improved dark channel algorithm with improved YOLOv5 and Deepsort [15–17]. Through these improvements, we reduced the complexity of defogging algorithm from $O(n^2)$ to $O(n)$ and simplified the complexity of YOLOv5, thus reducing the time of the defogging and detection algorithm. Compared to target detection and tracking without fog interference, the introduction of defogging processing makes the algorithm spend more time on single-frame image processing. For this reason, images are compressed without distortion to further ensure real-time tracking. The specific process is given in Figure 1.

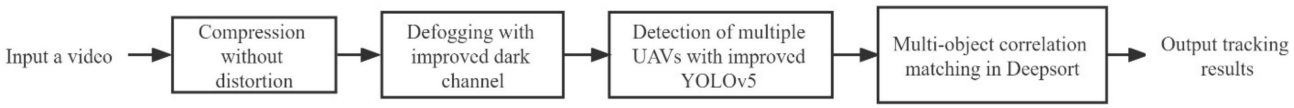


Figure 1. Process flow of tracking multiple UAVs in foggy weather.

2. Image Defogging Algorithm Based on Improved Dark Channel

The dark channel defogging algorithm uses the following model for image defogging [18]:

$$J(x) = \frac{I(x) - A(1 - t(x))}{t(x)} \quad (1)$$

where x stands for the pixel spatial coordinate of the image, $I(x)$ represent the captured foggy images, $J(x)$ represents the restored fogless images; A is the global atmospheric light value, and $t(x)$ is the transmissivity that can be estimated by Equation (1) when J^{dark} tends to zero according to the dark channel prior theory [19]. Moreover, the value of A is the maximum grayscale of the pixels in the original images with the top 0.1% luminance in the dark channel [20]. The process flow of the dark channel defogging algorithm is presented in Figure 2.

2.1. Determination of Transmissivity by Mean Filtering

In the dark channel defogging algorithm, the minimum filter is often used. However, after minimum filter processing, the restored defogging image has obvious white edges at the edge of the UAV target. This phenomenon affects the edge features of the UAV target itself and is not computer friendly for UAV target recognition. To solve this problem, we used mean filtering to process the dark channel image to estimate the transmittance. Additionally, the defogging coefficient is correlated with foggy images in dark channel defogging to achieve adaptive adjustment of defogging. The detailed calculation process is as follows.

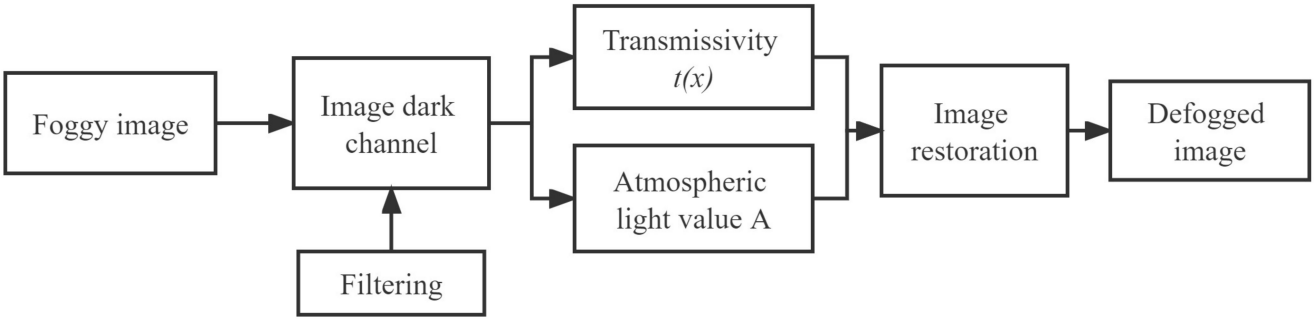


Figure 2. Process flow of the dark channel defogging algorithm.

In Equation (1), $I(x)$, $J(x)$, $t(x)$, and A are all above 0 and J^{dark} tends to zero:

$$A(1 - t(x)) = I(x) \tag{2}$$

We assumed a constant atmospheric light value, A, calculated from the minimum intensity in the R, G, and B channels:

$$A_0(1 - t(x)) = \min_{c\in\{r,g,b\}}(I^c(x)) \tag{3}$$

It is transformed to

$$t(x) = 1 - \frac{\min\limits_{c\in\{r,g,b\}}(I^c(x))}{A_0} \tag{4}$$

Here, mean filtering is carried out on the right side of the equation, and the transmittance $t(x)$, is calculated. The result after mean filtering can reflect the general trend of $t(x)$, but there is a certain absolute value difference between it and the real value. Therefore, an offset value $\varphi(0 < \varphi < 1)$ is made up for the filtering result. Moreover, to simplify the representation, $average_{sa}(\min\limits_{c\in\{r,g,b\}}(I^c(x)))$ is substituted with $\mathrm{M_{ave}}(x)$, where the average represents the mean filter processing, and sa represents the window size of the filter. The approximate evaluation of transmissivity is obtained as follows:

$$t(x) = 1 - \frac{\mathrm{M_{ave}}(x)}{A_0} + \varphi\frac{\mathrm{M_{ave}}(x)}{A_0} \tag{5}$$

Let $\delta = 1 - \varphi$, the above equation is expressed as follows:

$$t(x) = 1 - \delta\frac{\mathrm{M_{ave}}(x)}{A_0} \tag{6}$$

δ can regulate the darkness of images restored after defogging. The larger the value of δ, the lower the transmissivity $t(x)$ and the darker the restored image. To enable δ to dynamically adjust the brightness after defogging according to the fog image, δ can be set to be associated with the pixels of the original image, and the formula is as follows:

$$\delta = \rho\mathrm{m_{av}} \tag{7}$$

where $\mathrm{m_{av}}$ is the mean value of all elements in $\mathrm{M}(x)$, that is, the mean value of the minimum pixel at each pixel coordinate x of RGB channels in an original foggy image. Moreover, $0 \le \rho \le 1/\mathrm{m_{av}}$. If the value of δ is too low, it may lead to lower transmissivity and result in dark images after defogging. For this reason, the maximum threshold of δ is set to 0.9. Thus, we have

$$\delta = \min(\rho\mathrm{m_{av}},\ 0.9) \tag{8}$$

Equations (3), (6) and (8) are combined to obtain the following:

$$t(x) = \max\begin{pmatrix} 1 - \min(\rho \mathrm{m_{av}}, 0.9)\frac{\mathrm{M_{ave}}(x)}{A_0}, \\ 1 - \frac{\min\limits_{c\in\{r,g,b\}}(I^c(x))}{A_0} \end{pmatrix} \tag{9}$$

2.2. Estimation of Global Atmospheric Light Value

During dark channel defogging, the positions of the top 0.1% of pixels in the dark channel are determined. This operation requires comparing the pixel information of all pixels with other pixels to arrange them in order. We assumed that the number of pixels in the image is n. Then, the algorithm complexity of this operation reaches $O(n^2)$ and its operation amount will increase significantly with the increase in the image size.

In this study, we directly used the combination of the maximum pixels of the filtered image dark channel and the maximum pixels of the RGB channel of the original image to estimate the atmospheric light value. In the algorithm process, only the relationship between the current pixel value and the maximum pixel value needs to be compared, thus greatly reducing the computational complexity of the algorithm. The luminance of the restored image is slightly lower on the whole, but the complexity of the defogging algorithm can be reduced from $O(n^2)$ to $O(n)$, thus shortening the processing time of the algorithm.

Filtering is performed for inequality (4) to obtain

$$average_{sa}\left(1 - \frac{\mathrm{M}(x)}{A_0}\right) = 1 - \frac{average_{sa}(\mathrm{M}(x))}{A_0} = 1 - \frac{\mathrm{M_{ave}}(x)}{A_0} \in [0, 1] \tag{10}$$

From Equation (10), we obtain

$$\mathrm{M_{ave}}(x) \le A_0 \tag{11}$$

and

$$A_0 \le \max\left(\max_{c\epsilon\{r,g,b\}}(I^c(x))\right) \tag{12}$$

hence, A_0 may be expressed as

$$A_0 = \varepsilon \max\left(\max_{c\epsilon\{r,g,b\}}(I^c(x))\right) + (1-\varepsilon)\max(\mathrm{M_{ave}}(x)) \tag{13}$$

where $0 \le \varepsilon \le 1$.

The implementation steps of the algorithm are given in Table 1.

Table 1. Improved dark channel defogging algorithm steps.

1: Input foggy image $I(x)$.
2: Obtain a dark channel image.

$$M(x) = \min_{c\in\{r,g,b\}}(I^c(x))$$

3: Average filtering of M (x).

$$\mathrm{M_{ave}}(x) = average_{sa}\left(\min_{c\in\{r,g,b\}}(I^c(x))\right)$$

4: Calculate the mean value of all elements in $\mathrm{M_{ave}}(x)$ to obtain $\mathrm{m_{av}}$.
5: Calculate the global atmospheric light value.

$$A_0 = \varepsilon \max\left(\max_{c\epsilon\{r,g,b\}}(I^c(x))\right) + (1-\varepsilon)\max(\mathrm{M_{ave}}(x))$$

6: Calculate transmittance $t(x)$.

$$t(x) = \max\left(1 - \min(\rho \mathrm{m_{av}}, 0.9)\frac{\mathrm{M_{ave}}(x)}{A_0}, 1 - \frac{\min\limits_{c\in\{r,g,b\}}(I^c(x))}{A_0}\right)$$

7: Output: calculate the defogging image.

$$J(x) = \frac{I(x) - A(1-t(x))}{t(x)}$$

3. Tracking of Multiple UAVs with Improved YOLOv5 and Deepsort

After defogging the video frame of UAVs under fog conditions, an algorithm combining improved YOLOv5 and Deepsort was employed to track multiple UAVs.

The network structure of YOLOv5 is divided into input, backbone, neck, and prediction [21] as given in Figure 3.

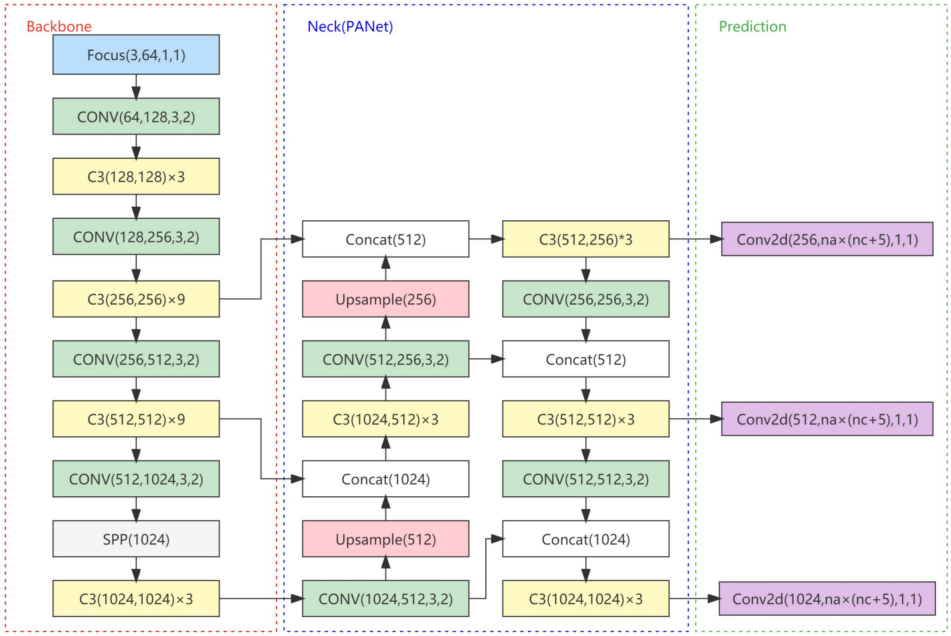


Figure 3. Network structure of YOLOv5.

3.1. Optimization and Improvement of YOLOv5 Network

With highly precise detection of interesting objects, the YOLOv5 network structure can shorten the time of detection. Before target detection and tracking, the defogging algorithm is introduced to defog the fog image, which increases the processing time. Although the defogging time is reduced by improving dark channel defogging, the overall algorithm still cannot meet the real-time performance requirements. The YOLOv5 network structure was optimized and improved to further shorten detection.

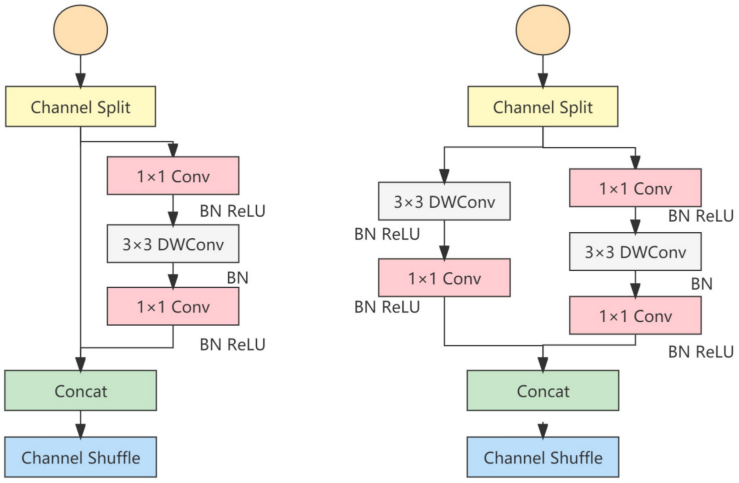
3.1.1. Removal of Focus Layer

The focus module is used in YOLOv5 to slice an image and expand the three RGB channels of the original image into 12 channels. Further convolution generates an initial down-sampled feature image that retains the valid information of the original image while improving the processing speed because of less calculation and lower parameters. However, frequent slice operation in the focus module increases the amount of calculation and parameters. In the process of sampling $640 \times 640 \times 3$ images to obtain $320 \times 320 \times 3$ feature maps, the amount of calculation and parameters of the focus operation becomes four times that of the ordinary convolution operation. As revealed in experiments, convolution can replace the focus module and perform satisfactorily without side effects. Hence, the focus layer was removed to further improve the speed.

3.1.2. Backbone Optimization Based on ShuffleNet V2

The backbone of YOLOv5 adopts the C3 module to extract object features, which is easier and faster than BottleneckCSP [22] in the previous versions. However, the C3 module utilizes multiple separable convolutions, so it occupies a large portion of memory if there are many channels and it is frequently implemented. In this case, the speed of the equipment is reduced to some extent. As a lightweight network model, ShuffleNet V2 [23] contains two block structures. As shown in Figure 4, Structure a has a channel split, so the

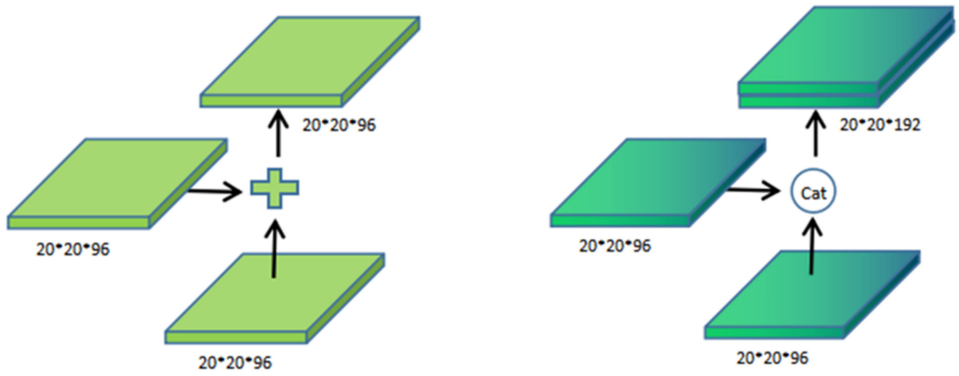
input feature image with c input channels is split into two branches: c1 and c2. The branch c2 is concatenated with c1 after three-layer convolution. Through the control module, input and output channels are kept consistent to speed up the reasoning speed of the model. Subsequently, channel shuffle is performed to reduce the branches of the network structure, improve the parallelism of the network, and shorten the duration of processing. Therefore, Structure a is mainly used to deepen the layers of the network. Meanwhile, Structure b has the same right branch as Structure a. Its left branch is concatenated, and the channel is shuffled with the right branch after convolution of input features. However, this structure cancels the channel split so as to allow the expansion of module output channels. Therefore, it was mainly employed to downsample and compress the feature layer.



(**a**) (**b**)

Figure 4. Structure of ShuffleNet V2 model. (**a**) ShuffleNet V2 structure. (**b**) ShuffleNet V2 structure.

In the neck layer, the improved network maintained the same structure of feature pyramid network (FPN) and pyramid attention network as in YOLOv5. However, the PAN structure contained as many output channels as input channels. Moreover, the "cat" operation was adjusted to "add", which further optimized memory access and utilization, as shown in Figure 5. The channels of the original YOLOv5 were pruned to redesign the network structure.



(**a**) (**b**)

Figure 5. Comparison of neck with PAN structure. (**a**) Improved neck PAN structure. (**b**) PAN structure in YOLOv5.

The optimization and improvement of YOLOv5 were achieved by deleting the focus module, replacing the backbone with the ShuttleNet module, and optimizing the network

in the neck layer. The improved detection network became less complex and processed faster. Its structure is presented in Figure 6.

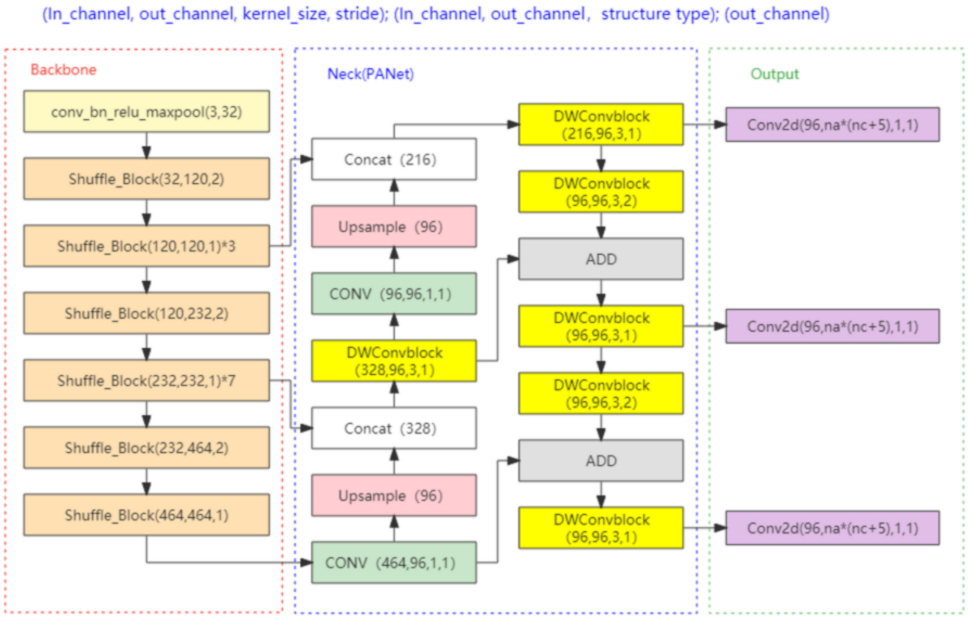


Figure 6. Detection network structure of improved YOLOv5.

3.2. Tracking of Multiple UAVs Based on Deepsort

The Deepsort algorithm takes the output of the YOLOv5 detector as its input to select the boxes for object detection and calculate the object association for matching. The tracking process flow [24] is presented in Figure 7.

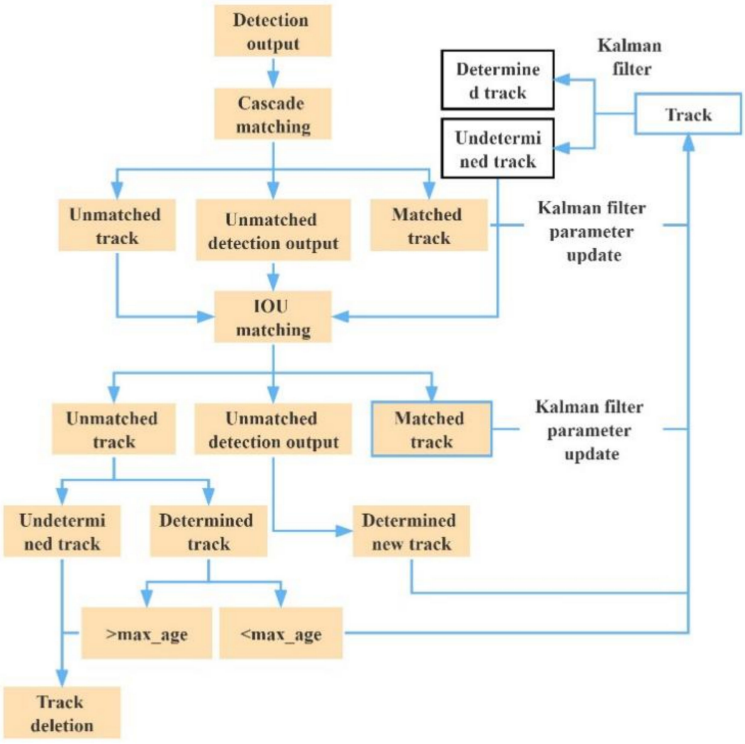


Figure 7. Process flow of Deepsort algorithm.

Multiobject tracking by Deepsort contains two operations, i.e., estimation of object state and matching and association of objects. In the first operation, Deepsort utilizes eight-dimensional state space ($u, v, \gamma, h, \dot{x}, \dot{y}, \dot{\gamma}$, and $\dot{h}$) and to record the motion state and information of objects. On this basis, the Kalman filter is employed to predict the motion state of objects. The motion information of objects predicted with the current frame is matched and associated with the object detection result output by the next frame. In the second operation, Deepsort associates the motion features of objects by virtue of Mahalanobis distance. In order to overcome the matching error caused by the abrupt change of object speed and jitter of shooting equipment because of single Mahalanobis distance, the appearance feature matching of objects is introduced for compensation. These two measures are combined through linear weighting into the final measure. Therefore, it can offer reliable association and matching with short-term prediction and compensate for the ID switch during object occlusion and camera shake.

4. Image Compression

While tracking multiple UAVs in a foggy video, the defogging algorithm is introduced, but it extends the time of processing frames in the video and lowers the frame rate. Hence, it is difficult to achieve real-time processing. In order to improve the processing speed of a single frame without reducing the accuracy [25,26], we used the method based on bilinear interpolation [27] to compress the image before defogging. The details are as follows.

Assuming that the size of the source image is a × b and the size of the compressed image is m × n, the horizontal and vertical compression ratios of the image are a/m and b/n. The pixel value of the compressed image can then be calculated based on the pixel ratio and the pixel points in the source image. Assuming that the coordinates of the pixel points in the compressed image are (I, J), the pixel points corresponding to the original image should be (i × a/m, j × b/m). Because i × a/m and j × b/m are not integers, the four adjacent pixels of the point are interpolated to obtain the pixel value of the point. The schematic diagram of such an operation is presented in Figure 8.

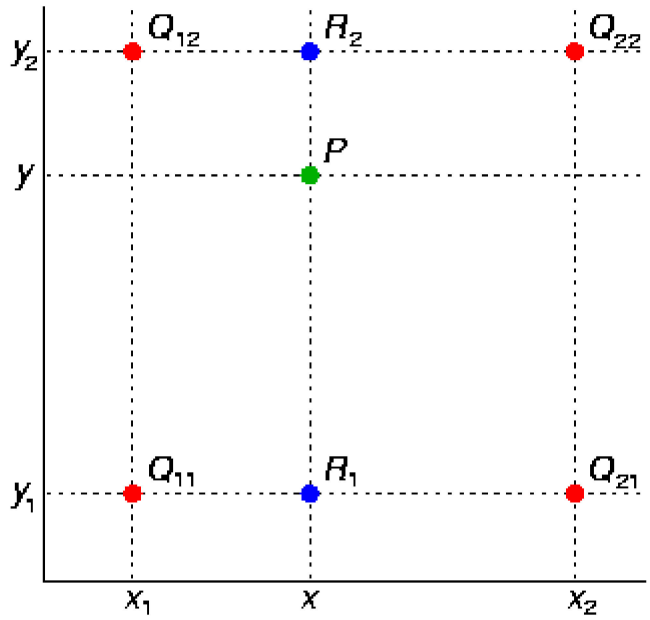


Figure 8. Schematic diagram of bilinear interpolation.

In Figure 8, P represents the pixel point of the source image corresponding to the pixel point of the compressed image; Q_{11}, Q_{12}, Q_{21}, and Q_{22} are four pixel points adjacent to point P in the original image. Q_{11} and Q_{21} points can be calculated in the horizontal direction to obtain the pixel value of R_1 point, and Q_{12} and Q_{22} points can be calculated in

the horizontal direction to obtain the pixel value of R_2 point. The specific calculation is as follows:

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_2}{x_2 - x_1} f(Q_{21}) \quad (14)$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (15)$$

R_1 and R_2 points can be interpolated in the vertical direction to obtain the pixel value of the P point:

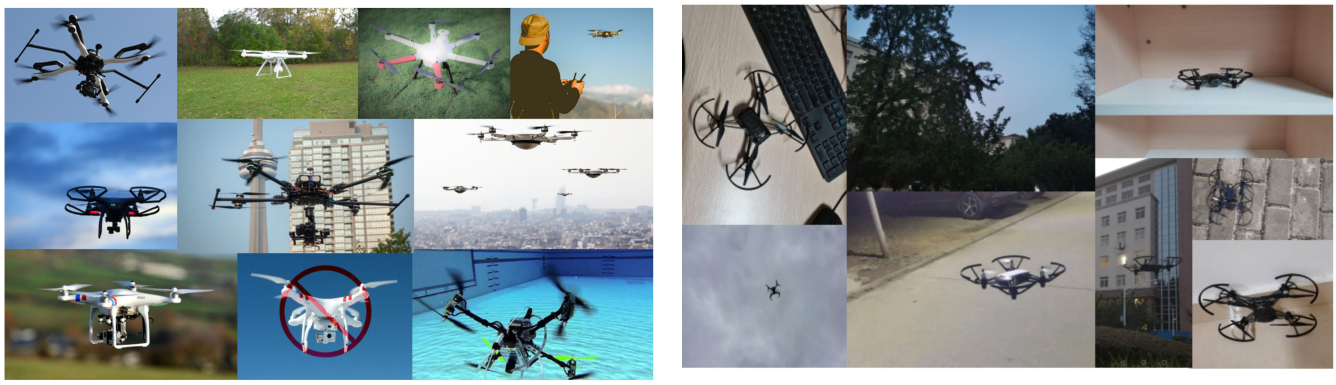
$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2) \quad (16)$$

5. Experiment Design and Result Analysis

5.1. Preparation for Experiments

Experiments were performed on a personal computer with Windows 10 operating system, Intel(R) Core(TM) i7-10750 processor, 16 G memory, NVIDIA Quadro (Lenovo, Beijing, China)T1000 GPU, PyTorch deep learning framework, and version 10.2 CUDA.

Because there is no public dataset for UAV target in the current target detection dataset, we built a UAV dataset for network model training by downloading network images and taking UAV flight photos. A total of 1727 UAV photos were collected. Each image contained one to six UAVs covering rotor UAVs of different models. The background included sky, land, underwater, city, vegetation, and other background environments. Some UAV images are shown in Figure 9.



(**a**) (**b**)

Figure 9. Images of UAV dataset. (**a**) UAV images downloaded from the network. (**b**) UAV images taken by a mobile phone.

The tool Labeling was used to label the UAV targets in the self-built dataset. The dataset was in the format of common objects in context (COCO). The dataset annotation process is shown in Figure 10.

After the dataset was labeled, extensible markup language (XML) documents were produced for the image. These XML documents contained object type and border information, particularly the type and position of labeled objects in the image. From the dataset, 1727 pieces of the images were labeled and classified at the ratio of 7:3 into the training set and testing set for 300 iterations.

In the experiment, the indexes for evaluating detection effect were accuracy, recall, and average precision. Multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) were taken as the indexes for the detection precision of multiple UAVs. The ID switches (IDs) of object tracking while matching indicated the stability of object tracking [28]. The processing time per frame of the image was taken as an index for speed.

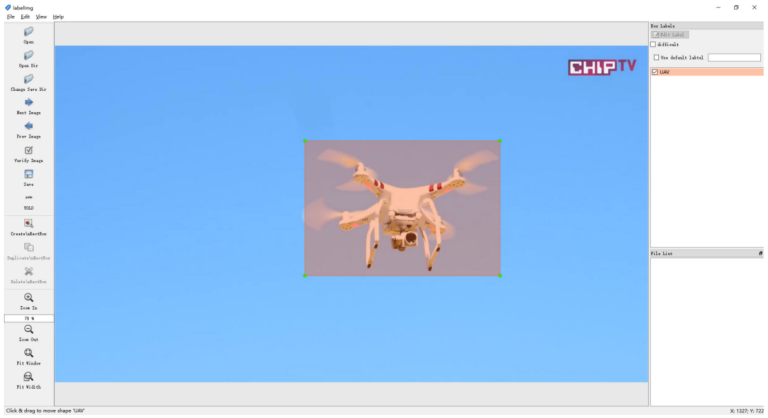


Figure 10. Image labeling of the dataset.

5.2. Comparison of Algorithm Improvement Effects

5.2.1. Comparison of YOLOv5 Network Improvement Effect

In order to verify the effect of our improvement to the YOLOv5 network structure, we carried out target detection experiments on the same dataset with the original YOLOv5s model and the improved YOLOv5 model; evaluated the complexity of the model with floating point operations (FLOPs), parameters, inference time, and model size; and evaluated the detection effect with average precision (AP). The experimental results are shown in Table 2.

Table 2. Comparison of YOLOv5 and improved YOLOv5 network structures.

Model	FLOPs	Parameters	Model Size	Reference Time	AP0.5	AP0.5:0.95
YOLOv5	16.3 G	7,053,910	13.7 M	27.9 ms	0.977	0.465
Improved YOLOv5	2.6 G	719,926	1.58 M	14.2 ms	0.99	0.503

Based on Table 2, the following conclusions were drawn.

Compared to the original YOLOv5 network model, the complexity of the improved YOLOv5 network model was reduced. The FLOPs were only 1/6 of the original, and the number of parameters was reduced to 1/10 of the original. In addition, the size of the model was reduced from 13.7 to 1.58 M, which greatly improved the inference time in the detection process, and the accuracy of the detection was also maintained at a high level.

5.2.2. Multi-UAV Target Tracking Experiments on UAVs under Normal Illumination

We used mobile devices to record the flight video images of three UAVs and verified the tracking ability of the improved algorithm for UAV targets based on the recorded video. During the video capture process, the UAVs were controlled to fly in a stable state, cross the trajectory, and fly to a far distance.

Experiments were conducted to test the tracking ability of the algorithm for UAV flight in stable state, track cross occlusion, and small target and scale change. The specific tracking results are shown below.

By analyzing Figures 11–13 and Table 3, the following conclusions were drawn:

(1) For multiple UAV targets in stable flight state, YOLOv5 + Deepsort algorithm and improved YOLOv5 + Deepsort algorithm had stable tracking abilities.
(2) For multiple UAV targets with mutual occlusion, when the targets were separated, the algorithm could still track the target and keep the original target ID unchanged without the problem of target ID matching error.
(3) For small scale targets and scale changing targets, the algorithm could also accurately identify and track them. In the process of tracking multiple UAV targets, as long as

the flying targets did not exceed the video capture range, no IDs were generated for the algorithm.

(4) Moreover, the improved YOLOv5 algorithm could significantly improve the tracking speed. The average tracking frame rate of the original tracking algorithm was 15 FPS, and the tracking effect could not meet the real-time tracking requirements. By reducing the complexity of the original network, YOLOv5 greatly reduced the detection time in the tracking process. The average tracking frame rate reached 43 FPS, far exceeding the real-time tracking performance requirements. Real-time and high-precision tracking of multiple UAV targets was realized.

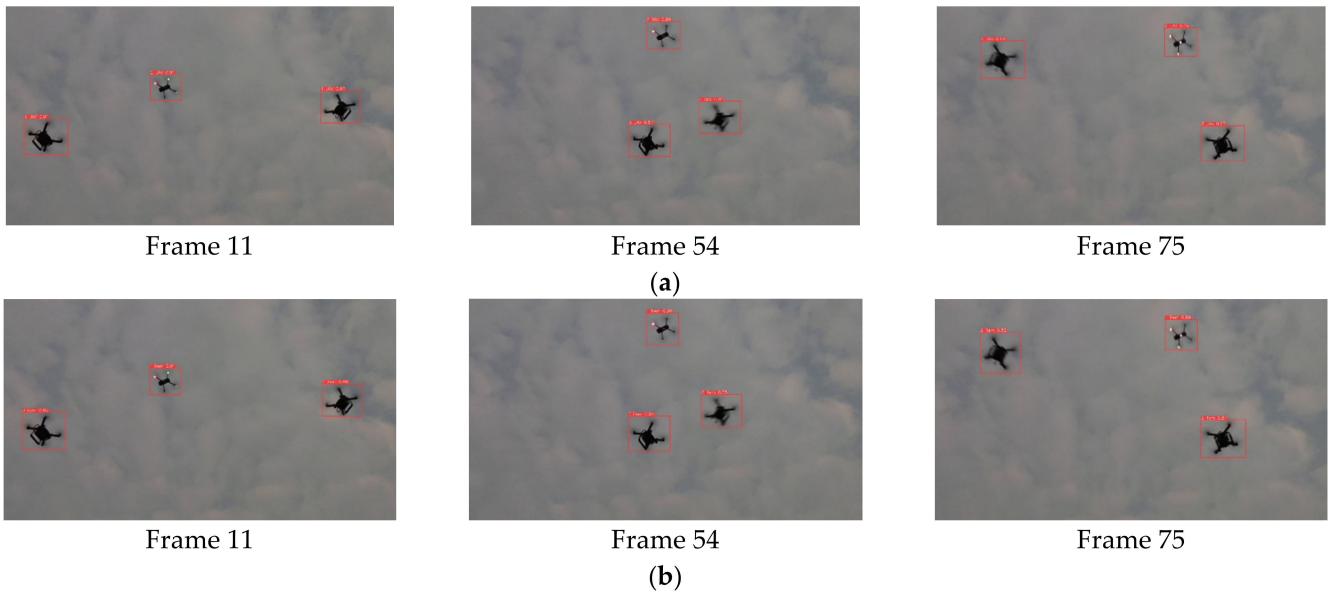


Figure 11. Algorithm tracking effect of UAVs in stable flight. (**a**) YOLOv5 + Deepsort; (**b**) improved YOLOv5 + Deepsort.

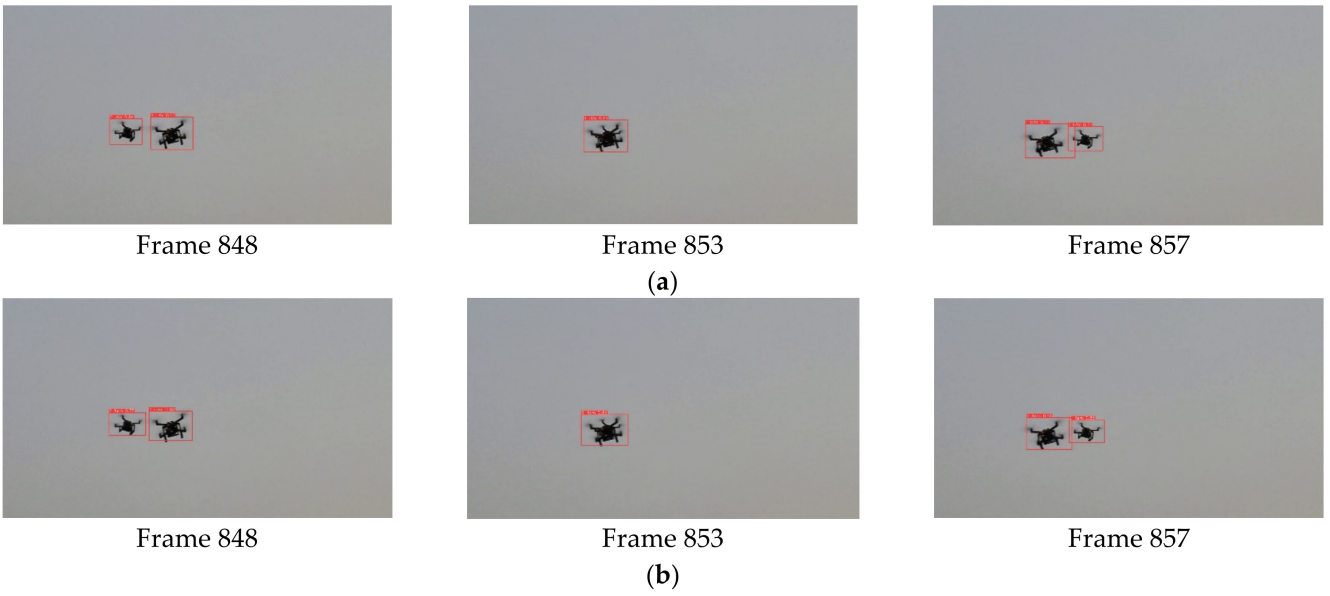
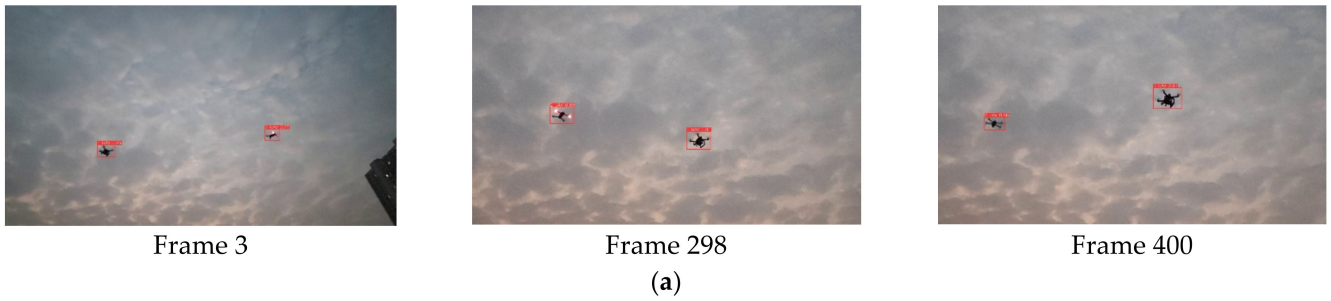


Figure 12. Algorithm tracking effect of UAVs in track cross occlusion. (**a**) YOLOv5 + Deepsort; (**b**) improved YOLOv5 + Deepsort.



(**a**)



(**b**)

Figure 13. Algorithm tracking effect of UAVs in small target and scale change. (**a**) YOLOv5 + Deepsort; (**b**) improved YOLOv5 + Deepsort.

Table 3. Target tracking effect index comparison of multiple UAVs.

Tracking Method	MOTA↑	MOTP↓	IDs↓	FPS
YOLOv5 + Deepsort	97.20%	0.241	2	15
Improved YOLOv5 + Deepsort	99.40%	0.227	2	43

5.3. Target Tracking Experiment of UAV under Fog Interference

5.3.1. Comparison of Detection Results with Different Defogging Algorithms

A self-produced dataset of UAVs was used to train the YOLOv5 model. Defogging was performed with different algorithms for 345 frames of image from the foggy video. The trained YOLOv5 was input for the object detection experiment. The average detection results with different defogging algorithms are given in Table 4. In the process of detection accuracy evaluation, we used the IOU value between the detection frame and the real frame as the judgment standard to determine whether the target detection position was accurate. In the calculation process, we set the IOU judgment threshold to 0.5. When the IOU value was higher than the set threshold, the predicted position for the current target was considered to be accurate.

Table 4. Comparison of detection results with different defogging algorithms for multiple UAVs.

Category	Index			
	Precision	Recall	AP0.5	AP0.5:0.95
Foggy image	0.739	0.87	0.876	0.381
Multiscale Retinex defogging	0.532	0.502	0.503	0.194
AHE defogging	0.757	0.829	0.87	0.375
CLAHE defogging	0.882	0.904	0.944	0.451
Dark channel defogging	0.894	0.813	0.928	0.49
Improved dark channel defogging	0.917	0.935	0.979	0.503

As revealed in Table 4, the improved dark channel defogging algorithm achieved the best results in terms of detection precision indexes. Hence, we conducted defogging on a foggy video image based on dark channel defogging and improved dark channel defogging algorithms to track multiple UAV targets.

5.3.2. Comparison of Target Tracking Results of Multiple UAVs under Fog Interference

In order to explore the tracking ability of the algorithm for multiple UAV targets under fog interference, we recorded flight videos of three UAVs in foggy weather and verified the tracking effect of the algorithm based on recorded videos. As the UAV target tracking ability under stable flight, mutual occlusion, and small scale and scale change had already been verified, as noted in Section 5.2.2, in order to ensure experimental safety, we only recorded relevant videos of UAV stable flight status under fog conditions. The test video contained 345 images in total, and the image size was 1920 × 1080.

An experiment was carried out to compare the tracking results of multiple UAVs with four methods. These methods were as follows:

Method 1: Unfogged video + YOLOv5 + Deepsort.

Method 2: Dark channel defogging + YOLOv5 + Deepsort

Method 3: Improved dark channel defogging + YOLOv5 + Deepsort.

Method 4: Improved dark channel defogging + improved YOLOv5 + Deepsort.

The 52nd frame of the image was selected from the video, and its tracking effect is shown in Figure 14. In this figure, a–d represents the tracking effects with Methods 1–4, respectively.



(**a**) (**b**) (**c**) (**d**)

Figure 14. Tracking effects with different algorithms. (**a**) Method 1; (**b**) Method 2; (**c**) Method 3; (**d**) Method 4.

As revealed in Figure 15, Method 1 could detect and track a UAV only at the confidence level of 0.78. Methods 2 and 3 could detect and track two UAVs at the confidence level of 0.67 and 0.67 and 0.71 and 0.83, respectively. However, Method 4 could detect all UAVs at the confidence level of 0.77, 0.82, and 0.47. Therefore, Method 4 had the best tracking effect.

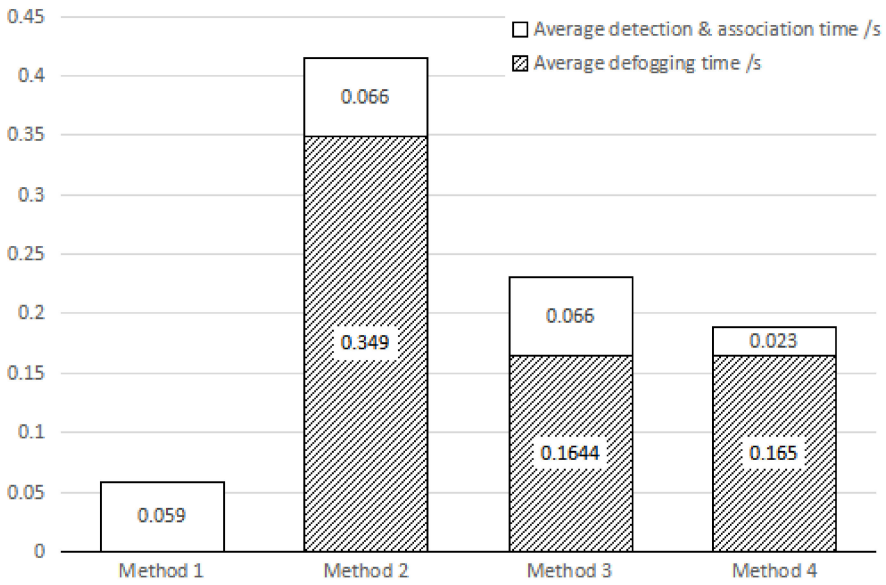


Figure 15. Average object tracking time per frame of the image with different algorithms.

In order to quantitatively evaluate the tracking effect, the tracking indicators under the four tracking methods were calculated. The specific tracking indicators are shown in Table 5 and Figure 15.

Table 5. Multiobject tracking indexes with different methods.

Defogging and Tracking Method	MOTA↑	MOTP↓	IDs↓	Tracking Time per Frame/s
Method 1	52.8%	0.315	1	0.059
Method 2	64.5%	0.322	0	0.415
Method 3	77.1%	0.263	0	0.230
Method 4	87.0%	0.215	0	0.188

Based on Table 5 and Figure 15, the following conclusions were drawn:

(1) Defogging could effectively improve the effects of multiobject tracking. In the experimental results of Method 1, the target detection for UAV failed due to fog interference, the tracking accuracy was only 52.8%, and the discontinuity of tracks led to the error of target ID matching and the generation of IDs. Generally, Method 4 achieved the best tracking indexes. It improved MOTA by 35.2% and MOTP by 31.7% compared to Method 1.

(2) Defogging increased the time of processing a single frame of the image. With Method 2, the time extended by 603.4% after defogging. It decreased by 54.7% compared to Method 4 after improving the dark channel and optimizing the YOLOv5 network structure, but it was still 218.6% higher than Method 1.

5.3.3. Real-Time Tracking with Compressed Images

It can be seen from Section 5.3.2 that by improving the dark channel defogging algorithm and YOOv5 network structure, the time of defogging and tracking algorithm was considerably shortened, but its average time of processing each frame of image was 0.188s, which cannot meet the requirement for real-time processing.

Considering the processing speed of the defogging algorithm is affected by the size of the input image, an image compression method was introduced to compress the size of the input image before the defogging algorithm, and bilinear interpolation was selected as the compression method. Target tracking experiments were conducted after compressing the same fog video outlined in Section 5.2.2. The tracking method used improved dark channel defogging + improved YOLOv5 + Deepsort. The tracking effect of different compression sizes are shown in Table 6 and Figure 16.

The analysis based on Table 6 and Figure 12 revealed the following:

(1) The smaller the compression size of images without distortion, the shorter the average defogging time and the shorter the tracking time per frame. However, when the size was too small, it led to a dramatic decrease in the tracking effect. When the size was further compressed, the size of the UAV target in the image was too small to highlight its feature information, so the tracking accuracy decreased and the discontinuity of tracking led to ID matching errors.

(2) Within a certain range of compression size without distortion, the accuracy of object tracking could exceed 85% in all cases. When the images were compressed to 576 × 324, the tracking time per frame was 0.036 s, and the accuracy of object tracking was 88.9%. Therefore, the requirements for accuracy and real-time processing were satisfied simultaneously.

(3) After the compression size of 576 × 324, Method 4 achieved a 36.1% higher tracking precision and 39% higher tracking speed than Method 1 in the tracking of multiple UAVs.

Table 6. Indexes for evaluating the tracking effect with different compression sizes.

Compression Size	MOTA↑	MOTP↓	IDs↓	Tracking Time per Frame/s
1920 × 1080	87.0%	0.215	0	0.188
960 × 540	88.3%	0.216	0	0.067
768 × 432	87.2%	0.218	0	0.047
576 × 324	88.9%	0.225	0	0.036
384 × 216	85.5%	0.242	0	0.029
192 × 108	51.5%	0.341	14	0.023

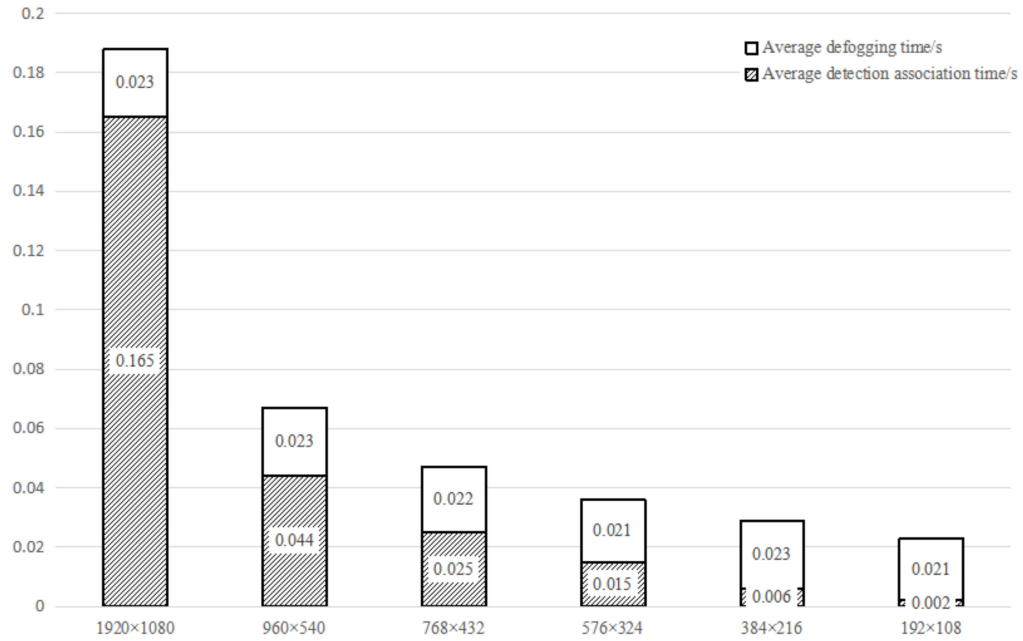


Figure 16. Comparison of tracking time with different compression sizes.

6. Conclusions

This paper presented the study and improvement of the dark channel defogging algorithm and YOLOv5 network structure. With the Deepsort algorithm, a video-based method for tracking multiple UAVs in foggy weather was designed and further combined with the technology of image compression without distortion. In this way, the proposed method can meet the requirement for real-time processing while ensuring highly precise and constantly stable tracking. Therefore, it is greatly practical.

Author Contributions: Conceptualization, L.W.; methodology, J.D.; software, J.D. and P.W.; validation, J.D., L.W. and P.W.; data curation, J.D.; writing—original draft preparation, J.D.; writing—review and editing, L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Jiang, L.; Yang, J. Detection Optimized Labeled Multi-Bernoulli Algorithm for Visual Multi-Target Tracking. *J. Front. Comput. Sci. Technol.* **2022**, 1–20.
2. Leal-Taixé, L.; Canton-Ferrer, C.; Schindler, K. Learning by Tracking: Siamese CNN for Robust Target Association. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June–1 July 2016; IEEE Press: New York, NY, USA, 2016; pp. 418–425.
3. Jiamin, R.; Ningsheng, G.; Zhenyang, H. Multi-target tracking algorithm based on YOLOv3 and Kalman Filter. *Comput. Appl. Softw.* **2020**, *37*, 169–176.
4. Salazar-Colores, E.U.S.; Moya-Sánchez, J.-M.; Ramos-Arreguín, E.; Cabal-Yépez, G.F.; Cortés, U. Fast Single Image Defogging with Robust Sky Detection. *IEEE Access* **2020**, *8*, 149176–149189. [CrossRef]
5. Liu, M.; Cheng, X.; Diao, C.; Chen, S. Object tracking algorithm based on kernelized correlation filter with dark channel defogging. *J. Tianjin Univ. Technol.* **2019**, *35*, 27339–27351.
6. Hongwei, W.; Jie, T.; Zhuopeng, X. A review of image defogging algorithms. *Softw. Guide* **2021**, *20*, 231–235.
7. Chen, J.; Xu, B. Research on machine vision image clearness under fog and haze weather conditions. *Comput. Eng.* **2017**, *43*, 280–285.
8. Chen, Z. Homomorphic filtering defogging of buoy images based on convolutional neural network. *Navig. China* **2020**, *43*, 84–88.
9. He, H.; Turghunjian, ·A.; He, X. A method of colored image defogging based on wavelet transform. *Comput. Technol. Dev.* **2020**, *30*, 60–64.
10. Wang, R.G.; Zhu, J.; Yang, W.T.; Fang, S.; Zhang, X.T. An improved local multi-scale Retinex algorithm based on illuminance image segmentation. *Acta Electron. Sin.* **2010**, *38*, 1181–1186.
11. Guo, L.; Song, J.; Li, X.; Huang, H.; Du, J.; Sheng, G. Single image haze removal using dark channel prior and adaptive transmission rate. In Proceedings of the 2018 2nd International Conference on Computer Science and Intelligent Communication(CSIC 2018), Xi'an, China, 16–17 August 2018.
12. Raanan, F. Dehazing using color-lines. *ACM Trans. Graph.* **2014**, *34*, 13.
13. Peng, L.; Jing, L.; Rui, Z. Research and implementation of single image fast dehazing algorithm. *China Comput. Commun.* **2019**, *31*, 43–44.
14. Ma, R.; Zhang, S. An improved color image defogging algorithm using dark channel model and enhancing saturation. *Opt. Int. J. Light Electron Opt.* **2019**, *180*, 997–1000. [CrossRef]
15. Zhang, X.; Ji, J.; Wang, L.; He, Z.; Liu, S. People's Fast Moving Detection Method in Buses Based on YOLOv5. *Int. J. Sens. Sens. Netw.* **2021**, *9*, 30.
16. Zhang, L.; Zhang, L.; Jiang, J.; Fang, W.; Liu, K. RealTime Detection and Tracking Method of Pilot's Head Position Based on MTCNN-DeepSORT. *J. Phys. Conf. Ser.* **2020**, *1682*, 012025. [CrossRef]
17. Qiu, X.; Sun, X.; Chen, Y.; Wang, X. *Pedestrian Detection and Counting Method based on YOLOv5+DeepSORT*; Tibet University: Lhasa, China, 2021.
18. Pu, H.; Li, Z.; Li, L. Research of image dehazing based on dark channel prior. *Laser Optoelectron. Prog.* **2021**, *58*, 175–181.
19. Qing, T.; Tongyang, Y.; Dan, Y.; Yun, W. A pedestrian detection method based on dark channel defogging and deep learning. *Laser Optoelectron. Prog.* **2018**, *634*, 187–195.
20. He, K.M.; Sun, J.; Tang, X. Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353.
21. Yin, J.; Qu, S.; Yao, Z.; Hu, X.; Qin, X.; Hua, P. Traffic sign recognition model in haze weather based on YOLOv5. *J. Comput. Appl.* **2022**, *42*, 2876.
22. YOLOV5 Object Detection-OSA-BottleneckCSP. Available online: https://zhuanlan.zhihu.com/p/462034609 (accessed on 18 March 2020).
23. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018.
24. Jin, L.; Hua, Q.; Guo, B.; Xie, X.; Yan, F.; Wu, B. Multi-target tracking of vehicles based on optimized DeepSort. *J. Zhejiang Univ. Eng. Sci.* **2021**, *55*, 1056–1064.
25. Song, S.; Zhan, Y.; Li, G. Hyperspectral Image Compression Employing Target Detection. *Comput. Technol. Dev.* **2009**, *19*, 1–3+158.
26. Sun, Z. *Research on Lossless Compression and Dehazing Algorithm in Front Precess System of Camera*; Shan Dong University: Jinan, China, 2018.
27. Huang, Y.; Liao, X.; Lu, Q. Infrared images interpolation algorithm combining bilinear interpolation and local mean. *Comput. Technol. Autom.* **2020**, *39*, 133–137. [CrossRef]
28. Ma, J.; Yao, Z.; Xu, C.; Chen, S. Multiple drones real-time tracking algorithm based on improved PP-YOLO and Deep-SORT. *J. Comput. Appl.* **2022**, *42*, 2885.