remote sensing

MDPI

Article

Toward More Robust and Real-Time Unmanned Aerial Vehicle Detection and Tracking via Cross-Scale Feature Aggregation Based on the Center Keypoint

Min Bao [1,*], Guyo Chala Urgessa [2], Mengdao Xing [3], Liang Han [4] and Rui Chen [5]

[1] School of Electronic Engineering, Xidian University, Xi'an 710071, China
[2] School of Telecommunication Engineering, Xidian University, Xi'an 710071, China; naurgessa@stu.xidian.edu.cn
[3] National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China; xmd@xidian.edu.cn
[4] School of Physics and Optoelectronic Engineering, Xidian University, Xi'an 710071, China; hlsdy@163.com
[5] State Key Laboratory of ISN, Xidian University, Xi'an 710071, China; rchen@xidian.edu.cn
* Correspondence: mbao@xidian.edu.cn

Abstract: Unmanned aerial vehicles (UAVs) play an essential role in various applications, such as transportation and intelligent environmental sensing. However, due to camera motion and complex environments, it can be difficult to recognize the UAV from its surroundings thus, traditional methods often miss detection of UAVs and generate false alarms. To address these issues, we propose a novel method for detecting and tracking UAVs. First, a cross-scale feature aggregation CenterNet (CFACN) is constructed to recognize the UAVs. CFACN is a free anchor-based center point estimation method that can effectively decrease the false alarm rate, the misdetection of small targets, and computational complexity. Secondly, the region of interest-scale-crop-resize (RSCR) method is utilized to merge CFACN and region-of-interest (ROI) CFACN (ROI-CFACN) further, in order to improve the accuracy at a lower computational cost. Finally, the Kalman filter is adopted to track the UAV. The effectiveness of our method is validated using a collected UAV dataset. The experimental results demonstrate that our methods can achieve higher accuracy with lower computational cost, being superior to BiFPN, CenterNet, YoLo, and their variants on the same dataset.

Keywords: cross-scale feature aggregation; center point estimation; region of interest; unmanned aerial vehicle

check for updates

Citation: Bao, M.; Chala Urgessa, G.; Xing, M.; Han, L.; Chen, R. Toward More Robust and Real-Time Unmanned Aerial Vehicle Detection and Tracking via Cross-Scale Feature Aggregation Based on the Center Keypoint. *Remote Sens.* **2021**, *13*, 1416. https://doi.org/10.3390/rs13081416

Academic Editor: Filiberto Pla

Received: 2 February 2021
Accepted: 4 April 2021
Published: 7 April 2021

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

Copyright: © 2021 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

Drones, also called unmanned aerial vehicles (UAVs), have become increasingly popular for both military purposes and domestic uses. However, due to their agility, accessibility, and low cost, drones can fly above prohibited areas and as such pose significant security risks. For example, remote-controlled drones have repeatedly violated the boundaries of protected areas such as airports and military bases. Additionally, they may also be used to engaged in illegal activities, such as invasion of privacy, smuggling, and industrial espionage [1,2]. Thus, there is a strong need to develop methods to detect drones and defend against them autonomously.

Many techonolgy have been utilized to deal with this problem, including those based on acoustic [3,4], lidar [5], radar [6], RF signal detection, and optical camera [7–11] sensors. Acoustic sensors [6,12], installed on microphone arrays, can be employed to detect the specific sound of drone rotors. However, this may not work in noisy environments, such as airports. Meanwhile, drone surveillance using lidar is still doubtful, due to the cost-effectiveness, massive data output, cloud sensitivity, and so on. Although, radar has been used for several decades to detect flying vehicles there are still some difficulties in detecting drones using radars when a drone's electromagnetic signature is low and its traveling velocity is slower [13,14]. By intercepting the communications between drones and ground

operators [14,15], RF-based detection has widely been used for jamming drone movement in the commercial market. However, this approach may fail when the drone navigates using a pre-programmed flight path, which does not require ground operators. With the improvement of deep learning—specifically deep convolutional neural network (CNN) algorithms—tracking with optical cameras has become a leading approach in the detection of drones, as it has presented several advantages, such as higher robustness and accuracy, larger ranges, and better interpretability [10].

AlexNet [16] first demonstrated the robustness and accuracy of deep CNN for object detection. Since then, more and more end-to-end CNN-based network models have been proposed for the detection task. Mainly because the designed features do not require a manual design, ultimately increases the generalization capacity of the model [17]. In the recent market, two popular CNN-based network architectures used for object detection exist. Among them, the two-stage [18–21] detector has better accuracy of detection over the single stage-detector [22,23] however, it is not ideal for real-time object detection due to the high computational cost required. In contrast, to improve the accuracy while preserving the low computational cost of the one-stage detector, several improvements have been made to the one-stage detector network, including the anchor shape prior box [23], the loss of class imbalance [24], the cascade of different feature resolution layers [25], and the feature pyramid network (FPN) [26].

A deep CNN can discover more critical features of the object in order to better detect and track targets obtained from optical cameras, compared with other machine learning algorithms [27]. However, the detection and tracking of UAVs for surveillance using a pan-tilt-zoom (PTZ) camera is challenging, compared to a static camera, due to the camera motion and the training parameters of the CNN model. For the standard object detection models to detect and track a target, the model may misdetect and generate a false alarm, or fail to track a target due to the computational complexity and response time of the PTZ motor.

To detect and track a UAV using a PTZ camera by overcoming the challenge mentioned above, it needs to design a low computational complexity algorithm. Many researchers have proposed drone detection methods using single-stage deep CNNs. As an example, the airborne visual detection and tracking of cooperative UAVs by exploiting a deep CNN has been proposed by [11]. In the proposed method, each frame is divided into a square grid, and each square grid is used as an input to a light-weight YoLo [23] network, in order to detect and track the drone. A deep learning-based strategy to detect and track drones using several cameras (DTDUSC) has also been proposed by [10]. In both [10,11] methods, light-weight YoLo is used as the backbone of the network model, taking advantage of its low computational cost. However, the detector may miss targets, mainly due to camera motion and the target being below the horizon. Besides, both network models struggle to detect small drones and may fail to generalize to drones with new or unusual aspect ratios or configurations. One of the main reasons for such a struggle is the utilization of an anchor box, due to the constraints of the bounding box's minimum and maximum size. Thus, utilizing anchors to detect a target reduces the detection accuracy, due to the size invariance of objects in different ranges, and increases the computational complexity, because of the post-processing necessity to filter the final bounding box from the candidate bounding boxes.

Researchers have also recently focused on anchor-free methods, such as CornerNet [28] and ExtremeNet [29]. CornerNet detects the upper-left and lower-right corners of the bounding box, in order to determine the target's position and size. However, the semantic information of these corners is comparatively weak and difficult to detect. Besides, in this method, post-processing is required to combine two points that belong to the same target. ExtremeNet uses the best key point estimation framework to find the extreme points by predicting the target center. However, its computational cost is high and its detection accuracy is not significantly improved, due to the higher number of points involved. In addition, the network pays too much attention to edges, which can

easily cause misdetection and false alarms. To address these issues, ref. [30] introduced CenterNet, which only needs to extract the center point of each object, without requiring post-processing, effectively reducing the false alarm detection and computational cost.

In addition to UAV detection, many researchers have also recently proposed tracking algorithms using traditional or machine learning-based methods, which can be integrated with a detection algorithm. Simple online real-time tracking (SORT) is proposed in [31], which is a simple and effective way of tracking multiple detected objects using the Kalman filter [32]. However, the contextual information of consecutive video frames was given little attention and the method was designed for online detection-tracking. Furthermore, the model fails to predict the target's next state, due to occlusions, different viewpoints, and so on. Many researchers have proposed methods to improve the tracking accuracy by adding or using deep learning approaches to curb this problem. Alternatively, ref. [33] used long-short term memory (LSTM) and proposed a real-time recurrent regression network for the visual tracking of generic objects (Re^3). This method showed promising results, in terms of the tracking approach, for real-time application. However, the tracking result was not suitable for small targets in a cluttered region, generated false alarms, and failed under rapid movement. Comparing SORT with Re^3, SORT is fast, straightforward, outperforms in predicting the future target state, and it performs better with small-sized targets under Gaussian noise.

Overall, as shown in Figure 1, to effectively detect and track a drone in real-time using a PTZ camera, we propose a novel CNN-based UAV detection algorithm integrated with a Kalman filter-based tracking algorithm. The detection method was designed with a small number of training parameters. Similarly, the tracking was designed by using CNNs with the Kalman filter, in order to reduce the computational cost and enlarge the target region within a frame, thus improving the detection accuracy for small targets.

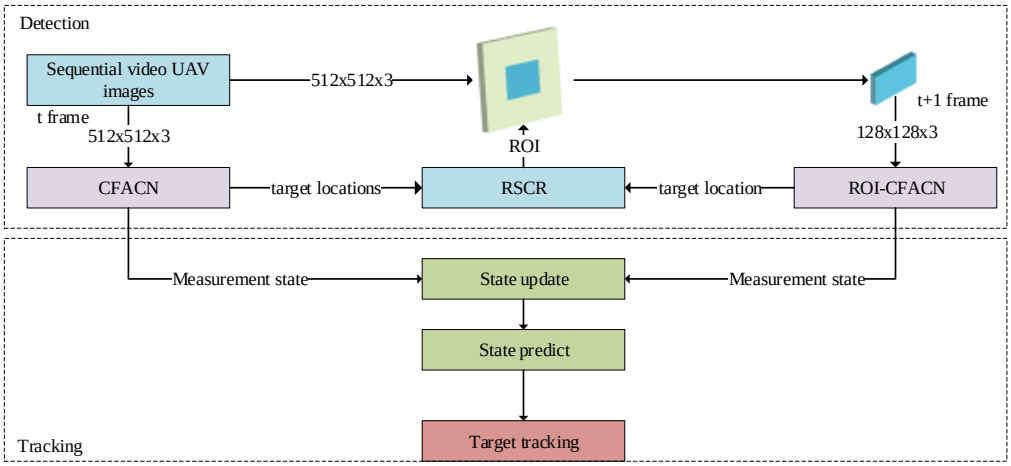
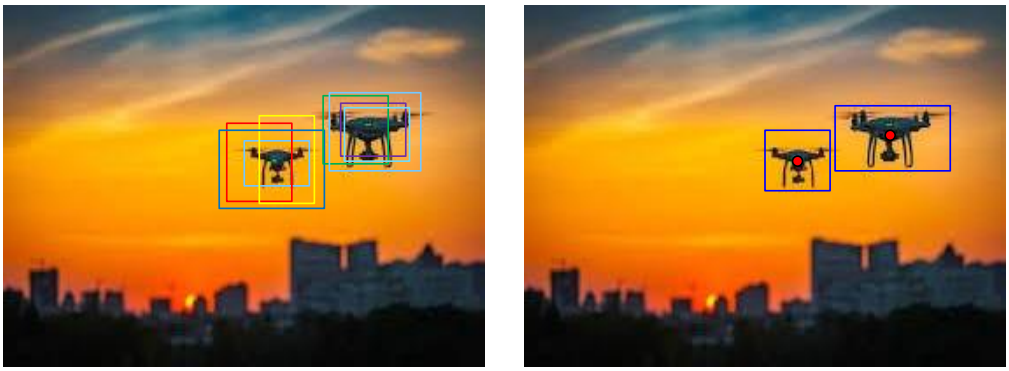


Figure 1. Flowchart of the proposed unmanned aerial vehicle (UAV) detection and tracking method. It consists of two parts: Detection and tracking.

In the detection step, the key point estimation of the modified CenterNet [30] and the network feature aggregation of FPN [26] motivated us to design a new CNN architecture called cross-scale feature aggregation CenterNet (CFACN), in order to obtain better performance with a small number of training parameters. We used the CenterNet key-point estimation approach and modified its key point estimation to use an anchor-free method, thus improving the detection of small targets and reducing the computational complexity. Figure 2 shows a comparison of anchor-based and key point-based detection approaches. In Figure 2a, the anchor boxes are generated to obtain a bounding box for the target, whereas Figure 2b shows the center key point-based method, using one key point for one object. Furthermore, in the tracking stage, we design a lightweight detector, region-of-interest

cross-scale feature aggregation CenterNet (ROI-CFACN), which is used as a tracker with the Kalman filter, in order to improve detection and training accuracy.



(a) (b)

Figure 2. (**a**) Standard anchor-based detection. (**b**) Center point-based detection.

The proposed methods use the CFACN and ROI-CFACN with the Kalman filter, in order to detect and track UAVs from video data. Firstly, the CFACN or ROI-CFACN networks are used to extract features for different pixel size frames at different frames, to recognize and obtain the bounding boxes of UAVs from a video frame. Then, the bounding boxes pass through the region-of-interest-scale-crop-resize (RSCR) method to get a ROI for the next frame. Finally, according to the result obtained from either CFACN or ROI-CFACN in the video frames, we use the Kalman filter algorithm to predict or update the UAV state directly using a result from CNN detection. The main contributions of this paper are summarized as follows:

(1) The novel CFACN and ROI-CFACN methods are proposed to detect UAVs from video data. These methods use cross-scale feature aggregation (CFA) to effectively estimate the center key point, size, and regression offset of the UAVs. In the estimation, CFA uses bi-directional information flows between different layers of features in up–down directions, in order to improve accuracy and efficiency. Furthermore, CFA helps the network to learn the effect of the up layer on the down layer (and vice versa), due to its feedback flow;

(2) The RSCR algorithm, which uses the contextual information of consecutive frames, is designed to merge the CFACN and ROI-CFACN methods. This algorithm helps to flow the information between the two proposed networks. It also helps the ROI-CFACN to focus on the ROI by removing the background effect and enlarging the small target. This algorithm not only improves the accuracy, but also further reduces the computational complexity of the method;

(3) A dynamic state estimation approach is designed to track the UAV. The dynamic state uses eight state-based vectors for target state estimation and tracks the drone by using either a simple detection-based online tracking, using the result obtained from the CNN, or a tracking-based detection approach, using the Kalman-estimated state.

The remainder of this paper is organized as follows: Section 2 introduces the proposed method in detail. In Section 3, the proposed method is evaluated. In Section 4, the discussion is presented. Finally, in Section 5, our conclusions are provided.

2. Proposed Method

For UAV detection applications using a PTZ camera, the overall latency can be obtained by assessing three stages (i.e., detection, tracking, and PTZ controller). The controller section is beyond the scope of this article. The remaining stages—detection and tracking—discussed in this article is designed to work in real-time applications and implemented

on the parallel algorithm concept, where each stage works separately and transfers data in between.

As shown in Figure 1, the overall proposed method mainly consists of CFACN, RSCR, ROI-CFACN, and dynamic state estimation to detect and track UAVs. In the detection part, CFAN is used to obtain the bounding boxes of a target in the previous frame. The final candidate bounding box is selected according to its size or the bounding box prediction score, and is used to initialize the tracker and set it as the RSRC input value. In the current frame, the input frame, of dimensions 512 × 512 pixels, is cropped to 128 × 128 pixels, according to the ROI obtained from the RSCR, and used in ROI-CFACN to predict the target bounding box, thus minimizing the target search region and enlarging small targets. The final candidate bounding box obtained from each frame is saved as the measured state. In the tracking part, a Bayesian approach using the Kalman filter algorithm is employed to predict the drone's state in the current frame and sends the necessary data to the controller, in order to control the rotating turret's direction and speed. Details about the CFACN, ROI-CFACN, RSCR, and dynamic state estimation methods are discussed in the following.

2.1. CFACN and ROI-CFACN Architecture

The CFACN uses 512 × 512 pixel frame as an input, in order to identify and localize the drone and propose candidate ROIs. ROI-CFACN, the second network, uses the cropped 128 × 128 pixel frame as input, in order to localize the drones with lower pixel scale by reducing the background complexity. This strategy allows us not only to reduce the computational cost, but also limits the number of false alarms, which tend to occur when the target search is carried out across the entire image plane. To address this, we designed a CFA method on the basis of ResNet-18. The overall proposed detector method is key point-based and built on top of residual blocks, with CFA as the building block of the network.

In the proposed method, a consecutive frame from a video is taken as input to a full convolutional CNN to generate a heatmap (to obtain the center point), size (height and width of the bounding box), and offset, as shown in Figure 3. First, assume $I \in \mathbb{R}^{W \times H \times 3}$ is an input frame obtained from the video, with width W and height H, from which we expect to predict $\hat{Y} \in [0,1]^{\frac{W}{S} \times \frac{H}{S} \times C}$, where S is output stride size and C is the number of key point categories. In the generated key point, $\hat{Y} = 1$. It is considered as a detected key point; if $\hat{Y} = 0$, it is treated as background. If the (x_1, y_1, x_2, y_2) is the ground truth bounding box of the target, its center point is set to 1 at $p = \left\lceil \widetilde{p} \right\rceil$ where $\widetilde{p} = \left(\frac{((x_1+x_2))/s}{2}, \frac{((y_1+y_2))/s}{2} \right)$ and its offset point at p is $\lambda = \widetilde{p} - p$. The other information about the target is obtained from the key point, stride, and image information. All the center point is obtained from the predicted $\hat{Y}$ and then regressed to obtain the target bounding box size. The value at each key point is used as the confidence score, and regression at its position is used to obtain the bounding box size. The position coordinates are then calculated as:

$$\begin{pmatrix} \hat{x}_i + \lambda\hat{x}_i - \frac{\hat{w}_i}{2}, \hat{y}_i + \lambda\hat{y}_i - \frac{\hat{h}_i}{2} \\ \hat{x}_i + \lambda\hat{x}_i + \frac{\hat{w}_i}{2}, \hat{y}_i + \lambda\hat{y}_i + \frac{\hat{h}_i}{2} \end{pmatrix} \tag{1}$$

where $\left(\lambda\hat{x}_i, \lambda\hat{y}_i\right) = \hat{O}_{\hat{x}_i, \hat{y}_i}$ is the offset prediction and $\left(\hat{w}_i, \hat{h}_i\right) = \hat{B}_{x_i, y_i}$ is the predicted target size.

CenterNet-based models only detect the center point and obtain minimum target information, resulting in more misdetections and false alarms for UAV targets. We designed a modification of the key point estimation step of CenterNet and a new CNN architecture, called CFA, in order to extract more important information through its feedback connection, to solve this problem.



Figure 3. Outputs layer of cross-scale feature aggregation CenterNet (CFACN) and region-of-interest (ROI)-CFACN.

Cross-Scale Feature Aggregation

Conventional multi-scale feature aggregation aims to aggregate different feature layers. Formally, after a feature is extracted using the backbone network, given a list of multi-scale features $P^{in} = (P_{l1}^{in}, P_{l2}^{in}, P_{l3}^{in}, P_{l4}^{in}, P_{l5}^{in})$, as shown in Figure 4a, where P_{l1}^{in} represents the feature output at level l_i, our objective is to find the transformation f that can effectively aggregate different features and output a list of new features: $P^{out} = f(P^{in})$. As an example, Figure 4b shows the conventional top-down FPN approach [26]. The convolutional feature levels from layers 3–7 are taken as input features $P^{in} = (P_3^{in}, P_4^{in}, P_5^{in}, P_5^{in}, P_6^{in}, P_7^{in})$, where P_i^{in} represents a feature level with resolution of $1/2^i$ in the input image.

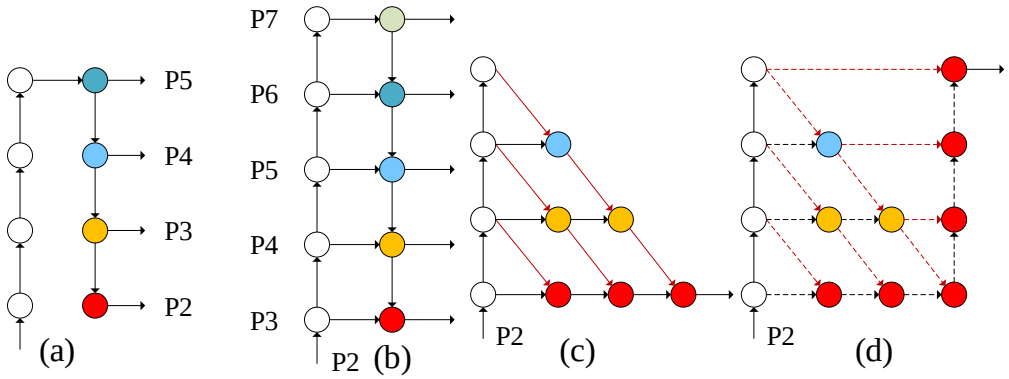


Figure 4. Design of different feature networks. (**a**) ResNet with transpose convolutions as a backbone. (**b**) Feature pyramid network (FPN). (**c**) Deep layer aggregation (DLA) network and (**d**) CenterNet, wich is a modified DLA with more skip connection.

For instance, if the input resolution is 512 × 512, then P_3^{in} represents the feature level 3 $\left(512/2^3 = 64\right)$ with a resolution of 64 × 64. FPN aggregates multi-scale features in a top-down manner:

$$P_7^{out} = Conv(P_7^{in})$$
$$P_6^{out} = Conv(P_6^{in} + Resize(P_7^{out}))$$
$$\ldots$$
$$P_3^{out} = Conv(P_3^{in} + Resize(P_4^{out}))$$

where Resize is usually an upsampling or downsampling operation for feature matching and Conv is usually a convolutional operation for feature processing. This kind of conventional method generally suppresses the accuracy of the results, due to one-way information flow between the layers. In addition, as shown in Figure 4c,d, deep multi-scale feature aggregation (MFA) has been proposed to improve the result. However, this network also requires more parameters, thus making its computational complexity high. As a concrete example, CenterNet uses different backbones to compare their accuracy and efficiency

trade-offs, from which it has been shown that the deep MFA has better accuracy, but reduced efficiency. In order to improve both the accuracy and efficiency under limited computational resources, we propose a novel MFA- and CFA-based network, as shown in Figure 5.

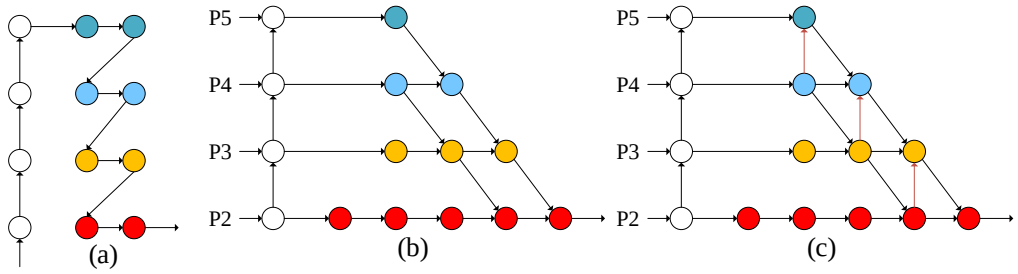


Figure 5. Design of proposed feature networks. (**a**) CenterNet network. (**b**) The proposed multi-scale feature aggregation (MFA) network. (**c**) The proposed CFA network with better accuracy and efficiency trade-offs.

As shown in Figure 5a, in CenterNet, the network is designed to have a single information flow through the network. In order to improve the accuracy, multi-scale feature aggregation CenterNet (MFACN) was designed using the MFA parallel information flow network, as shown in Figure 5b. To further improve MFACN, the CFA bi-directional parallel flow network with a feedback architecture was designed, as shown in Figure 5c.

ResNet [34], shown in Figure 4a, is used to extract the features in different layers as a backbone. Then, we use the ResNet outputs from layers 2 to 5 (i.e., P2, P3, P4, and P5) as inputs to CFA. In the design of a CFA, we first minimize the up–down nodes. Our target is simple: Minimize the computational cost and improve the detection accuracy, by minimizing the up-node aggregation between different layers and connected in cross-triangular form. This led us to design a simple network. Secondly, we increase the down-node aggregation by connecting the last intermediate (P4, P3, and P2) layers with the last convolution layer of each stage (P5, P4, and P3, respectively) in the CFA, as shown by the red connections in Figure 5c, in order to improve the detection accuracy. Furthermore, to obtain an improved detector through the down–up nodes, using downsampling or upsampling layers (e.g., a pooling or upsampling layer) is not effective when the network layer is small. To address this problem, we use a convolutional layer with a stride of two as the downsampling layer and a transposed convolution layer as the upsampling layer, in order to better understand the correlation between the different sizes of output feature layers. The CFA is obtained as:

$$
\begin{aligned}
{P_5}^{out} &= \sigma(Conv(P_5) + TransConv({P_4}^{td})) \\
{P_4}^{out} &= \sigma(\sigma(Conv({P_4}^{td} + TransConv({P_5}^{out})) + ConvSt({P_3}^{td})) \\
{P_3}^{out} &= \sigma(\sigma(Conv({P_3}^{td} + TransConv({P_4}^{out})) + ConvSt({P_2}^{td})) \\
{P_2}^{out} &= \sigma(Conv({P_2}^{td} + TransConv({P_3}^{out})))
\end{aligned}
$$

where σ is the ReLu function, ConvSt is a convolution and downsampling operation with stride 2 for feature processing and feature matching, TransConv is a transposed convolution and upsampling operation for feature processing and feature matching, and ${P_4}^{td}$ is the intermediate feature at level 4 on the top-down pathway, which shows the information flow of different features in an up–down direction.

The CFA is simple but effective, allowing for a higher-resolution output (stride 4). At the same time, we change the channels of the four transposed convolution layers of CFA to 512, 256, 128, and 64, respectively. The up-convolutional filters are initialized using the normal Gaussian distribution with mean zero.

As shown in Figure 6, the base detector CFACN mainly consists of three parts: ResNet-18 as its backbone, CFA, and the prediction header network. Both CFACN and ROI-CFACN

frameworks include 3 × 3 convolution, 1 × 1 convolution, Keypoint heatmap prediction header, regression prediction header, and size prediction header.

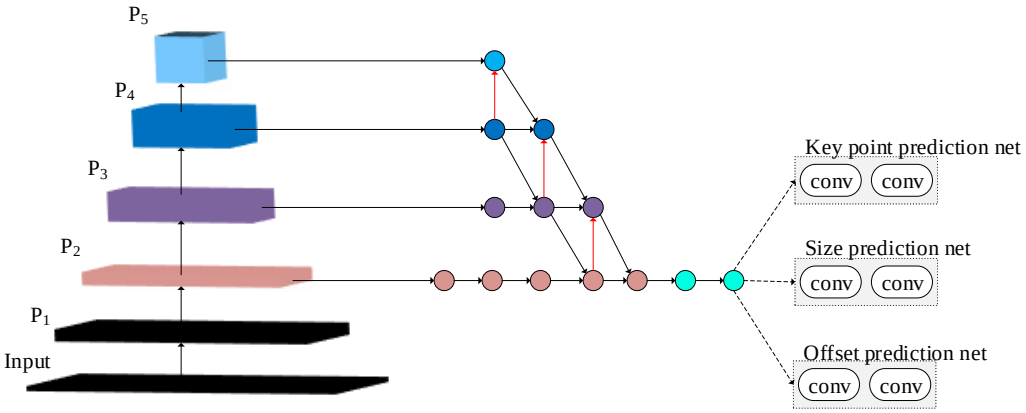


Figure 6. The CFACN and ROI-CFACN overall architecture.

ROI-CFACN is the lightweight version of CFACN; the difference is that the channel depth of the CFA layer in the aggregation network is 256, 128, 64, and 32, respectively. The backbone in both scenarios has the same layer and channel depth. As shown in Figure 6, for each of the frameworks, the features of the backbones are passed through the CFA, then through an isolated 3 × 3 convolution, ReLu, and another 1 × 1 convolution. In total, CFACN and ROI-CFACN have a backbone, cross-scale aggregation, two 3 × 3 convolutional layers, one 1 × 1 convolutional layer, and three prediction layers.

In addition to the advantages of CFA—that is, key point estimation and post-processing without non-maximum suppression (NMS)—we also need to consider the contextual information of consecutive frames in the videos. Utilizing the contextual information of detected UAVs in CFACN and RSCR allows us to obtain the ROI of the next frame for ROI-CFACN. This further reduces the effect of background complexity and enlarges the UAV area to improve the accuracy and reduce the computational complexity. Then, ROI-CFACN is used to detect the UAV within the ROI obtained from the RSCR algorithm. However, for the ROI-CFACN to be effective, the base detector plays a significant role. If the accuracy of the base detector is low, ROI-CFACN cannot be used, and the method is not adequate. On the other hand, when the detection accuracy of the base detector is high, the method is more effective.

2.2. ROI Formulation

The final candidate bounding boxes obtained in frame t from the detection result of CFACN 512 × 512 pixel size is passed through RSCR and used as a ROI in frame $t + 1$. However, when using ROI in ROI-CFACN, we have two major issues. First, the target location in frame t does not always overlap the target in frame $t + 1$; second, the network input of ROI-CFACN has a fixed size. To solve these issues, we consider the maximum distance the drone could jump from between frames t and $t + 1$, thus widening our search region area. To widen the search region and avoid the error of the prediction bounding box mismatching the ground truth, the final candidate bounding box is first scaled by a constant η and set as an ROI. In our experiment, η was set to 2. Furthermore, the image is cropped and reshaped to a 128 × 128 pixel size. Finally, the reshaped image is used as input for the ROI-CFACN.

2.3. Tracking on the Current Image Plane

The goal of the tracker is to associate localized objects across a sequence of video frames. The tracking algorithm obtains the measurement state, from either CFACN or ROI-CFACN, in order to estimate the motion and related data. In our work, we modified the dynamic model of SORT, based on the Kalman filter algorithm, to find the optimized state

estimate from the input state, previous state estimate, and mathematical models. SORT works best for linear systems with Gaussian processes involved. However, SORT may fail in many applications, due to occlusions, different viewpoints, and so on. To improve this, the Kalman filter associated with lightweight deep learning and ROI was used. When the state scenario has a Gaussian distribution, the prediction state is used to find ROI otherwise, a deep learning framework is utilized to recover the point of view. Overall, in the tracking algorithm, the Bayesian optimization method—which returns more accurate results than just one single observed measurement—is used first. Whenever a new measurement comes in, the filter updates the new estimate, such that the error estimation vector between the estimated states and the measured state is minimized. The recursive manner of state measuring, prediction, and updating make it a useful method under estimation accuracy and time constraints, as highly required for the considered application.

Secondly, to avoid the aspect ratio of the target bounding box being constant as in SORT. In the dynamic model, the state of a drone is defined in an eight-state space, as shown in Equation (2):

$$s_v = \left[c_x, c_y, w, h, \dot{c_x}, \dot{c_y}, \dot{w}, \dot{h} \right]^T \tag{2}$$

where c_x, c_y represents the center of each target bounding box; w, h represents the width and height of each target bounding box; and $\dot{c_x}, \dot{c_y}, \dot{w}, \dot{h}$ represent their respective velocities. When either of the detectors detects the target location and is associated with a target, the velocity component is solved optimally by a Kalman filter framework and the detection bounding box is used to update the target state and the turret rotation is adjusted accordingly (i.e., online detection-tracking). If no detection is associated with the target, its state is simply predicted (without correction) using the linear velocity model (i.e., tracking-detection).

Finally, to avoid an increase of the computational cost when the detection is not associated with a target, a recovery time is utilized for the ROI-CFACN, to detect a target within five consecutive frames in the tracking-detection state. During the recovery time, if the ROI-CFACN does not obtain the target bounding box, the whole algorithm is reset and the CFACN method is re-adopted.

Our approach is quite general, as it can handle a variety of UAV detection situations [35,36]. Empirical performance evaluations established the advantages of the proposed method over other state-of-the-art algorithms, as detailed below.

3. Result

3.1. Experimental Setup

We evaluated our proposed method, in terms of its robustness, effectiveness, and efficiency. The proposed method was implemented in Python 3.7.3 with OpenCV 4.0.1 and TensorFlow, on a computer with an Intel(R) Core(TM) i5-8400 CPU, 16 GB of RAM, 2.8 GHz processor speed, and an 8-GB NVidia 1080 GPU. To evaluate the performance of the proposed method, we collected a data set, comprised of six videos and 1000 selected images of drones from Google Images [37]. From the collected 4700 images, for the CFACN, we randomly selected and labeled 3427 images and for the ROI-CFACN, 2450 randomly selected images were labeled. Then, the data set was organized into three sets, as shown in Table 1.

Table 1. The total number of collected datasets.

Frameworks	Training Set	Validation Set	Test Set
CFACN	2057 images	685 images	685 images
ROI-CFACN	1340 images	555 images	555 images

The data set was labeled using the free AI source code tool LabelImg. As shown in Figure 7, the labeled data set was exported in PASCAL Visual Object Classes (VOC) standard format and confirmed by professional interpretation at Xidian University.

```
<object>
    <name>drone</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>100</xmin>
        <ymin>139</ymin>
        <xmax>401</xmax>
        <ymax>302</ymax>
    </bndbox>
</object>
```

Figure 7. Dataset annotation example.

As shown in Figure 8, when collecting the data set, we considered the challenges that may occur, such as illumination, shadow inferences, background inferences, scale variation, occlusion, in-plane, out-plane, and rotation.

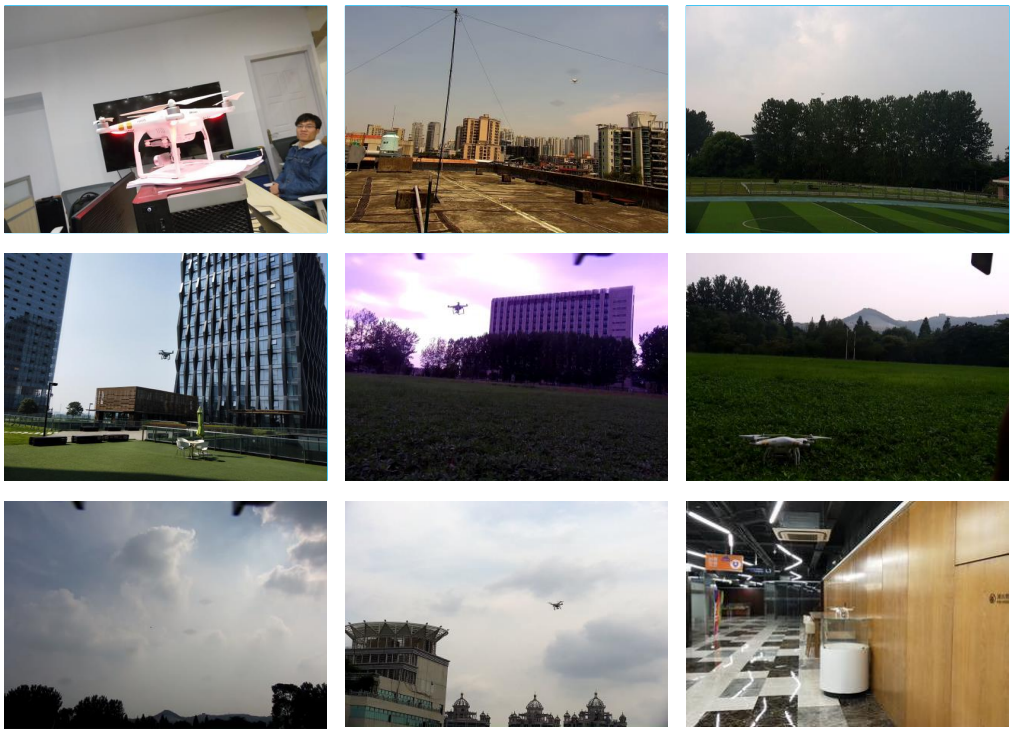


Figure 8. Collected dataset to train and evaluate a network.

Random scale, flip, crop, enhancement, and rotation operations were utilized for data augmentation during training. Each model was trained using the Adam optimizer. The initial learning rate was set to 0.001 and linearly annealed down (using the cosine decay rule) when the training epoch did not decrease for two consecutive epochs. The backbone of the CFACN detection network used ResNet-18. The loss functions used were focal loss [24] (with $\alpha = 2$ and $\beta = 4$) for key point estimation and L1 loss for both size and local offset estimation. During training, each network model trained for 100 epochs with a total batch size of 32.

During inference, the camera and rotating turret were mounted in an outdoor case at the top of the building. In the tracking algorithm, the transition state matrix $H \in \mathbb{R}^{8\times8}$ and measurement matrix $C \in \mathbb{R}^{4\times8}$ were fixed during the experiment. The diagonal values, corresponding to the position (i.e., (x, y)) and size (i.e., w, h)) covariance in the prediction noise covariance matrix Q and measurement covariance matrix $\Re$, respectively, were set to 0.01 and 0.1.

3.2. Performance Metrics

For the evaluation, we used key evaluation metrics, including average precision (AP), precision–recall curve to determine how many drones were detected correctly (true positive), and how many false positives were generated using the intersection over union (IoU) parameter. For each predicted bounding box and ground truth bounding box, the IoU score was evaluated and compared to the fixed threshold score to categorize the predicted bounding box into true positive (TP) or false positive (FP). The precision–recall curve for the methods was evaluated using Equations (3) and (4):

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{All detections}} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\text{All ground truths}}. \tag{4}$$

AP is defined as:

$$\text{AP} = \int_0^1 p(r)dr \tag{5}$$

where p represents precision and r represents recall, in which p is a function that takes r as a parameter, which is equal to taking the area under the curve. The simulation results in different settings at the time of training and inference were used for comparison, in order to evaluate the precision and efficiency of the proposed methods.

3.3. Experimental Evaluation

To show the effectiveness of the CFA, we compared the CFACN with MFACN and CenterNet. Furthermore, the proposed method was tested in the field under different challenging circumstances and we examined the proposed method's detection result, compared to other conventional drone detection methods using the same data set at training and evaluation. We used training and validation data sets at the training time and test data sets at inference time.

In Figure 9, Figure 9a shows the original image, Figure 9b shows the ground truth, and Figure 9c,d represent the test results for MFACN and CFACN, respectively. In Figure 9b, the red rectangle represents the real targeted drone location, while those in Figure 9c,d represent the detection results. The detection result in Figure 9b was used as the reference and, from Figure 9c,d, we can see that the MFACN had good detection results, but suffered from misdetection and false alarms. After the addition of the cross-connection to the network architecture (i.e., in CFACN), the misdetection and detection of the false positives decreased, as shown in Figure 9d, and the detection accuracy increased.

MFACN, CFACN, and CenterNet were built on top of ResNet and, so, use critical point estimation to learn the target's center point. To further show the model's efficiency, we compared the models with CenterNet models built on top of different ResNet backbone architectures.

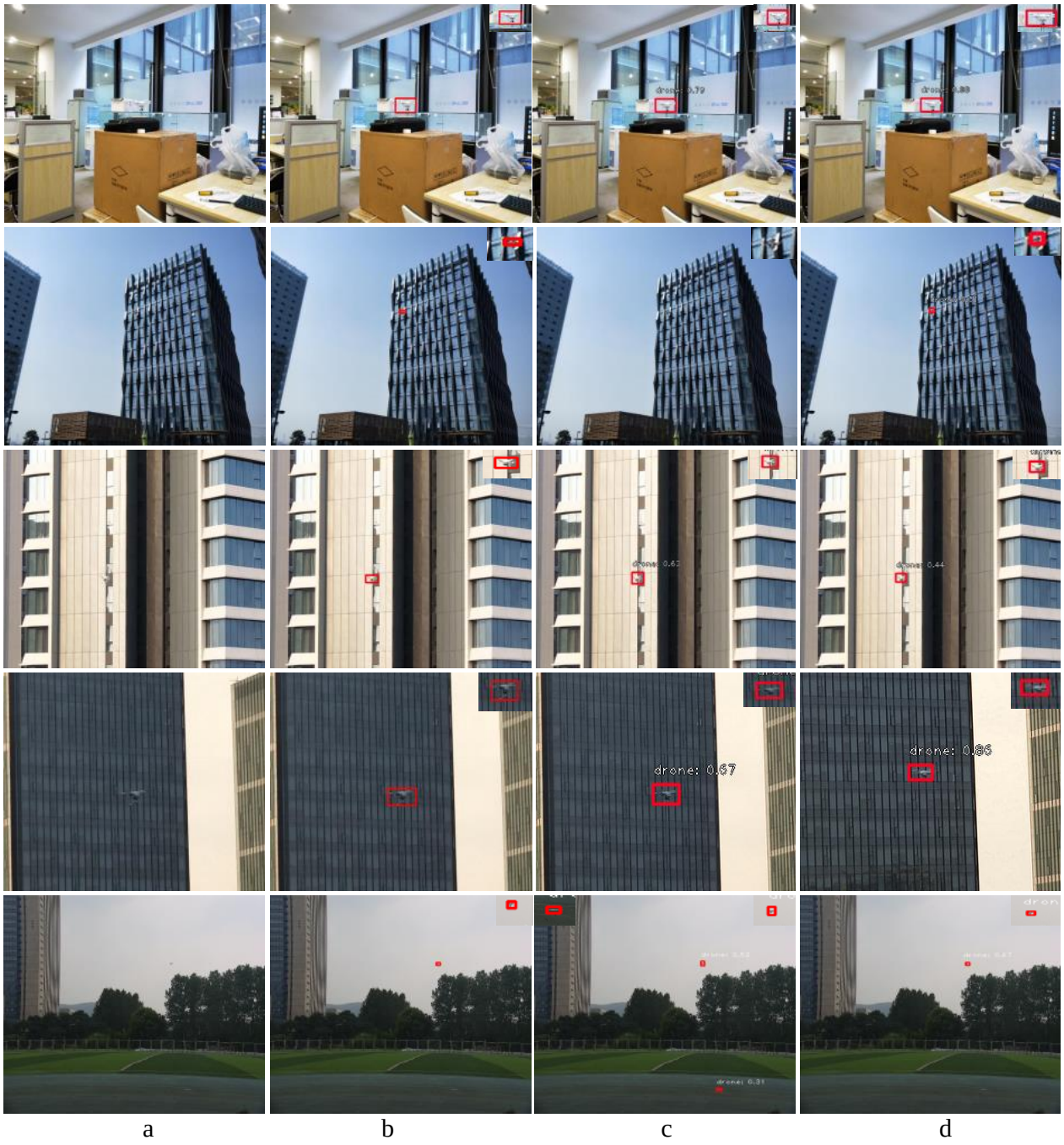


Figure 9. Detection results: (**a**) Original images; (**b**) ground truth; (**c**) detection result of MFACN; and (**d**) detection result of CFACN.

Figure 10 shows a comparison of CenterNet built on top of a different backbone with MFACN and CFACN. Figure 10a shows the ground truth, which was used as reference. Figure 10b shows that the result of CenterNet built on top of ResNet-18 suffered from false alarms and misdetection, in the cluttered region below the horizon, when the target size is small and the camera was in motion. Figure 10c shows the result of CenterNet utilizing ResNet-34 as its backbone network. The result also shows false alarms, low confidence, and misdetection after adding layers. Figure 10d shows the results of MFACN, demonstrating that the proposed method improves the detection accuracy over CenterNet built on different backbones, which had more training parameters. However, the method still showed false alarms and misdetection. Figure 10e shows the result of the proposed detector

network, which reduced the false alarm rate, effectively detected the small target, and had better detection accuracy without additional processing. Futhermore, the computational cost was reduced, compared to CenterNet built on top of ResNet-34.

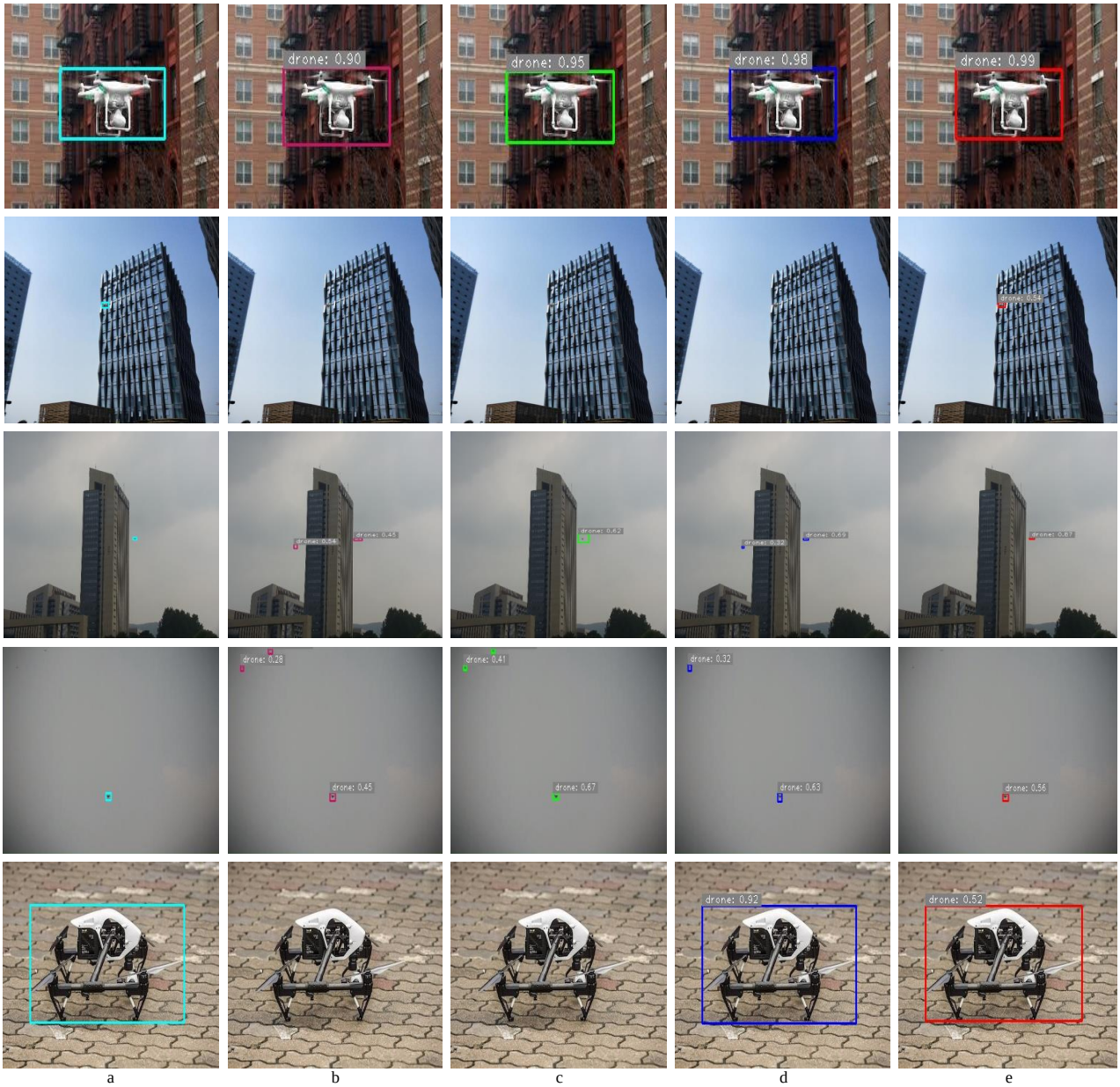


Figure 10. Comparison of the different detectors: (**a**) Ground truth; (**b**) detection result of CenterNet with ResNet-18 as its backbone; (**c**) detection result of CenterNet with ResNet-34 as its backbone; (**d**) detection result of MFACN with ResNet-18 as its backbone; and (**e**) detection result of the proposed method, CFACN, with ResNet-18 as its backbone.

As shown from the average precision result in Table 2, the proposed method shows the detection results in different IoU = [0.50, 0.60, 0.70, 0.80, 0.90], where AP_{50} is AP at IoU = 0.50 (PASCAL VOC metric). The proposed method's precision-recall result shows its superiority compared to the CenterNet and MFACN. Overall, the experiment result of the proposed methods effectively reduces false alarms and increases detection accuracy and confidence score compared to the CenterNet built on top of different backbones.

Table 2. The detection performance of CenterNet using ResNet-18 or ResNet-34 as its backbone, the proposed MFACN method using ResNet-18 as its backbone, and the proposed CFACN method using ResNet-18 as its backbone, in terms of different intersection over union (IoU) values.

Method	AP_{50} (%)	AP_{60} (%)	AP_{70} (%)	AP_{80} (%)	AP_{90} (%)
CenterNet [ResNet-18]	84.2	76.3	64.3	53.6	30.0
CenterNet [ResNet-34]	93.9	88.1	80.1	67.1	34.1
MFACN	95.0	90.6	83.0	66.8	34.0
CFACN	96.2	91.9	84.0	67.9	35.7

To show the effectiveness of CFA over MFA, we also evaluated the two methods using specific object sizes (i.e., small, medium, and large). The small-sized target was categorized as objects with size less than 12 × 12 pixels; medium-sized targets were within the range of 13 × 13 to 27 × 27 pixels; and a target above 27 × 27 pixels in size was categorized as a large-sized target.

As shown in Table 3, CFA was more effective in detecting small targets and had almost the same detection accuracy for a large-sized target. Overall, CFA was superior to MFA.

Table 3. The detection performance of MFACN and CFACN methods for different target sizes.

Method	AP_{small} (%)	AP_{medium} (%)	AP_{large} (%)
MFACN	72.76	94.46	98.38
CFACN	75.57	95.65	98.60

3.3.1. Comparison with Other Methods

To evaluate the performance of the proposed method, compared to anchor-based detectors, we compared the proposed CFACN method with the YoLov5 [38], BiFPN [39], and DTDUSC methods. The model accuracy and running time of the algorithms were compared, in order to determine the efficiency of the proposed model. Among the competing models, BiFPN is a scalable network that has seven different models. However, the proposed method was only compared to three (i.e., D0, D1, and D2) BiFPN models, ignoring the remaining BiFPN (D3–D7) models, due to their high computational complexity.

As shown in Table 4, the proposed method obtained the second-highest mean-average-precision (mAP) and the highest efficiency. BiFPN-(D0–D2) had false alarms and miss-detection when the background was cluttered. However, as the number of parameters increased from D0–D2, the accuracy improved and the model efficiency reduced. At the same time, all the models had drawbacks in the detection of a small target. The DTDUSC method was not effective under cluttered frames and complex backgrounds, and it was not cost-effective either. Even though it utilized additional computation for a small target, the results showed a high false alarm rate. YoLov5 obtained the highest mAP. However, its efficiency was the third highest, compared to the other four models, and the proposed model ran twice as fast. Overall, CFACN improved upon the detection accuracy of BiFPN-(D0-D2), DTDUSC, CenterNet, and MFACN, while obtaining an almost equal detection accuracy as YoLov5. However, the detection results showed misdetection under cluttered and complex backgrounds. To reduce such misdetection, the proposed method integrates CFACN with ROI-CFACN to further improve the detection accuracy and decrease the computational cost in applications where the computational cost is limited. In general, merging the two CNN models using contextual information reduced the misdetection of small targets by using the ROI to enlarge the target surroundings and, thus, further reducing the false alarm rate under cluttered and complex backgrounds.

Table 4. The mean average precision and computational time of four different detectors. DTDUSC: Deep learning-based strategy to detect and track drones using several cameras.

Method	Input Size	mAP (%)	Time (ms)
BiFPN-D0	512 × 512	83.65	51.63
BiFPN-D1	512 × 512	86.52	71.42
BiFPN-D2	512 × 512	88.98	83.33
DTDUSC	512 × 512	90.1	34.48
CFACN	512 × 512	96.2	23.80
YoLov5	512 × 512	96.9	50.00

3.3.2. Computational Performance

Conventionally, as a network becomes deeper, its detection accuracy improves however, only adding a deeper layer may not be an effective method. In this work, with only CFACN, we achieved an average speed of 43 frames per second (FPS) with a 96.2% mAP (as shown in Table 5) in the collected data set. Furthermore, when we combined the two deep learning frameworks of CFACN and ROI-CFACN, the accuracy and precision performance increased by 0.91% and the computational complexity was reduced.

The benefits obtained from ROI-CFACN were reducing the computational cost (by reducing the search region) and improving small targets' detection (by enlarging the selected ROI).

Table 5. Overall detection accuracy and frames per second (FPS) results of the proposed methods without tracking algorithm.

Methods	Input Size	mAP%	FPS
CFACN	512 × 512	96.2	43
ROI-CFACN	128 × 128	82.2	72
Proposed detection method [CFACN + ROI-CFACN]	512 × 512	97.11	52

To further show the advantage of the tracking algorithm with respect to accuracy and computational cost, we compared the overall proposed method with a different tracking algorithm and without it. The overall proposed method includes localization and predicting the target location using the Kalman filter and achieved 56 FPS with a 98.3% mAP in the collected data set.

As shown in Table 6, the merging CFACN with the ROI-CFACN improved the detection accuracy by 0.91% mAP, and the computational times reduced by 4.02 ms. This indicates that merging the lightweight using the ROI improves both accuracy and efficiency. To further improve the accuracy and reduce the computational complexity, we incorporate the CFACN, ROI-CFACN, and tracking algorithm. The combination of the ROI-CFACN and SORT tracking algorithm with the CFACN improved the CFACN detection accuracy by 1.54% mAP and computational time reduced by 4.74 ms. With the proposed tracking algorithm, the detection accuracy is enhanced by 2.1% mAP, and computational time is reduced by 5.4 ms. Overall, the combination of the proposed tracking algorithm with ROI-CFACN can achieve an additional 2.1% mAP accuracy in a low computational cost comparing to the CFACN. It is also superior to ROI-CFACN with SORT.

Table 6. The comparison results of the overall proposed method with detection standalone (without tracking algorithm) and with simple online real-time tracking (SORT) (including the proposed detector as detector algorithm and SORT for tracking).

Frameworks	mAP%	FPS
Proposed method without tracking [CFACN + ROI-CFACN]	97.11	52
Proposed detection method with SORT [CFACN + ROI-CFACN + SORT]	97.74	54
Overall proposed detection and tracking method	98.30	56

Figure 11 shows a comparison between the detection results of CFACN alone and of CFACN with ROI-CFACN. As shown in the figure, the CFACN network detected the drone at frame 315 and misdetected between frames 316 to 318. When it reached frame 319, the CFACN started to detect the targets. As presented in this article, the detection result obtained from frame 315 was used as the ROI to minimize misdetection. Then, the ROI was passed through ROI-CFACN, in order to obtain the target bounding box location for frame 316, rather than using CFACN. As shown in Figure 11 (CFACN+ROI-CFACN), from frames 316 to 318, all targets were detected and tracked. This shows that, by using the two frameworks (i.e., CFACN and ROI-CFACN), the detection accuracy can be increased, while reducing both misdetection and the computational cost. This improvement is mainly due to the detector algorithm using more time in the ROI-CFACN and RSCR path than the CFACN and RSCR path.

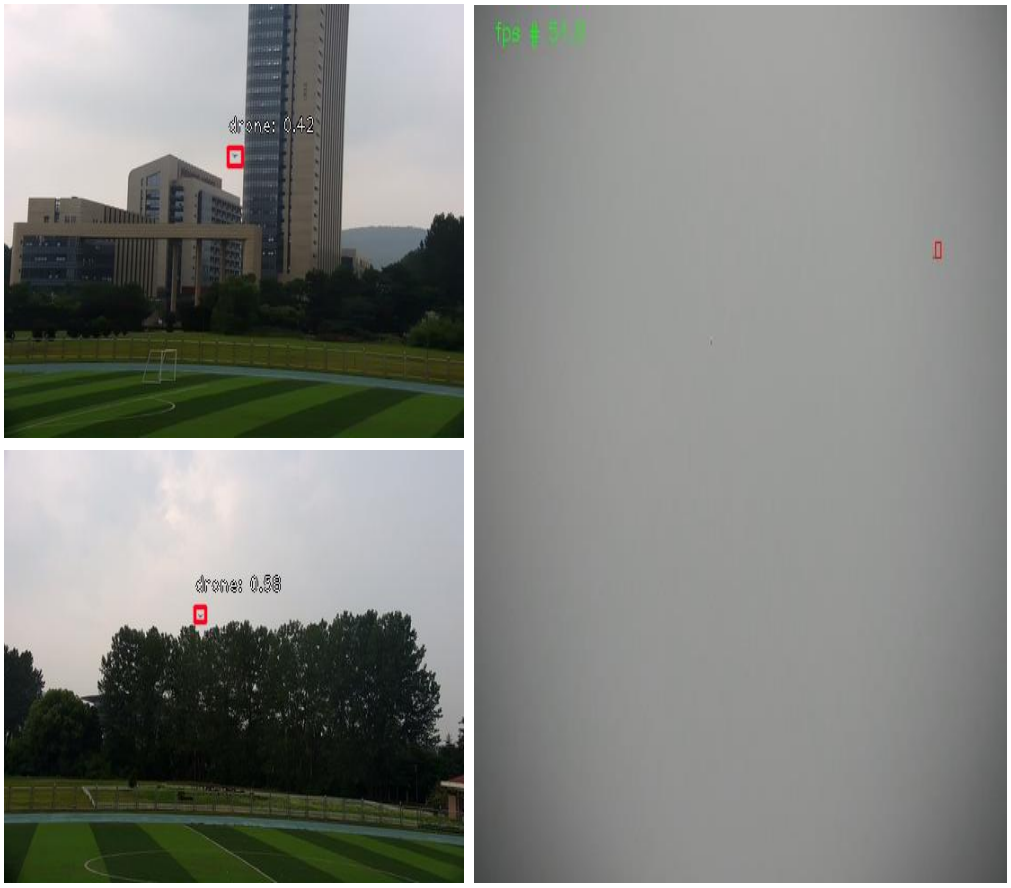


Figure 11. Comparison of CFACN detector and the proposed detection and tracking method [CFACN + ROI-CFACN].

3.4. Experiment Results in Different Challenges Scenarios

It is worth mentioning that the target object's size plays a crucial role in detection and tracking applications. Small-sized objects are difficult to detect, as they have low resolution and are influenced by noise; after repeated convolution operations, the existing network does not fully represent the essential features of small targets. We achieved good detection accuracy by extracting the features at different convolution levels, in order to aggregate different scale features, and enlarged small targets by using ROIs. CFACN could detect a drone of size 6 $\times$ 6 pixels, while ROI-CFACN could detect a drone covering 5 $\times$ 5 pixels (i.e., at a distance of up to 1 km), as shown in Figure 12. Other CNN methods require a large amount of downsampling during the convolutional process to reduce redundancy and the computational cost. The method proposed in this article only needs 4$\times$ downsampling, such that it can detect small targets. Besides, the non-utilization of anchors also significantly improved the detection of small drones, as discussed previously. In summary, cross-scale aggregation, the downsampling scale, ROI, and free-anchor box helped us to detect a 5 $\times$ 5-pixel size drones with a high accuracy.

Figure 12 shows the detection and tracking results for the proposed method. In Figure 12a, the confidence score is shown, while Figure 12b shows the algorithm's running time, in terms of FPS. As shown in Figures 12 and 13, the combination of CFACN and ROI-CFACN indicates the proposed method adequacy and robustness in detecting and tracking drones. For a drone of size greater than 8 $\times$ 8 pixels, the network could effectively localize the drone in a complex and cluttered environment. However, for those of a smaller size, the CFACN sometimes could not detect the target drone, if it was below the horizon or had a complex background however, if the ROI was obtained, the ROI-CFACN could detect the target drone above the horizon or in a complex background.



a b

Figure 12. Results of the proposed system under far distance: (**a**) Illustration of confidence score and (**b**) illustration of FPS.



Figure 13. Illustration of the proposed system in different challenging scenarios (i.e., occlusion, complex background, camera motion, and so on).

Furthermore, as shown in Figure 13, the proposed method showed robustness to challenges such as camera motion, occlusion, complex background, far distance, scale variation, variable illumination, in-plane, and out-plane. The proposed network could also detect the low-resolution target caused by the camera motion without any minor change however, for low-resolution images with a complex background, the algorithm failed to detect and track the target. In general, the proposed method can easily detect and track a drone with a low false alarm rate under challenges such as occlusion. In addition, with complex backgrounds, insufficient lighting, drones below the horizon, in-plane, out-plane, in a scaled variant where the drone becomes smaller or bigger, under illumination variation (e.g., where the drone flies against the sun), and in a cluttered video image, the proposed method can effectively detect and track the drone.

3.5. Target Detection and Tracking in the Collected Data Set

Figure 14 shows more results for the proposed method. As can be seen in Figure 14a, a small-sized target could be detected and tracked. Simultaneously, the detection and tracking method effectively detected and tracked a low-resolution drone at a far distance. Figure 14b shows a drone that was difficult to identify, due to camera motion and the complex background however, as shown in Figure 14b, the detection, tracking, and direction of the drone were obtained. Figure 14c shows the tracking path for controller adjustment, such that the drone would move toward the center of the image plane.



a b c

Figure 14. Detection and tracking results for the proposed method: (**a**) Detection and tracking result for a small drone; (**b**) illustration of detection, tracking, and direction of the target; and (**c**) illustration of tracking and direction of the target. The direction shows the state shift of the drone from the previous state to the current state. The teal color indicates the detection result, red indicates the tracking state, and black indicates the turret rotation direction.

3.6. Outdoor Experiment Result

As shown in Figure 15, the proposed method's effectiveness, for different ranges and in real-time, is illustrated. On the top-left side (i.e., in frame 1), the drone was detected and tracked at the frame's top-right. In frame 1468, the controller adjusted the rotating turret to focus and rotate to the left, toward the target, in order to align the target centroid into the image plane and to adjust its focus area to the center plane. In frame 1764, the rotating turret tracked the target and stayed to the left side on the bottom. Finally, frame 2881 shows how the target was tracked along the horizontal direction autonomously.



Figure 15. Illustration of the proposed method used for the detection and tracking of drones in real-time.

As can be seen in Figure 15, the implemented method could effectively and robustly detect and track a selected target. We observed that using various kinds of hardware and testing them in different environments helped us to prove the robustness, applicability, and necessity of the conducted experiment. In most of the tests, the proposed method in this article showed an improved accuracy however, the proposed method failed to detect and track the target in some cases and environments. Specifically, the method had a drawback in detecting small drones when the background was cluttered or complex and when the camera motion was not stable. It could also not detect when two drone centroid overlapped thus, further work is needed to detect both drones, instead of just one.

Figure 16 shows the experimental results of detection and tracking in a failure case. The target was difficult to visualize, but the CFACN effectively discovered the target in frame 2 and adjusted the controller accordingly. In frames 3 and 4, the ROI-CFACN effectively found the target, but missed the target in frames 5–7. Then, in the recovery time, the Kalman prediction state was utilized to find the ROI in frame 8, such that the ROI-CFACN localized the target in frame 9.



Figure 16. Detection and tracking result of the proposed method with a failure case.

4. Discussion

Most recently proposed methods have shown that detecting and tracking drones from video data is a challenging task. However, by using deep CNN approaches, it has become much easier to detect and identify drones. This article proposed an autonomous real-time center key point-based unmanned aerial vehicle detection and tracking method using CFA for surveillance applications. In the proposed method, two CNN frameworks—CFACN and ROI-CFACN—and the Kalman prediction state were used to detect the drone. Moreover, to track the target, the Kalman filter updated state for ROI-CFACN was adopted. The utilization of both frameworks (i.e., CFACN and ROI-CFACN) facilitated the effective discovery of drones in real-time. To further improve the method's effectiveness, the detection and tracking algorithms were applied to work in a parallel manner (i.e., asynchronous mode). The two parallel algorithms were used to improve the method's robustness and to reduce the latency that occurs due to the rotating turret. On the tracking side, the implemented tracking approach is simple, fast, and online, which helps to track and boost the tracking speed under limited computational time. Even though the proposed method was designed for drone surveillance applications, it can also be used in various other applications, such as UAV collision avoidance, monitoring of other objects, and so on. Currently, we are focusing on how to improve the detection accuracy and how to detect two drones when their centroids overlap.

5. Conclusions

This article aimed to detect, identify, and track a drone for surveillance applications. The results of the implemented method indicated that the detection method and tracking algorithm proposed in this article are key to successfully implementing real-time center key point-based unmanned aerial vehicle detection and tracking through cross-scale feature aggregation. The CFACN, ROI-CFACN, and Kalman filter provided a suitable framework for detecting and tracking a drone against challenges such as long-term occlusion, a drone below the horizon, camera motion, scale variation, and variable illumination. The overall method achieved 98.3% mAP at 56 FPs on our dataset. In summary, we can boldly state that utilizing CFACN with ROI-CFACN provided a method that could achieve better performance than other similar apporaches, such as BiFPN, YoloV5, CenterNet, and their variations. The detection and tracking approaches were simple, fast, accurate, and end-to-end differentiable without using any post-processing. Our results were encouraging, revealing a new means for real-time recognition and tracking-related tasks. For future work, further investigation is still needed to improve the accuracy when identifying drones at the far range below the horizon—where the drone's appearance may have very low contrast, with respect to the local background—and when two target centers overlap.

Author Contributions: M.B. and G.C.U. were mainly responsible for the construction of the unmanned aerial vehicle data set, development of the proposed algorithm, writing of the manuscript, and conducting the experiments. M.X., L.H. and R.C. commented on the manuscript and made useful suggestions. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Fire and Rescue Department technology Program of MEM under Grant 2020XFZD05, in part by the National Key R & D Program of China under Grant 2018YFC0810202, in part by Shaanxi Innovative Talents Promotion Plan-Science and Technology Innovation Team under Grant 2019TD-002, and in part by the Fundamental Research Funds for the Central Universities under Grant JB190204.

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

References

1. Birnbach, S.; Baker, R.; Martinovic, I. Wi-Fly: Detecting Privacy Invasion Attacks by Consumer Drones. In Proceedings of the NDSS, San Diego, CA, USA, 26 February–1 March 2017; pp. 1–25. [CrossRef]
2. Gill, R. Drones and Business Espionage—A New Corporate Threat. 2017. Available online: https://www.dronedefence.co.uk/drones-and-business-espionage/ (accessed on 23 September 2020).

3. Czaszejko, T.; Sookun, J. Acoustic emission from partial discharges in cable termination. In Proceedings of the International Symposium on Electrical Insulating Materials, Niigata, Japan, 1–5 June 2014; pp. 42–45. [CrossRef]
4. Samaras, S.; Diamantidou, E.; Ataloglou, D.; Sakellariou, N.; Vafeiadis, A.; Magoulianitis, V.; Lalas, A.; Dimou, A.; Zarpalas, D.; Votis, K.; et al. Deep learning on multi sensor data for counter Uav applications—A systematic review. *Sensors* **2019**, *19*, 4837. [CrossRef] [PubMed]
5. Hammer, M.; Hebel, M.; Laurenzis, M.; Arens, M. Lidar-based detection and tracking of small UAVs. In Proceedings of the SPIE SECURITY + DEFENCE, Berlin, Germany, 10–13 September 2018; Volume 10799, pp. 177–185. [CrossRef]
6. Hommes, A.; Shoykhetbrod, A.; Noetel, D.; Stanko, S.; Laurenzis, M.; Hengy, S.; Christnacher, F. Detection of acoustic, electro-optical and RADAR signatures of small unmanned aerial vehicles. In Proceedings of the SPIE SECURITY + DEFENCE, Edinburgh, UK, 26–29 September 2016; Volume 9997. [CrossRef]
7. Sapkota, K.R.; Roelofsen, S.; Rozantsev, A.; Lepetit, V.; Gillet, D.; Fua, P.; Martinoli, A. Vision-based Unmanned Aerial Vehicle detection and tracking for sense and avoid systems. In Proceedings of the EEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1556–1561. [CrossRef]
8. Wu, Y.; Sui, Y.; Wang, G. Vision-Based Real-Time Aerial Object Localization and Tracking for UAV Sensing System. *IEEE Access* **2017**, *5*, 23969–23978. [CrossRef]
9. Lyu, Y.; Pan, Q.; Zhao, C.; Zhang, Y.; Hu, J. Feature article: Vision-based UAV collision avoidance with 2D dynamic safety envelope. *IEEE Aerosp. Electron. Syst. Mag.* **2016**, *31*, 16–26. [CrossRef]
10. Unlu, E.; Zenou, E.; Riviere, N.; Dupouy, P.E. Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSJ Trans. Comput. Vis. Appl.* **2019**, *11*. [CrossRef]
11. Opromolla, R.; Inchingolo, G.; Fasano, G. Airborne visual detection and tracking of cooperative UAVs exploiting deep learning. *Sensors* **2019**, *19*, 4332. [CrossRef] [PubMed]
12. Hauzenberger, L.; Ohlsson, E.H. Drone Detection using Audio Analysis. Master's Thesis, Department of Electrical and Information Technology, Lund, Sweden, 2015.
13. Mendis, G.J.; Randeny, T.; Wei, J.; Madanayake, A. Deep learning based doppler radar for micro UAS detection and classification. In Proceedings of the MILCOM 2016—2016 IEEE Military Communications Conference, Baltimore, MD, USA, 1–3 November 2016; pp. 924–929. [CrossRef]
14. Ganti, S.R.; Kim, Y. Implementation of detection and tracking mechanism for small UAS. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 1254–1260. [CrossRef]
15. Nguyen, P.; Ravindranathan, M.; Nguyen, A.; Han, R.; Vu, T. Investigating cost-effective RF-based detection of drones. In Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use, Singapore, 26 June 2016; pp. 17–22. [CrossRef]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
17. LeNet-5, Convolutional Neural Networks. 2015. Available online: http://yann.lecun.com/exdb/lenet (accessed on 14 May 2018).
18. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
20. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid Task Cascade for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
21. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; Volume 2015, pp. 1440–1448. [CrossRef]
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *European Conference on Computer Vision*; Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics); Springer: Cham, Switzerland, 2016; Volume 9905 LNCS, pp. 21–37. [CrossRef]
23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
24. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, 2980–2988. [CrossRef]
25. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
26. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2019.
27. Liu, T.; Abd-Elrahman, A.; Morton, J.; Wilhelm, V.L. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *GISci. Remote Sens.* **2018**, *55*, 243–264. [CrossRef]

28. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
29. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 850–859.
30. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
31. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and real-time tracking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3464–3468.
32. Kalman, R. Emil, A New Approach to Linear Filtering and Prediction Problems. *Trans. ASME J. Basic Eng. Ser. D* **1960**, *82*, 35–45. [CrossRef]
33. Gordon, D.; Farhadi, A.; Fox, D. Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects. *arXiv* **2018**, arXiv:1705.06368.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
35. Krišto, M.; Ivasic-Kos, M.; Pobar, M. Thermal Object Detection in Difficult Weather Conditions Using YOLO. *IEEE Access* **2020**, *8*, 125459–125476. [CrossRef]
36. Ivasic-Kos, M.; Kristo, M.; Pobar, M. Person Detection in Thermal Videos Using YOLO. In *Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing*; Bi, Y., Bhatia, R., Kapoor, S., Eds.; Springer: Cham, Swizerland, 2020; Volume 1038.
37. Drone Image. Available online: www.google.com (accessed on 20 February 2020).
38. YoloV5. Available online: https://github.com/avBuffer/Yolov5_tf (accessed on 13 March 2021).
39. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.