electronics

MDPI

Article

An Improved Yolov5 for Multi-Rotor UAV Detection

Bailin Liu [1,2,*] and Huan Luo [2,3]

[1] School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China
[2] New Network and Testing Control National and Local Joint Engineering Laboratory, Xi'an 710021, China; luohuan@st.xatu.edu.cn
[3] Ordnance Science and Technology College, Xi'an Technological University, Xi'an 710021, China
* Correspondence: xatulbl@xatu.edu.cn

Abstract: Multi-rotor drones have a wide range of applications in practical scenarios; however, the use of multi-rotor drones for illegal acts is also on the rise, in order to improve the recognition accuracy of multi-rotor drones. A new multi-rotor drone detection algorithm is proposed. Firstly, the Yolov5 backbone is replaced with Efficientlite, thus reducing the number of parameters in the model. Secondly, adaptively spatial feature fusion is injected into the head of the baseline model to facilitate the fusion of feature maps with different spatial resolutions, in order to balance the accuracy loss caused by the lightweight of the model backbone. Finally, a constraint of angle is introduced into the original regression loss function to avoid the mismatch between the prediction frame and the real frame orientation during the training process in order to improve the speed of network convergence. Experiments show that the improved Yolov5s exhibits better detection performance, which provides a superior method for detecting multi-rotor UAVs in real-world scenarios.

Keywords: multi-rotor UAV detection; Yolov5; adaptively spatial feature fusion

Citation: Liu, B.; Luo, H. An Improved Yolov5 for Multi-Rotor UAV Detection. *Electronics* **2022**, *11*, 2330. https://doi.org/10.3390/electronics11152330

Academic Editor: Carlos Tavares Calafate

Received: 1 July 2022
Accepted: 24 July 2022
Published: 27 July 2022

1. Introduction

With the rise of the fourth industrial revolution and the rapid development of artificial intelligence, computer vision and other technologies, multi-rotor UAV manufacturing technology is developing rapidly. It has become an important part of military technological warfare [1]. Its applications in the battlefield are mainly divided into reconnaissance and surveillance, accompanying escort, precision destruction strikes, and cluster operations. Not only that, but it also has quite a wide range of applications in the fields of agricultural production, animal husbandry, industry, and urban management, for example, environmental pollution monitoring, mineral exploration, and the detection of life activities of wild animals [2]. A multi-rotor UAV is a special type of unmanned rotorcraft with three or more rotor axes. Compared to ordinary UAVs, it is low-cost, flexible and lightweight, easy to install, has the ability to take off and land independently, has a high degree of intelligence, can adapt to various complex terrain environments, is easy to maneuver, and can fly in various attitudes, such as hovering forward and side flight in tight spaces [3]. The previous multi-rotor UAV detection methods mainly rely on audio signal analysis, radar data analysis, radio frequency signal analysis and computer vision technology for detection [4]. Hauzenberger et al. proposed a speech detection method through linear predictive coding (LPC) to analyze the unique sounds emitted by UAVs to detect UAVs [5]. However, in noisy environments, when the wings of multi-rotor UAVs rotate, the tiny sound produced is easily disturbed by noise. Mohajerin et al. proposed to use the characteristics of radar trajectories to detect UAVs [6]. However, radar signals have a short detection distance in bad weather, such as rain, fog and haze, which cannot meet the long-distance detection requirements [7]. Al -Emadi et al. proposed a UAV detection technique using CNN to collect RF signals generated during the communication process between the UAV and the controller for study in the training phase [8]; however, some UAVs may not be controlled

via a wireless connection, but it has been programmed to fly a specific route so that the network cannot collect the radio frequency signals generated during the communication process.

With the wide application of neural networks in the field of computer vision in recent years, it provides a new idea for multi-rotor UAV detection. Target detection algorithms are mainly divided into traditional target detection algorithms based on artificial features and target detection techniques based on deep neural networks [9]. Yicheng Liu et al. proposed a method to detect UAVs by using a support vector machine and naive Bayesian manual feature extraction. This traditional manual feature extraction method not only relies on manual feature extraction but also requires a lot of computation [10]. Influenced by the successful application of deep learning in the field of computer vision, multi-rotor UAV detection has also made some preliminary attempts using deep learning in recent years. Vasileios Magoulianitis et al. propose to enlarge the image twice by super-resolution technology (SR) before it enters the detection network to increase its recall ability, and then use super-resolution technology and a Fast R-CNN model at the same time for end-to-end matching [11]. In order to give full play to the effect of joint optimization, this traditional convolutional neural network-based UAV detection method has a great improvement compared to the manual extraction method, but this method takes a long time and cannot meet the real-time detection requirements. demand. Wei et al. used SSD Inception V2 [12], SSD MobileNet [13], R-FCN Resnet [14], Faster R-CNN Inception Resnet [15], Faster R-CNN Resnet [16] and Yolov2 [17] to UAV perform real-time detection. In terms of speed, SSD MobileNet is the fastest model, followed by Yolov2, followed by SSD Inception V2. In terms of detection accuracy, the best effect is given by FRCNN Inception Resnet, followed by RFCN Resnet, and then Yolov2. Yolov2 achieves a balance between detection speed and accuracy [18]. Nader Al-Qubaydhi et al. used Yolov3, Yolov4, and Yolov5 to detect UAVs respectively, analyzed these results respectively, and found that Yolov5 is superior in UAV detection [19].

The current mainstream target detection technology based on deep neural networks is divided into a one-stage method and two-stage method. For the two-stage approach, the first stage generates a number of candidate region boxes based on a region suggestion network (RPN), the second stage classifies the candidate regions by a softmax function to determine whether there are detection targets in these boxes, and then the suggested boxes are corrected by a boundary regression function [20]. The above-mentioned Faster R-CNN Inception Resnet, Faster R-CNN Resnet, and R-FCN Resnet belong to two-stage methods. Although variant algorithms, such as Faster R-CNN Inception Resnet and Faster R-CNN Resnet, have made many improvements on the basis of R-CNN, this has not changed the disadvantage that the second-order target detection is slow in terms of detection speed, so some researchers took a different approach, combining the generation of the candidate region frame and the regression of the candidate region box into one step, and proposed a single-stage detector, SSD. The single-stage detector is much better than the two-stage detector in terms of inference speed, and the detection accuracy cannot meet people's needs. So, the researchers made improvements on the basis of SSD and launched the above-mentioned algorithms, such as SSD MobileNet, SSD Inception V2, and Yolo series. They are all one-stage methods. In the one-stage approach, the detection network directly classifies and regresses anchor boxes that are densely sampled from the feature map, and omits the region proposal network (RPN). The two-stage detection method achieves good results in detection accuracy, but the real-time performance is poor. Meanwhile, single-stage detection methods have lower accuracy but faster detection speed [21].

In most previous studies, the limited computational resources of a large number of practical application platforms are often ignored in order to improve the detection accuracy of network detection models. Therefore, the focus of research on multi-rotor UAV target detection should be on how to improve detection accuracy while keeping the model lightweight, and in this paper, we propose an improved Yolov5 multi-rotor UAV detection model. First, the Yolov5 backbone is replaced with Efficientlite, thus reducing

the number of parameters in the model. Then, the adaptive feature fusion technique is injected into the head of the baseline model to suppress the inconsistent information in the feature maps at different scales and facilitate the fusion of feature maps with different spatial resolutions in the model to balance the accuracy loss caused by the lightweight of the backbone part. Finally, the original loss function in Yolov5 is replaced with SIoU, and the angle cost criterion is introduced into the previous loss function metric to improve the speed of network convergence. Experiments show that the improved Yolov5 model can show better detection performance in the UAV_data dataset.

2. Materials and Methods

2.1. Yolov5

Yolov5 is a very popular, single-stage target detector that has a total of 4 models, Yolov5s, Yolov5m, Yolov5l, and Yolov5x, representing Yolov5_small, Yolov5_media, Yolov5_large, and Yolov5_extra large, respectively. As the model depth and width of the model increase, the number of parameters increases as well. Glen Jocher controls the depth of the model and the number of convolutional kernels with the depth_multiple and width_multiple parameters, respectively, to meet different detection requirements. yolov5 is divided into the following four parts: the input, the backbone network model, the neck network model, and the output. The backbone model is a convolutional neural network used to accumulate fine-grained images and generate image feature maps. The neck network model is responsible for combining the image features collected by the backbone model and then passing the integrated feature maps to the output, which is responsible for the detection and classification of the model [22]. The framework of the yolov5s is shown in Figure 1.

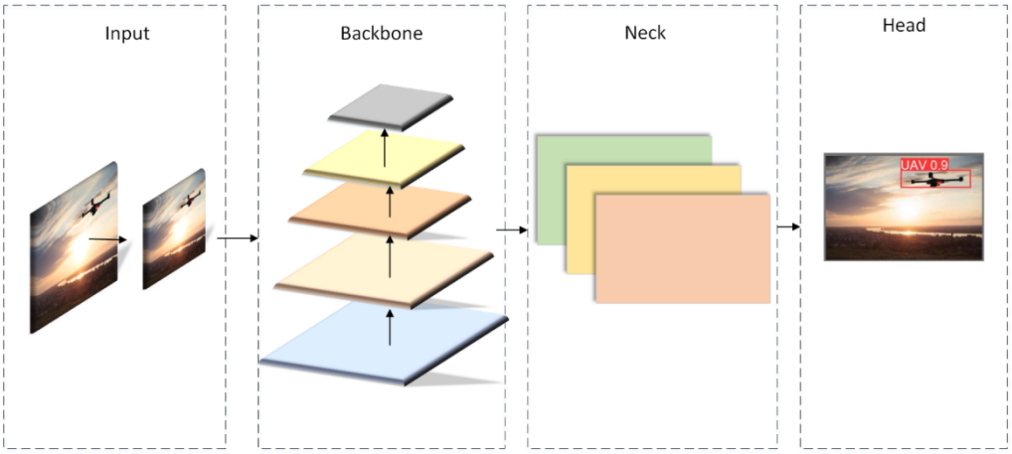

Figure 1. The framework of Yolov5s.

Iuput:

Yolov5 uses mosaic data augmentation on the input side and has built-in adaptive anchoring and adaptive scaling. Mosaic data augmentation is achieved by randomly cropping any four images and stitching them onto a single image as training data, reducing the size of the detection target in the image in terms of pixel points and allowing Yolov5 to perform better in small target detection. Once the dataset enters the network, Yolov5 scales the images in the dataset to a size of 640 × 640 and then uses the k-means clustering algorithm to calculate the anchor boxes that match the annotated boxes in this dataset and compares them with the preset anchor boxes. If the calculated best recall is greater than 0.98, it means that the preset anchor box size meets the requirements of this dataset; if it does not, then the network parameters are updated in the reverse direction.

Backbone:

In the backbone model, Yolov5 uses the FOCUS, SPP, and CSP structures. Focus is a slicing operation that obtains a value for every pixel in an image, similar to neighborhood downsampling, so that the input channel is expanded by a factor of four, but without information loss. The new image is then convolved to obtain a bipartite downsampled feature map without information loss. Yolov5 uses the CSPNet residual structure for both the backbone and neck network models, which divides the feature mapping of the basic layer into two parts and then combines them in a cross-stage hierarchy, reducing the computational effort while ensuring the integrity of the feature information [23].

Neck:

Yolov5 used a combination of FPN and PAN in the neck network model part. The shallow feature map will have more location information and less semantic information and as the number of neural network layers increases, the deeper feature map will have more semantic information, some small pixel points will be ignored and some location information will be lost; however, both types of information are important for target detection, so a deeper network structure to obtain more semantic information while preserving the original location information is essential for a good network structure. FPN passes strong semantic features from the top feature map to the lower feature map. Meanwhile, PAN conveys stronger localization features from the lower feature maps to the higher feature maps, and these two structures together enhance the feature fusion capability of the neck network model part [24].

2.2. Replace the Yolov5 Backbone with Efficientlite

Traditional convolutional neural networks improve the detection performance by deepening the number of network layers, widening the number of channel letters, and increasing the image resolution [25]. However, too deep a network structure causes the gradient to disappear and the network accuracy gain to be reduced, too wide a network structure causes the detection network to fail to extract the rich semantic information in the deeper layers of the image, and too high an image resolution causes additional difficulties in training. The backbone part of the Yolov5 model relies on five ordinary convolution operations of step 2 to spatially downscale the feature maps to obtain feature maps with different spatial resolutions and to learn the residual features through four C3 modules, which divide the feature information from the upper layers after spatial downscaling into two parts, with one part going through multiple bottleneck modules that are stacked and the other part needing to go through only one ordinary convolution module, to deepen the network structure as much as possible, while ensuring that the shallow rich location information is not lost and avoiding the loss of network update momentum due to gradient disappearance. In this research component, the baseline model backbone is replaced with the Efficientlite lightweight model. Efficientlite was proposed by Google in March 2022, and the structure searches the depth, width, and resolution of the composite scaling network through multi-objective neural architecture. Compared to Efficient, Efficientlite replaces the previous version, eliminates the squeeze-and-excitation structure and replaces the original swish activation function with the Relu6 activation function to avoid the loss of feature information in the non-linear layers [26]. It consists of a 3×3 normal convolutional layer, 7 MBConv and a 1×1 normal convolutional layer, average pooling layer and fully connected layer. MBConv is a feedforward neural network with fast connectivity, including a 1×1 ordinary convolutional module at the beginning and end of each of the $n \times n$ depth-separable convolutional modules to expand and compress the feature channels, and finally a dropout layer, where in each convolutional module is a combination of a convolutional layer, batchnormalization and Relu6 activation function, the internal structure of MBconv is shown in the Figure 2. The depth-separable convolution layer is divided into two parts, channel-by-channel convolution and point-by-point convolution [27]. The depth-separable convolution layer takes an image with $h \times w$ pixel points and c channels and passes it through a convolution operation with a kernel of $(h - H + 1) \times (w - W + 1)$ and a filter with 1 channel to output a feature map of $H \times W \times c$. The image is then expanded by C

$1 \times 1 \times c$ convolution kernels for channel expansion, as shown in Figure 3. Compared to normal convolution, as shown in Figure 4, depth-separable convolution greatly reduces the number of parameters and reduces the computational overhead required by the model, while obtaining the same sensory field.

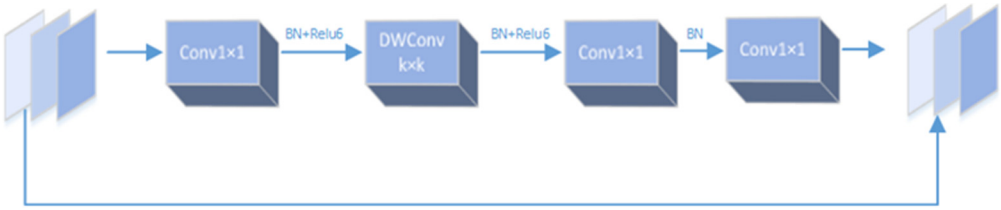

Figure 2. MBConv internal structure.

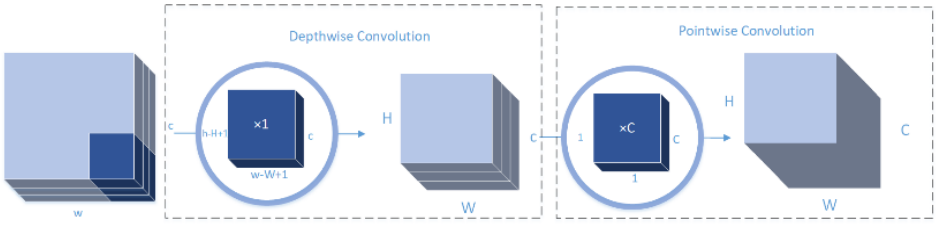

Figure 3. Depth-separable convolution.

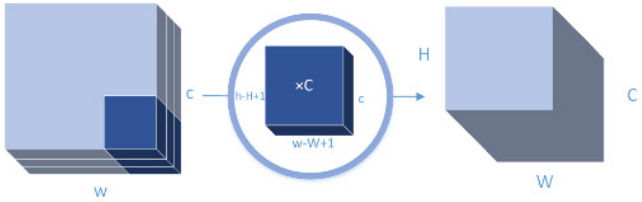

Figure 4. Ordinary convolution.

2.3. Improvements to the Feature Fusion Part of the Baseline Model

In order to balance the loss of accuracy caused by simplifying the network model, this paper improves the feature fusion part of the baseline model. Large targets are usually associated with deep feature maps and small targets with shallow feature maps. The detection network deepens the layers of the network by down-sampling and convolution layer by layer to obtain more semantic information and to retain feature maps of different depths. This is performed because usually deeper feature maps have more semantic information; however, some small pixel points are ignored and some location information is lost, while shallow feature maps have more location information and less semantic information, both of which are both important information for target detection. In the original baseline model Yolov5, feature maps at different scales rely on PANet as well as FPN for fusion and FPN is currently the most classical feature pyramid structure. When the image is input to the detection network, feature extraction is performed by the backbone part, feature maps with different spatial resolutions are produced at different depths of the network, FPN passes this feature information at different depths from the top to bottom, retaining the relevant spatial location information of the feature maps at different scales, supplementing the limited semantic information of the anchor points of the network prediction, and fusing this feature information by 1×1 convolution operation. However, this top-down laterally connected pyramid structure is prone to loss for shallow feature maps, resulting in low accuracy of small target detection. Both FPN and PANet compress the feature channels through convolutional operations, which inevitably brings about the problem of information redundancy. In this process, FPN and PANet give the same level of attention to the feature maps at each scale. However, feature maps with different spatial

resolutions have a large semantic gap due to different depths. The feature maps at different scales contain sometimes conflicting information about the size of target instances, which can interfere with the calculation of gradients during the training of the network and reduce the effectiveness of the feature pyramids. To avoid losing important information during the feature fusion process, Guiyi Yang et al. proposed a new model based on FPN, called PFF-FPN, which chose to use three different FPN structures to pass features and fuse the feature information of the corresponding layers to enhance the important information [28]. Golnaz Ghaisi et al. also proposed a new feature pyramid network structure based on FPN called NAS-FPN, which solves the large-scale search space problem in feature pyramids by combining a scalable search space with a neural combination search algorithm [29]. Zexuan Guo et al. proposed to use the BiFPN [30] structure to replace the original PANET structure in Yolov5. BiFPN is also a variant generated based on FPN that gives weights to different feature layers in the fusion process through residual connections and enables the fusion of features in both top-down and bottom-up paths. In this paper, the original feature fusion approach of Yolov5 is improved by introducing the adaptive feature fusion structure (ASFF), which consists of two parts, constant scaling, and adaptive fusion. For a spatially fractionated feature map at one level, ASFF adjusts other feature maps at different scales to the same resolution by up-sampling and down-sampling. When a target is specified and considered as positive in a feature map at one level, the corresponding regions in the feature maps at the other levels will be considered as the background, and then adaptively learn the weight parameters by back-propagating through the network, giving smaller weights to the locations where contradictory regions exist. The network is then trained to find the best fusion point by back-propagating the adaptive learning weight parameters, giving smaller weights to the locations where there are contradictory regions, larger weights to regions with consistent size information and filtering out redundant information that interferes with the normal detection of the network [31].

For constant scaling, since different levels of spatially resolved feature maps have different channel counts and resolutions, ASFF uses different scaling strategies for spatially fractionated feature maps at different levels. When the L2 level feature map is specified as positive, the detection network first compresses the L1 level feature map by a 1×1 convolution operation and then uses interpolation to increase the resolution. For L3-level feature maps, both the number of channels and the resolution can be modified using only 3×3 convolution layers with a step size of 2.

For adaptive fusion, I three detection heads of ASFF have the same structure, represented by ASFF_detect2, for example, when the n levels are adjusted to the feature vector $G_{ij}^{n \to 2}$ aI (i, j) on the L2 level feature map.

$$G_{ij}^{2} = \alpha_{ij}^{2} \cdot G_{ij}^{1 \to 2} + \beta_{ij}^{2} \cdot G_{ij}^{2 \to 2} + \gamma_{ij}^{2} \cdot G_{ij}^{3 \to 2} \tag{1}$$

The weights of L1, L2 and L3 for L2 are denoted by α_{ij}^{2}, β_{ij}^{2} and γ_{ij}^{2}, respectively.

$$\alpha_{ij}^{2} = \frac{e^{\lambda_{\alpha_{ij}}^{2}}}{e^{\lambda_{\alpha_{ij}}^{2}} + e^{\lambda_{\beta_{ij}}^{2}} + e^{\lambda_{\gamma_{ij}}^{2}}} \tag{2}$$

$$\beta_{ij}^{2} = \frac{e^{\lambda_{\beta_{ij}}^{2}}}{e^{\lambda_{\alpha_{ij}}^{2}} + e^{\lambda_{\beta_{ij}}^{2}} + e^{\lambda_{\gamma_{ij}}^{2}}} \tag{3}$$

$$\gamma_{ij}^{2} = \frac{e^{\lambda_{\gamma_{ij}}^{2}}}{e^{\lambda_{\alpha_{ij}}^{2}} + e^{\lambda_{\beta_{ij}}^{2}} + e^{\lambda_{\gamma_{ij}}^{2}}} \tag{4}$$

Three weights, $\lambda^2_{\alpha_{ij}}$, $\lambda^2_{\beta_{ij}}$, $\lambda^2_{\gamma_{ij}}$, are obtained by backward learning propagation of the network, α^2_{ij}, β^2_{ij} and γ^2_{ij} are obtained by the softmax function. α^2_{ij}, β^2_{ij} and γ^2_{ij} belong to the interval from 0 to 1 and add up to 1.

According to the improvement of the backbone and the head of the baseline model Yolov5 in Sections 2.2 and 2.3, the improved Yolov5 model is obtained, and the specific structure is shown in Figure 5.

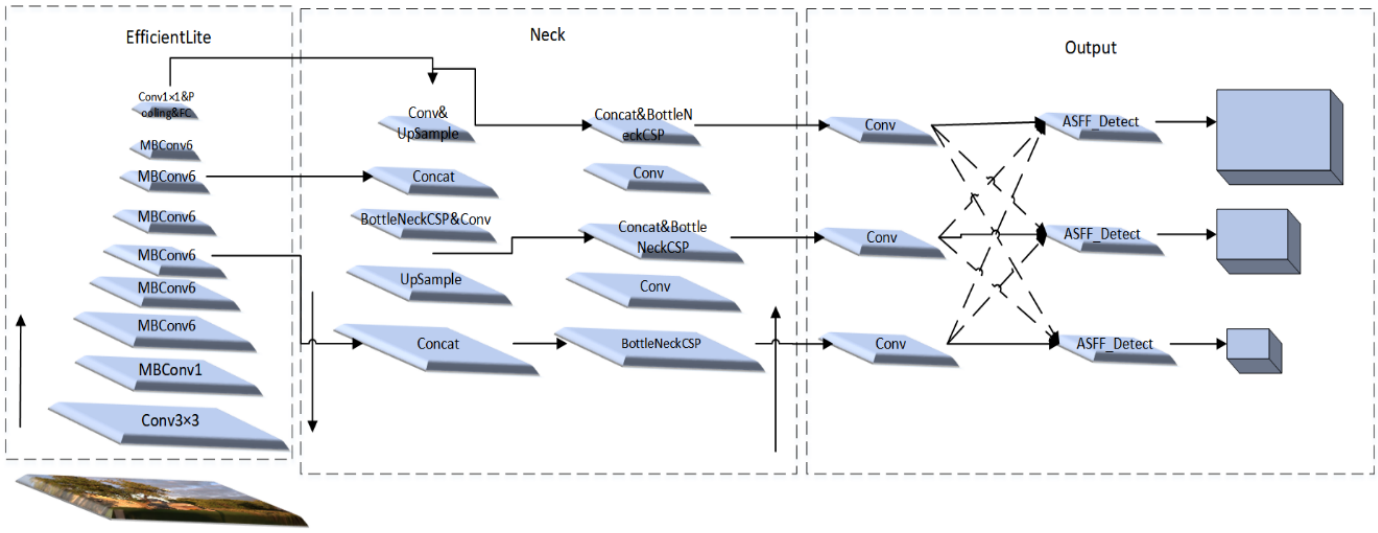

Figure 5. The internal structure of improved Yolov5.

2.4. Optimization of the Regression Loss Function

Yolov5′s built-in loss function is GIoU, which is based on a variant generated by IoU. It compensates for the problem that IoU prevents the network from optimizing when the overlap area of the two objectives is zero [32]. However, GIoU is more dependent on the IoU term and requires several iterations to converge when the predicted bounding box is horizontal or vertical. To solve this problem, Zhaohui Zheng et al. proposed DIoU, which speeds up the convergence of the network by minimally normalizing the centroids of the two bounding boxes, and suggested that a good bounding box loss function should have three important metrics, namely overlap area, centroid distance and aspect ratio [33]. The SIoU introduced in this paper argues that the above-mentioned IoU, DIoU, and GIoU all ignore the problem of directional mismatch between the GT and the prediction detection frame, which may lead to the problem of the prediction frame floating around during the training process, thus affecting the convergence speed of the network. To solve this problem, SIoU first makes a prediction in the x- or y-axis direction and then allows the prediction frame to continue to move in that direction. To achieve this, the module tends to zero α or β as much as possible and introduces four metrics in the SIoU module, angle cost, distance cost, shape cost, and IoU cost [34]; the formula for angle cost is shown in Formula (5), and the schematic is shown in the Figure 6.

$$\Lambda = 1 - 2 * \sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{5}$$

where

$$x = \frac{c_h}{\sigma} = \sin(\alpha) \tag{6}$$

where σ is the distance between the centroid of the ground truth bounding box and the centroid of the prediction box.

$$c_h = max(b^{gt}_{c_y}, b_{c_y}) - min(b^{gt}_{c_y}, b_{c_y}) \tag{7}$$

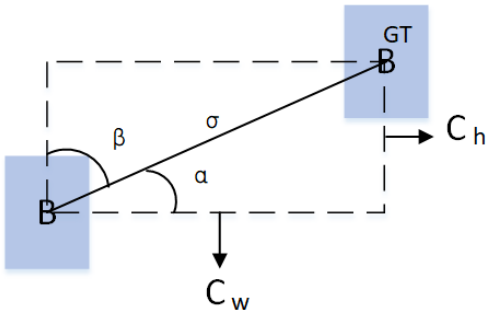

Figure 6. The scheme for calculation of angle cost.

The formula for distance cost is shown in Formula (8).

$$L_{dis} = \sum_{t=x,y} \left(1 - e^{-\gamma \rho_t}\right) \tag{8}$$

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w}\right)^2, \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h}\right)^2, \gamma = 2 - \Lambda \tag{9}$$

From Formula (8), it can be observed that the smaller the angle between the center point of the ground truth bounding box and the center point of the prediction box, the smaller the value of L_{dis} and the smaller the weight of distance cost in the loss function, and the larger L_{dis} becomes as the angle approaches $\frac{\pi}{4}$.

The formula for shape cost is shown in Formula (10).

$$L_{shape} = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta} \tag{10}$$

$$\omega_w = \frac{\left|w - w^{gt}\right|}{max(w, w^{gt})}, \omega_h = \frac{\left|h - h^{gt}\right|}{max(h, h^{gt})} \tag{11}$$

where w and w^{gt} refer to the width of the prediction box and the ground truth bounding box, respectively, and h and h^{gt} refer to the height of the prediction box and the ground truth bounding box, respectively.

The final loss function is shown in Formula (12).

$$L_{box} = 1 - IoU + \frac{L_{dis} + L_{shape}}{2} \tag{12}$$

3. Experimental Setup

3.1. Dataset

Similar to the human cognitive process, neural networks learn what an object is and are learning processes that require many samples to provide a sufficient number of features, and the quality and quantity of datasets play an important role in the performance of the network during training. However, the number of existing mature UAV samples is not very sufficient. Some datasets are cut from video streams, so the background information is more similar, which may lead to overfitting of the network during training, and some samples have too low pixel quality and the images are too blurred, which will make it difficult for the network to identify its features after convolution and pooling operations. Some of the datasets are mostly frontal views of the UAV in good lighting conditions, but in real scenarios, the UAV has various attitude changes and is affected by different lighting, and the pixel size in the image is greatly affected by the distance of the UAV from the lens. It can learn the details of the multi-rotor UAV feature information, but it will reduce the robustness of the detection network. Therefore, UAV_data should have some samples of drones under non-ideal conditions. We integrate the dataset obtained from kaggle and

augment the UAV dataset with data by adding random noise points and a binarization operation, which is a common process in image processing and refers to adding a random proportion of noise points to the image based on a set value. Binarization is the process of taking a grey-scale image with 256 levels of brightness and selecting the appropriate threshold to obtain a binarized image (black and white) that still reflects the overall and local characteristics of the image, the schematic is shown in Figure 7.

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

Figure 7. Original and data-enhanced images: (**a**) original, (**b**) randomly noise-increased; (**c**) binarized.

The final multi-rotor UAV dataset contains 1259 images of multi-rotor UAVs and the corresponding txt format annotation files; however, the annotation files in the dataset are manually annotated, so there are some errors, for example, the annotation boxes do not fit very closely to the actual detection target edges, and there are also some mislabels and omissions. Therefore, the dataset needs to be re-labeled and divided into a training set and a test set in a 4:1 ratio, and then the file format needs to be changed to the format required by Yolov5.

3.2. Training

To validate the effectiveness of the improved model, ablation experiments were conducted based on the UAV_data dataset, with the data expansion learning strategies of online-copy-paste and mixup. The experiments were conducted on a workstation equipped with an AMD Ryzen 9 5900HS processor, NVIDIA GeForce RTX 3050 Laptop graphics processor (16 gb RAM) and 512 gb RAM, and configured with CUDA11.4 and cuDNN11.4 to invoke GPU acceleration. The deep learning framework chosen was Pytorch, and the operating system was Windows 10. These experiments were conducted using SGD (stochastic gradient descent) to optimize the learning rate during training, with the same hyperparameter settings, weight decay set to 0.0005, momentum set to 0.8, batch size of 16 and epochs of 150.

4. Discussion

In this paper, the mean average precision (mAP) and the number of parameters were chosen as the main evaluation metrics for model detection performance and measuring model size, with precision and recall as reference metrics. Four potential categories can be generated by detecting network predictions, including true positive (TP), false positive (FP), true negative (TN), and false-negative (FN). Where TP refers to the number of correctly marked UAV positive samples, FP is the number of incorrectly marked UAV positive samples, TN is the number of correctly marked UAV negative samples and FN is the number of incorrectly marked UAV negative samples [35]. If the IoU between the detection box and the drone enclosing box is greater than 0.5, it is marked as TP. Otherwise, the detection box is marked as FP. If the drone enclosing box does not have a matching detection box, it is marked as FN. The IoU introduced here is the ratio of the intersection and the

concatenation of the network predicted edge (anchor box) and the true edge (true box), which can be expressed by the following equation:

$$\text{IoU} = \frac{area(ab \cap tb)}{area(ab \cup tb)} \tag{13}$$

The formula for precision is shown in the Formula (14) and the formula for recall is shown in the Formula (15). In simple terms, recall refers to the rate of complete checks and precision refers to the rate of accurate checks.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

Since in our detection task there is only one category of multi-rotor UAVs, N in the mAP equation is equal to 1 and in this case, mAP is equivalent to AP. Generally speaking, the higher the mAP, the higher the accuracy of the network. mAP is calculated as follows:

$$\text{mAP} = \frac{1}{N} \sum_{n=1}^{N} A_p(n) \tag{16}$$

In order to verify the effectiveness of the selected backbone part, a comparison test was conducted using the current mainstream lightweight backbone network based on the UAV_data dataset, respectively.

Table 1 shows that Yolov5s_EfficientLite has 46.3% fewer parameters than Yolov5s, but mAP0.5 has only 1% less, the recall is 0.5% higher than the original model, GFLOPS is 54.5% lower, and precision is only 2.53% lower, although Yolov5s_MobileNetV3 has 0.24M fewer parameters and the precision loss is nearly 3%. Yolov5s_ShuffleV2 not only has 0.19 M more parameters than Yolov5s_efficientLite, but also has 4.4% lower precision than Yolov5s.

Table 1. Comparison results of lightweight models.

Methods	Precision (%)	Recall (%)	mAP0.5 (%)	Parameters (M)
Yolov5s	94.96	89.01	92.78	7.02
Yolov5s_MobileNetV3	87.29	86.33	89.81	3.52
Yolov5s_ShuffleV2	87.36	86.84	88.38	3.95
Yolov5s_EfficientLite	92.43	89.52	91.76	3.76

In order to further verify the effectiveness of the proposed algorithm, we compared other current target detection algorithms in the field of computer vision with the algorithm proposed in this paper based on a new multi-rotor UAV dataset for experimental purposes. From Table 2, we can observe that our proposed algorithm is 2.04% higher than the baseline model mAP, while the number of parameters is only increased by 2.17 M, and compared with Yolov5s_ASFF, the number of parameters is less 26.2% with only a 0.63 loss in accuracy, although Yolov3-Tiny has 0.52 M less than our proposed network model and 7.04 less accuracy.

Table 2. Results of comparison of target detection models.

Methods	Precision (%)	Recall (%)	mAP0.5 (%)	Parameters (M)
Yolov3-Tiny	89.53	76.44	87.18	8.67
Yolov5s	94.96	89.01	92.78	7.02
Ours	93.54	91.09	94.82	9.19
Yolov5s_ASFF	94.14	93.17	95.45	12.46

We chose to use a multi-rotor UAV video obtained from the network to simulate the actual scene and used the best model derived from the training of Yolov5s and Yolov5s_CAM + ASFF + SIoU to detect this video, respectively, as shown in Figure 8. After the experiments, we found that the existing Yolov5s algorithm can already achieve a high accuracy during the training process, but there is still much room for improvement in the performance of the actual detection scenario. As shown in Figure 8a, it is clear that the baseline model Yolov5 produces missed detections, while in Figure 8b, it is clear that the improved Yolov5 can accurately identify multi-rotor UAVs. In Figure 8c, although the original model Yolov5 can also detect the UAV in the image, it has a lower confidence level of 0.35 for the detected target, while the improved Yolov5s has a higher confidence level of 0.61 for the detected UAV, indicating that the use of the improved Yolov5 has a better detection performance for multi-rotor UAV detection.

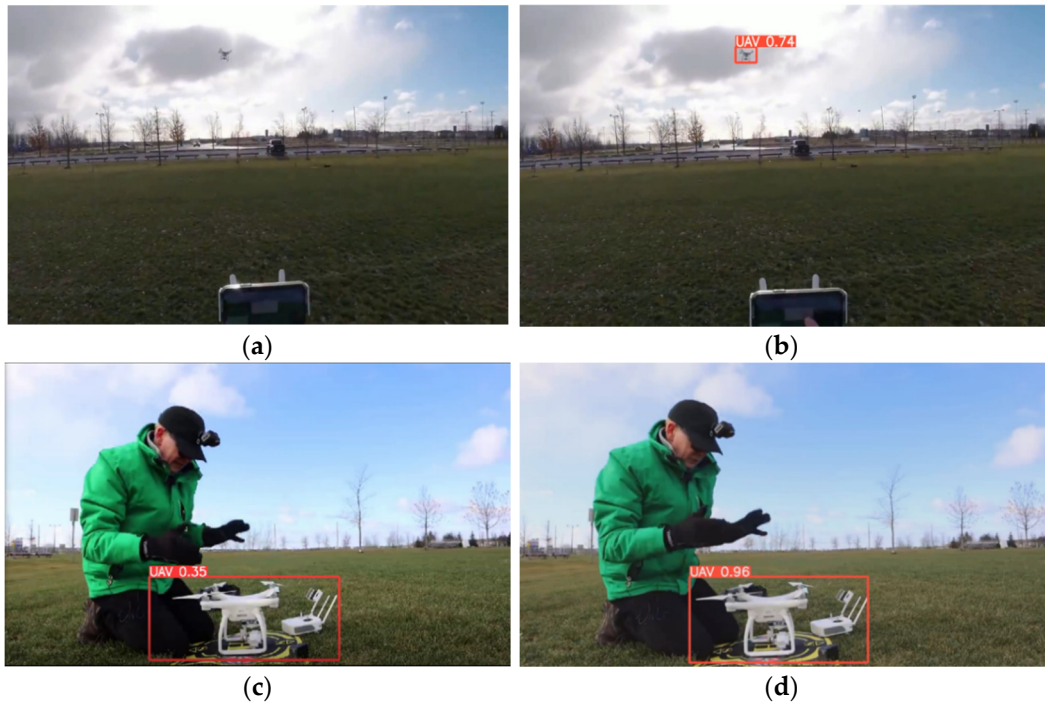

Figure 8. Comparison of network detection results before and after improvement ((**a**,**c**) show the results using yolov5; (**b**,**d**) show the results using improved algorithm).

5. Conclusions

In this paper, an improved Yolov5-based algorithm for multi-rotor UAV target detection is proposed. Firstly, the backbone part of the baseline model is replaced with EfficientLite to reduce the model parameters and lower the network computational overhead; secondly, the fusion of feature maps at different scales of the model is facilitated by introducing adaptive feature fusion techniques to balance the loss of accuracy caused by the lightweighting of the model, and then the angle as a constraint is introduced into the original loss function in the baseline model, reducing the degrees of freedom of the prediction frame. Then, different lightweight structures were experimentally compared as the backbone of the network based on the new multi-rotor UAV dataset, and EfficientLite was found to be balanced in terms of number of parameters and detection accuracy. Finally, we compare the improved model with the baseline model and other target detection algorithms and find that the improved model improves the target detection accuracy, while increasing the number of parameters by a smaller amount. In future work, we will continue to refine the improved model and attempt to deploy it to hardware platforms.

Author Contributions: Supervision, B.L.; writing—original draft preparation, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Shaanxi Provincial Natural Science Basic Research Program Project grant number 2019JM-603.

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. Han, H. Analysis of the Status of Basic Industries in Military Drone. *J. Converg. Cult. Technol.* **2020**, *6*, 493–498.
2. Mukhamediev, R.I.; Symagulov, A.; Kuchin, Y.; Zaitseva, E.; Bekbotayeva, A.; Yakunin, K.; Assanov, I.; Levashenko, V.; Popova, Y.; Akzhalova, A.; et al. Review of Some Applications of Unmanned Aerial Vehicles Technology in the Resource-Rich Country. *Appl. Sci.* **2021**, *11*, 10171. [CrossRef]
3. Noor, N.M.; Abdullah, A.; Hashim, M. Remote sensing UAV/drones and its applications for urban areas: A review. *IOP Conf. Ser. Earth Environ. Sci.* **2018**, *169*, 012003. [CrossRef]
4. Lei, T.; Tao, H.; Xu, C. Drone identification and location tracking based on YOLOv3. *Chin. J. Eng.* **2020**, *42*, 463–468.
5. Hauzenberger, L.; Holmberg Ohlsson, E. Drone Detection using Audio Analysis. Master's Thesis, Lund University, Lund, Switerland, 2015.
6. Mohajerin, N.; Histon, J.; Dizaji, R.; Waslander, S.L. Feature extraction and radar track classification for detecting UAVs in civillian airspace. In Proceedings of the 2014 IEEE Radar Conference, Cincinnati, OH, USA, 19–23 May 2014; pp. 0674–0679. [CrossRef]
7. Alipour-Fanid, A.; Dabaghchian, M.; Wang, N.; Wang, P.; Zhao, L.; Zeng, K. Machine Learning-Based Delay-Aware UAV Detection and Operation Mode Identification over Encrypted Wi-Fi Traffic. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 2346–2360. [CrossRef]
8. Al-Emadi, S.; Al-Senaid, F. Drone Detection Approach Based on Radio-Frequency Using Convolutional Neural Network. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020.
9. Wen, H.; Dai, F.; Yuan, Y. A Study of Yolo Algorithm for Target Detection. *Proc. Int. Conf. Artif. Life Robot.* **2021**, *26*, 622–625. [CrossRef]
10. Liu, Y.; Liao, L.; Wu, H.; Qin, J.; He, L.; Yang, G.; Zhang, H.; Zhang, J. Trajectory and image-based detection and identification of UAV. *Vis. Comput.* **2020**, *37*, 1769–1780. [CrossRef]
11. Magoulianitis, V.; Ataloglou, D.; Dimou, A.; Zarpalas, D.; Daras, P. Does Deep Super-Resolution Enhance UAV Detection? In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6. [CrossRef]
12. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
13. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. *arXiv* **2015**, arXiv:1512.02325.
14. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 379–387.
15. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
16. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with regionproposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
17. 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.
18. Wei, B.; Barczyk, M. Experimental Evaluation of Computer Vision and Machine Learning-Based UAV Detection and Ranging. *Drones* **2021**, *5*, 37. [CrossRef]
19. Al-Qubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Abdelaziz, A.A.; Razaque, A.; Alotaibi, A. Unauthorized Unmanned Aerial Vehicle Detection using YOLOv5 and Transfer Learning. *Preprints* **2022**. [CrossRef]
20. Guo, Z.; Wang, C.; Yang, G.; Huang, Z.; Li, G. MSFT-YOLO: Improved YOLOv5 Based on Transformer for Detecting Defects of Steel Surface. *Sensors* **2022**, *22*, 3467. [CrossRef] [PubMed]
21. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788. [CrossRef]

22. Rahman, R.; Bin Azad, Z.; Bakhtiar Hasan, M. Densely-Populated Traffic Detection Using YOLOv5 and Non-maximum Suppression Ensembling. In *Proceedings of the International Conference on Big Data, IoT, and Machine Learning. Lecture Notes on Data Engineering and Communications Technologies*; Arefin, M.S., Kaiser, M.S., Bandyopadhyay, A., Ahad, M.A.R., Ray, K., Eds.; Springer: Singapore, 2022; Volume 95. [CrossRef]
23. Zhu, L.; Geng, X.; Li, Z.; Liu, C. Improving Yolov5 with Attention Mechanism for Detecting Boulders from Planetary Images. *Remote Sens.* **2021**, *13*, 3776. [CrossRef]
24. Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Process.* **2021**, *15*, 3623–3637. [CrossRef]
25. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019.
26. Guo, J.-M.; Yang, J.-S.; Seshathiri, S.; Wu, H.-W. A Light-Weight CNN for Object Detection with Sparse Model and Knowledge Distillation. *Electronics* **2022**, *11*, 575. [CrossRef]
27. Cao, J.; Li, Y.; Sun, M.; Chen, Y.; Lischinski, D.; Cohen-Or, D.; Chen, B.; Tu, C. DO-Conv: Depthwise Over-Parameterized Convolutional Layer. *IEEE Trans. Image Process.* **2022**, *31*, 3726–3736. [CrossRef] [PubMed]
28. Yang, G.; Wang, Z.; Zhuang, S. PFF-FPN: A Parallel Feature Fusion Module Based on FPN in Pedestrian Detection. In Proceedings of the 2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI), Shanghai, China, 27–29 August 2021; pp. 377–381. [CrossRef]
29. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038. [CrossRef]
30. 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, 13–19 June 2020.
31. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
32. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [CrossRef]
33. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *arXiv* **2019**, arXiv:1911.08287. [CrossRef]
34. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
35. Zhao, J.; Zhang, X.; Yan, J.; Qiu, X.; Yao, X.; Tian, Y.; Zhu, Y.; Cao, W. A Wheat Spike Detection Method in UAV Images Based on Improved Yolov5. *Remote Sens.* **2021**, *13*, 3095. [CrossRef]