aerospace

MDPI

Article

EWH-YOLO: Efficient Small Unmanned Aerial Vehicle Detection with Weighted Bidirectional Feature Fusion and Hybrid Bounding Box Regression Loss

Wei Cheng * and Yunfeng Cao

College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
* Correspondence: nuaa_cw@nuaa.edu.cn

Abstract

Vision-based unmanned aerial vehicle (UAV) detection has become increasingly important since it is a key technology in aerial collision avoidance systems. However, the detection of small UAVs is still unsatisfactory in practical applications. To address this problem, this paper proposes EWH-YOLO, a novel deep convolutional neural network-based method for small UAV detection. First, an efficient feature extraction network is designed to exact the multi-level features of small UAVs while reducing the network parameters and computational complexity. Second, a weighted bidirectional feature fusion network is proposed to enhance the low-level and high-level features in the output feature maps. Third, a hybrid bounding box regression loss is introduced to evaluate the difference between the predicted bounding box and the ground-truth bounding box during training and improve the detection accuracy. Finally, a new dataset is created on the basis of considering small UAVs to verify the detection performance. Compared with the state-of-the-art methods, the proposed method achieves higher detection accuracy with lower model complexity. The experimental results demonstrate that the proposed detector significantly improves the detection performance of small UAVs.

Keywords: small UAV detection; deep convolutional neural network; efficient feature extraction; weighted bidirectional feature fusion; hybrid bounding box regression loss

1. Introduction

In recent years, unmanned aerial vehicles (UAVs) have become increasingly popular in various fields, such as logistics [1], agriculture [2], and inspection [3]. Given the rapid proliferation of UAVs, the technology of vision-based UAV detection has gained widespread attention since it plays a crucial role in aerial collision avoidance systems. However, the existing methods still cannot meet the application requirements due to the poor detection performance in detecting small UAVs. Therefore, it is of great significance to develop small UAV detection technology.

Thanks to the evolution of deep learning techniques, vision-based object detection methods that rely on deep learning have achieved significant progress [4]. The existing deep learning-based detectors can be separated into two streams. One stream comprises two-stage detectors which are represented by R-CNN series [5–9]. The two-stage approaches first generate the region proposals from the input images and then perform bounding box regression and classification based on the region proposals. The other comprises one-stage approaches which are represented by YOLO (You Only Look Once) series [10–19]. The

Check for updates

Academic Editor: Mostafa Hassanalian

Received: 13 July 2026
Revised: 22 August 2026
Accepted: 3 September 2026
Published: 4 September 2026

Copyright: © 2026 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.

one-stage approaches do not generate candidate regions and directly regress the bounding box and predict the object category. Compared with two-stage methods, one-stage methods have advantages in terms of model complexity. Currently, deep learning-based detectors are widely used in various application scenes [20–22]. However, the detection of small UAVs still faces significant limitations. First, due to the limited feature information and poor visual appearance of small UAVs, it is challenging to effectively extract and integrate the low-level and high-level information required for small UAV detection. Furthermore, many existing approaches have high network parameters and computational complexity, making them less suitable for resource-constrained devices. Additionally, small UAVs suffer from limited spatial size in images, which can easily result in inaccurate bounding box regression during training.

To address the above limitations, this paper proposes EWH-YOLO, a novel deep learning-based small UAV detector using an efficient feature extraction network (EFEN), a weighted bidirectional feature fusion network (WBFFN) and a hybrid bounding box regression loss (HBBRL). The EWH-YOLO significantly improves the detection accuracy while reducing model complexity. The primary contributions of this paper are summarized below.

(1). An efficient feature extraction network is constructed for the feature extraction of small UAVs. The proposed network can extract multi-level features required for small UAV detection while reducing network parameters and computational complexity.

(2). A weighted bidirectional feature fusion network is proposed to enhance the feature representation ability of small UAVs. It can make the output feature maps contain more representative low-level and high-level features to improve detection accuracy.

(3). A hybrid bounding box regression loss is introduced to train the network. It can effectively evaluate the difference between the true bounding box and the predicted bounding box during training and enhance the location learning ability of small UAVs.

(4). A new dataset containing small UAVs in various practical scenarios is constructed to train and test the EWH-YOLO. The experimental results show that the EWH-YOLO achieves better detection performance among other state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 describes the related works on vision-based UAV detection, feature extraction network, multi-level feature fusion, and bounding box regression loss. Section 3 presents a detailed exposition of the EWH-YOLO. Section 4 describes the self-constructed dataset, implementation details, evaluation metrics, comparative experiments, and ablation experiments. Section 5 provides the conclusions.

2. Related Works

2.1. Vision-Basied UAV Detection

Traditional vision-based UAV detection methods are generally designed based on hand-crafted features and classifiers, including Haar-like features [23], histogram of oriented gradient (HOG) [24], support vector machine (SVM) [25], and AdaBoost [26]. For example, Sapkotav et al. [27] utilized the AdaBoost algorithm with HOG feature descriptors for UAV detection tasks. Since the feature extraction capability is limited, traditional detection methods have poor accuracy and robustness.

In recent years, deep learning-based object detection methods have achieved great success and gradually replaced traditional detection methods. Currently, some researchers have begun to adopt deep learning-based methods to tackle the problem of UAV detection. Ojdanić et al. [21] developed a telescope-based optical UAV detection method by employing a Faster R-CNN detector. Yang et al. [28] proposed an enhanced YOLOv8 for the detection of bionic flapping-wing drones. Arsenos et al. [29] evaluated the detection performance of representative deep-learning algorithms in aerial object

detection. Lee et al. [30] designed a micro air vehicle (MAV) interceptor that adopted YOLOv4-tiny to detect the MAVs. Opromolla et al. [31] constructed a vision-based sense-and-avoid architecture that adopted a YOLOv2 detector to detect uncooperative flying objects. Although the aforementioned works have attempted to introduce deep neural network for UAV detection, the detection of small UAVs is still facing significant challenges in practical applications.

2.2. Feature Extraction Network

Given that feature extraction networks heavily affect detection performance, the design of feature extraction networks has received widespread attention and research in recent years. AlexNet [32], VGG [33] and Darknet-19 [11] are the representative earlier feature extraction networks applied to object detection, mainly composed of stacking multiple convolutional layers. With the increase in stacking layers, the deep network will encounter the degradation problem. He et al. [34] proposed a deep residual learning framework and introduced the residual network. They addressed the degradation problem via shortcut connections. Based on the Darknet-19 and residual network, Redmon et al. [12] designed a Darknet-53 for feature extraction. To further optimize the detection performance of the Darknet-53, Bochkovskiy et al. [13] employed CSPDarknet53 to obtain feature maps. They combined the cross stage partial network (CSPNet) [35] and Darknet-53 to build the feature extraction network. Despite improving the detection accuracy for some cases, these feature extraction networks still fail to achieve a satisfactory trade-off between detection accuracy and model complexity.

2.3. Multi-Level Feature Fusion

Multi-level feature fusion is an effective method to improve the detection accuracy of small objects. In early studies, deep convolutional neural network-based detectors only used the high-level features from the backbone to localize objects and predict their categories. Although the high-level feature maps have rich semantic information, these methods lack low-level information for the detection of small objects. To improve the detection accuracy, Liu et al. [36] adopted multiple feature maps with different scales to detect different sized objects. However, the feature representation ability of single-level feature maps is still insufficient for the detection of small objects. For this defect, Lin et al. [8] proposed the feature pyramid network (FPN). It builds a feature pyramid to merge the high-level features with the lower-level feature maps, which improves the detection accuracy of small objects. To enrich the low-level information in the output feature maps, Liu et al. [37] proposed a path aggregation network (PAN). Currently, PAN is being applied to many state-of-the-art object detection methods. However, the small UAV features in PAN is limited, and the low-level and high-level features are not fully used to improve the accuracy. Additionally, due to differences between multi-level feature maps, PAN does not effectively integrate the significant features of small UAVs.

2.4. Bounding Box Regression Loss

In object detection, learning the bounding box parameters is a significant task for object localization. Many existing detectors employ the intersection over union (IoU) loss [38] for learning the bounding box parameters. The overlap area is used to measure the distance between the predicted bounding box and ground-truth bounding box in the IoU loss. However, when the bounding boxes do not overlap, the IoU loss cannot evaluate the closeness between bounding boxes. Some improved works have been proposed to address this problem. Rezatofighi et al. [39] proposed a generalized IoU (GIoU) loss. They found the smallest convex shapes enclosing both the predicted bounding box and ground-truth bounding box to evaluate the distance between bounding boxes in non-overlapping cases.

Zheng et al. [40] developed a CIoU loss by considering the overlap area, central point distance and aspect ratio. However, these methods still have unsatisfactory performance for the bounding box regression of small UAVs.

3. Proposed Method

In this paper, a novel deep learning-based detector is designed for the detection of small UAVs, which is constructed based on YOLO11. The overall structure of the proposed EWH-YOLO, as depicted in Figure 1, is composed of the backbone, neck and head. First, the backbone extracts multi-level features from the input images. Then, the neck is adopted to realize the fusion of multi-level features. Finally, the head is used to realize object classification and localization. Details of the specific improvements are presented below.

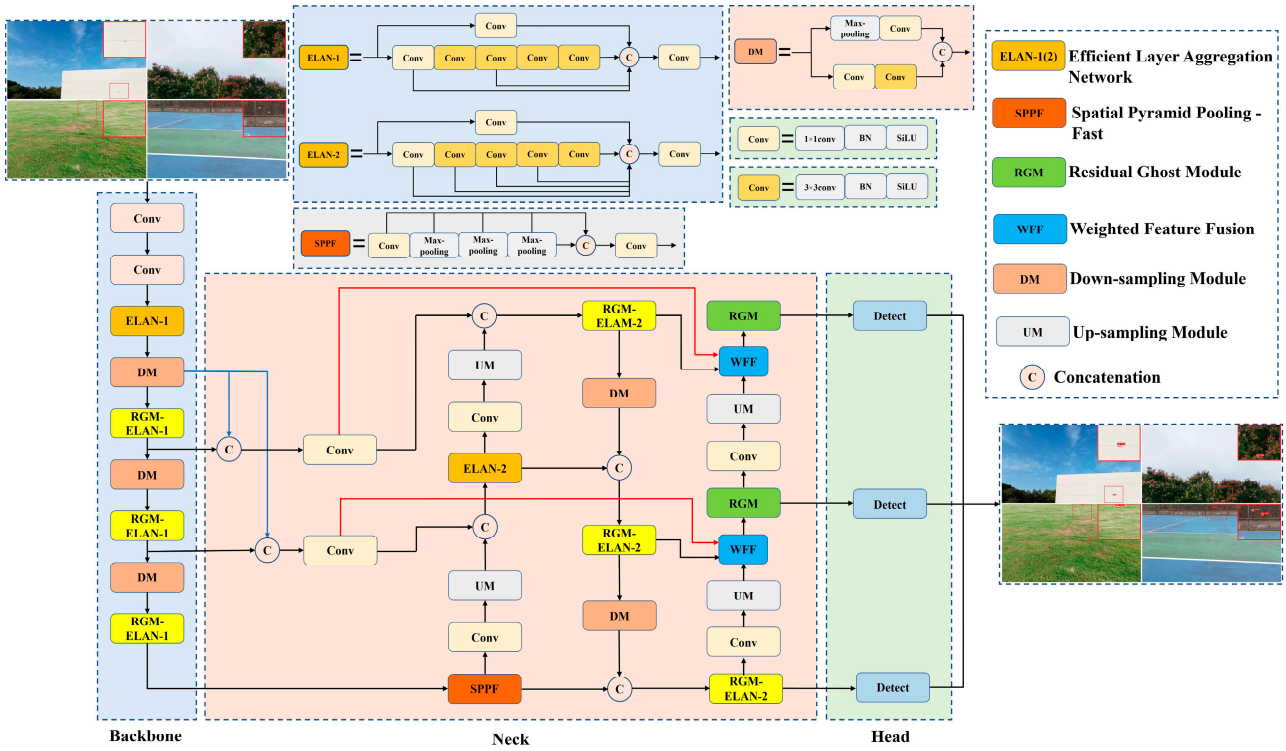
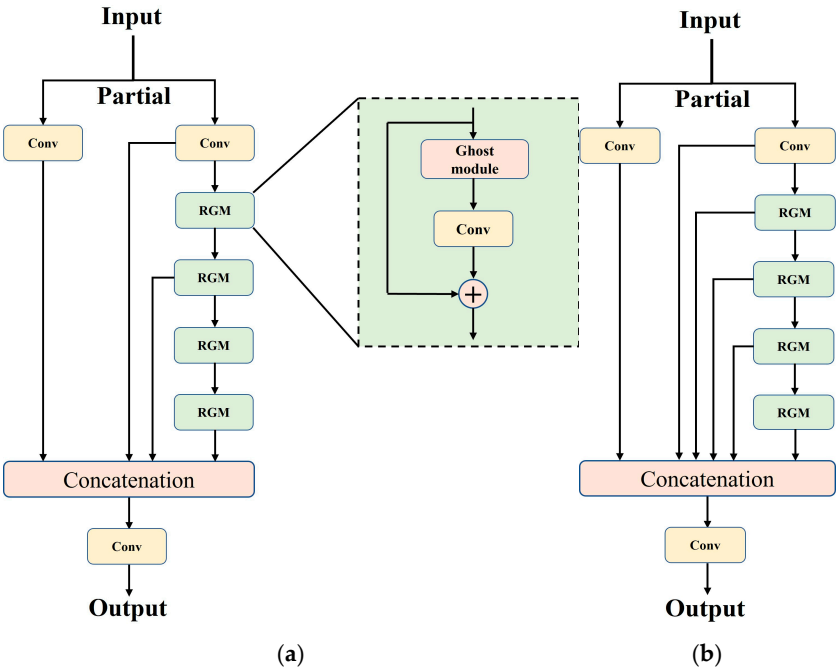


Figure 1. The overall structure of the proposed EWH-YOLO.

3.1. Efficient Feature Extraction Network

Feature extraction networks are used to obtain multi-level features for object detection. In recent years, many feature extraction networks are proposed to improve the detection accuracy. Due to parallelizing more gradient flow branches, the efficient layer aggregation network (ELAN) [14,41] has good feature extraction ability. However, it has high network parameters and computational complexity. For this reason, a residual Ghost module-based ELAN(RGM-ELAN) is proposed and utilized to construct an efficient feature extraction network. As depicted in Figure 2, the RGM-ELAN reduces network parameters and computational complexity by using RGM to replace 3×3 standard convolutions. The RGM is designed based on the Ghost module [42], standard convolution and residual connection [34]. The residual connection in RGM directly maps the original features learned by the previous layer to the subsequent layers without any loss, which effectively improves the degradation problem.



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

Figure 2. The detailed structure of RGM-ELAN. (**a**) RGM-ELAN-1. (**b**) RGM-ELAN-2.

As depicted in Figure 3, the implementation of the Ghost module can be split into two steps. First, the standard convolution is used to obtain intrinsic feature maps. Then, a series of cheap linear operations are applied to each intrinsic feature map to produce more output features. Assuming that the size of input feature maps is $C \times H \times W$, the size of output feature maps is $N \times H' \times W'$, and the size of the convolution kernel is $K \times K$, where H and W are the height and width of input feature maps, respectively, C is the number of input channels, H' and W' are the height and width of output feature maps, respectively, and N is the number of output channels. For standard convolution, the network parameters and floating-point operations (FLOPs) can be calculated as:

$$P_{SC} = N \times C \times K \times K \tag{1}$$

$$F_{SC} = N \times C \times H' \times W' \times K \times K \tag{2}$$

where P_{SC} represents the network parameters of standard convolution, and F_{SC} represents the FLOPs of standard convolution. For the Ghost module, the network parameters and FLOPs can be calculated as:

$$P_{GM} = \frac{N}{S} \times C \times K \times K + (S-1) \times \frac{N}{S} \times D \times D \tag{3}$$

$$F_{GM} = \frac{N}{S} \times C \times H' \times W' \times K \times K + (S-1) \times \frac{N}{S} \times H' \times W' \times D \times D \tag{4}$$

where P_{GM} represents the network parameters of the Ghost module, F_{GM} represents the FLOPs of the Ghost module, S is the number of Ghost features, and D is the kernel size of cheap linear operation. Therefore, the ratio of parameters and FLOPs can be calculated as:

$$R_P = \frac{P_{SC}}{P_{GM}} = \frac{N \times C \times K \times K}{\frac{N}{S} \times C \times K \times K + (S-1) \times \frac{N}{S} \times D \times D} \approx \frac{S \times C}{S + C - 1} \approx S \tag{5}$$

$$R_F = \frac{F_{SC}}{F_{GM}} = \frac{N \times C \times H' \times W' \times K \times K}{\frac{N}{S} \times C \times H' \times W' \times K \times K + (S-1) \times \frac{N}{S} \times H' \times W' \times D \times D} \approx \frac{S \times C}{S + C - 1} \approx S \tag{6}$$

where R_P represents the ratio of parameters, R_F represents the ratio of FLOPs, D is equal to K, the value of S is 2, and $S \ll C$. It can be seen that the parameters and FLOPs of standard convolution are approximately 2 times that of the Ghost module. Following [14], the down-sampling module used in the efficient feature extraction network consists of max-pooling and convolutions. As shown in Figure 1, the proposed feature extraction network can effectively extract multi-level features and reduce the model complexity.

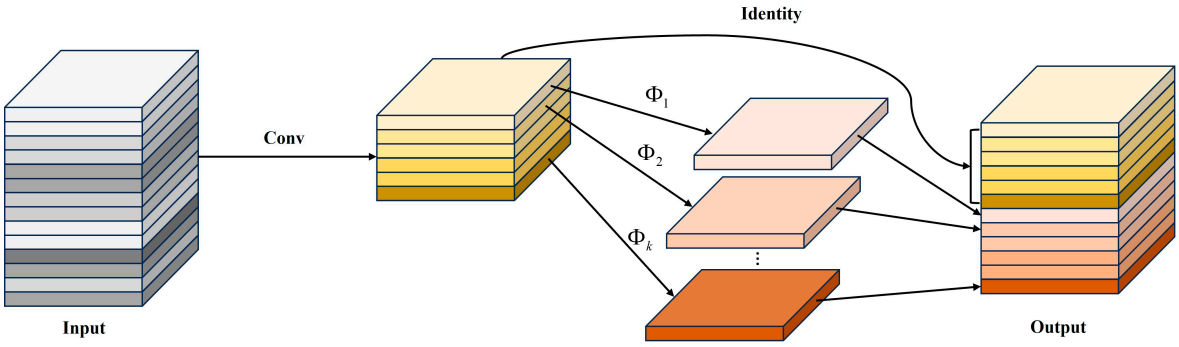


Figure 3. The detailed structure of the Ghost module.

3.2. Weighted Bidirectional Feature Fusion Network

In deep convolutional neural network, the fusion of multi-level features is an effective method to improve feature representation capability. Currently, PAN [37] is a representative feature fusion strategy and is widely used in object detection methods. However, PAN cannot effectively fuse the extracted feature information due to the limited feature information and poor visual appearance of small UAVs. Therefore, this paper proposes a weighted bidirectional feature fusion network (WBFFN) by enhancing the low-level and high-level features of small UAVs.

The detailed structure of the WBFFN is shown in Figure 4. It can be seen that the constructed network fuses multiscale features through bidirectional parallel feature fusion paths and skip connections. First, to enhance the low-level information of small UAVs, the shallower feature maps are introduced directly into the output feature maps of the backbone. Then, PAN is used to fuse the different level features. Moreover, considering that the high-level features have rich semantic information of small UAVs, a semantic information transmission operation is added from high-level feature maps to low-level feature maps. To deal with the loss of feature information, the skip connections are added between the input feature maps and output feature maps. In addition, a weighted feature fusion approach [43] is used to learn the importance of input feature maps. Taking F_2 in Figure 4 as an example, the feature fusion operation can be described as follows:

$$F_2 = \mathrm{Conv}\left(\frac{w_0 \cdot P_2 + w_1 \cdot N_2 + w_2 \cdot \mathrm{UM}(\mathrm{Conv}(N_3))}{w_0 + w_1 + w_2 + \varepsilon}\right) \tag{7}$$

where w_0, w_1 and w_2 are learnable weights, and ε is a small value used to prevent numerical instability. Compared with PAN, the proposed WBFFN can more fully integrate low-level and high-level information for detecting small UAVs.

3.3. Hybrid Bounding Box Regression Loss

During the training process, the bounding box regression loss is used to determine the position of the UAVs in the image. It is difficult to accurately locate the small UAVs during training due to low resolution in images. To improve the location learning ability and detection accuracy, a hybrid bounding box regression loss (HBBRL) is constructed based

on efficient intersection over union (EIoU) loss [44] and normalized Wasserstein distance (NWD) loss [45] for training the network.

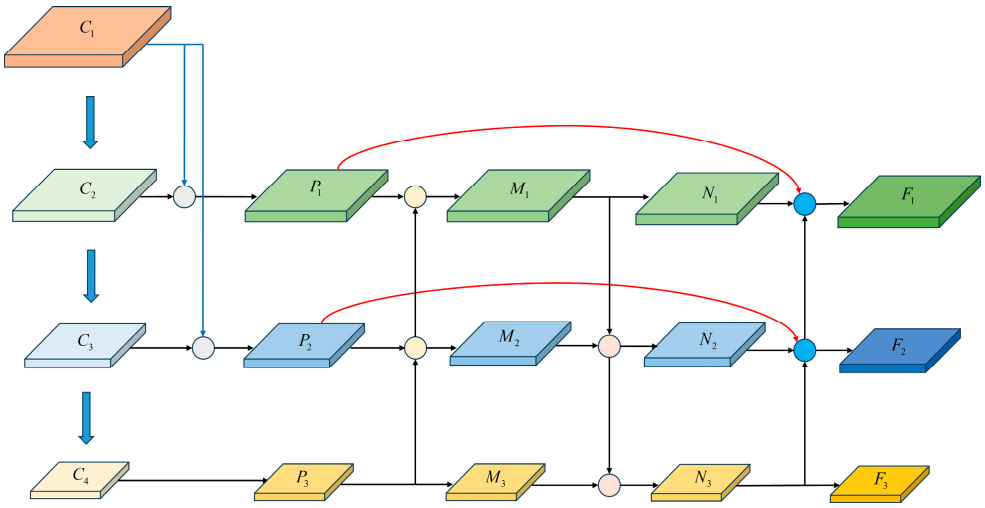


Figure 4. The detailed structure of the WBFFN.

As illustrated in Figure 5, B_p is the predicted bounding box, and B_{gt} is a ground-truth bounding box. w_p and h_p represent width and height of the predicted bounding box, respectively. w_{gt} and h_{gt} represent width and height of the ground-truth bounding box, respectively. w_c and h_c represent width and height of the smallest enclosing box covering the predicted bounding box and the ground-truth bounding box, respectively. b_p and b_{gt} are the central points of the predicted bounding box and the ground-truth bounding box, respectively. The EIoU loss is defined as follows:

$$L_{EIOU} = 1 - \mathrm{IoU} + \frac{\rho^2(b_p, b_{gt})}{(w_c)^2 + (h_c)^2} + \frac{\rho^2(w_p, w_{gt})}{(w_c)^2} + \frac{\rho^2(h_p, h_{gt})}{(h_c)^2} \tag{8}$$

$$\mathrm{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \tag{9}$$

where $\rho(\cdot)$ represents the Euclidean distance. The EIoU can improve the accuracy of bounding box regression during training since it contains three geometric factors including the overlap area, the central point, and the side length.

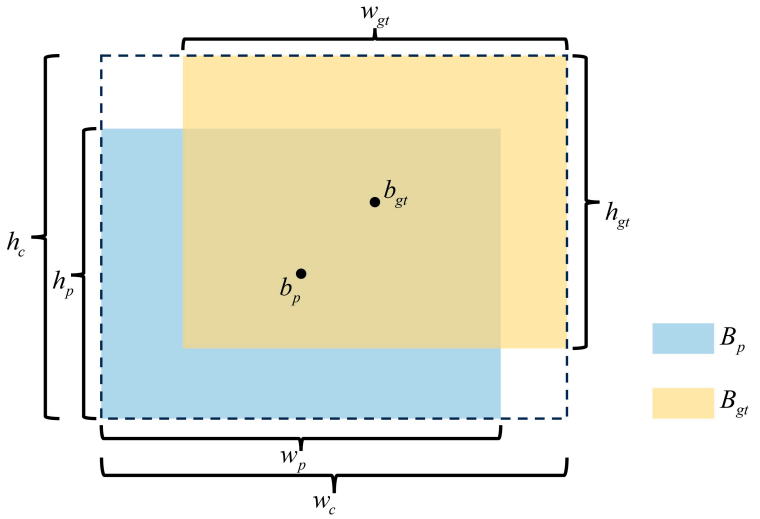


Figure 5. Illustration of EIoU loss.

Since IOU-based loss evaluates similarity by calculating the geometric overlap, the small object bounding boxes are more sensitive to the position deviation. In contrast, the NWD loss models bounding boxes into a two-dimensional (2D) Gaussian distribution and adopts the Wasserstein distance to evaluate the difference between bounding boxes. Given that the NWD loss is less sensitive to the scale changes in bounding boxes, it is more suitable for evaluating the similarity between small bounding boxes. Therefore, this paper introduces NWD loss to further improve the localization accuracy of small bounding boxes. The calculation formula is described as follows:

$$L_{NWD} = 1 - \exp\left(-\frac{\sqrt{W_2^2(N_p, N_{gt})}}{E}\right) \tag{10}$$

$$W_2^2(N_p, N_{gt}) = \left\| [cx_p, cy_p, \frac{w_p}{2}, \frac{h_p}{2}]^{\mathrm{T}}, [cx_{gt}, cy_{gt}, \frac{w_{gt}}{2}, \frac{h_{gt}}{2}]^{\mathrm{T}} \right\|_2^2 \tag{11}$$

where $W_2^2(N_p, N_{gt})$ is a distance metric, and E is a constant. N_p and N_{gt} represent two 2D Gaussian distributions which are modeled from the predicted bounding box and ground-truth bounding box, respectively. (cx_p, cy_p) and (cx_{gt}, cy_{gt}) are the center coordinates.

A size-aware sigmoid weighted summation [46] is adopted to combine the EIoU loss and NWD loss. Finally, the HBBRL can be computed as follows:

$$L_H = \frac{1}{1 + e^{-(\sqrt{A}/L - 1)}} \times L_{EIoU} + \left(1 - \frac{1}{1 + e^{-(\sqrt{A}/L - 1)}}\right) \times L_{NWD} \tag{12}$$

where A represents the area of the ground-truth bounding box, and L represents a constant.

4. Experimental Results and Analysis

4.1. Dataset Description

A new dataset is collected and annotated on the basis of considering small UAVs to evaluate the EWH-YOLO. The dataset contains 3900 images in total, with 3120 training images, and 780 test images. During the dataset construction process, we remove images that do not contain UAVs and duplicate images. Therefore, there are no empty background images in the dataset and each image contains either one or two annotated instances. All images are manually labeled by labelImg [47]. Some samples in the dataset are shown in Figure 6. The object size distribution of the self-constructed dataset is shown in Figure 7. The self-constructed dataset contains images of small UAVs with pixel areas less than 32×32 pixels [48] in various practical scenarios.

4.2. Implementation Details

All our experiments are conducted on Pytorch [49] deep learning framework with an NVIDIA Geforce GTX 3060 GPU card. In the experiments, the size of the input images is set to 576×576, The stochastic gradient descent (SGD) optimizer is employed to train the EWH-YOLO. The initial learning rate, weight decay and momentum factor are set to 0.01, 0.0005 and 0.937. ε is set to 0.0001. The constants E and L are set to 32. Data augmentation methods are adopted, including mosaic augmentation with a probability of 1.0, HSV augmentation with hsv_h = 0.015, hsv_s = 0.7, and hsv_v = 0.4, scale = 0.5, translation = 0.1, and fliplr = 0.5. The network is trained for 100 epochs with a batch size of 8. YOLO11 is adopted as the baseline.



Figure 6. Some samples of the self-constructed dataset.

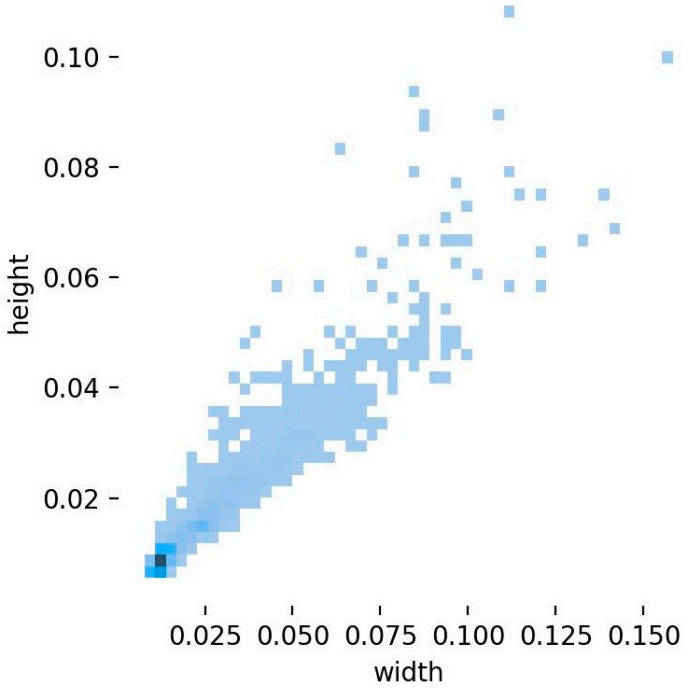


Figure 7. The object size distribution of the self-constructed dataset.

4.3. Evaluation Metrics

The average precision (AP) is employed to evaluate detection accuracy. The formulas for the calculation of AP are as follows:

$$\mathrm{AP} = \int_0^1 PdR \tag{13}$$

$$P = \frac{TP}{TP + FP} \times 100\% \quad (14)$$

$$R = \frac{TP}{TP + FN} \times 100\% \quad (15)$$

where TP, FP and FN refer to the number of true positives, false positives, and false negatives, respectively. P and R are precision and recall, respectively. Meanwhile, model complexity is evaluated by network parameters, model size and FLOPs, and detection speed is evaluated by frames per second (FPS).

4.4. Comparisons with State-of-the-Art Methods

The proposed EWH-YOLO is compared with state-of-the-art object detection methods, including GELAN [15], YOLOv9 [15], YOLOv10 [16], YOLO11 [17] and HyperYOLO [18] and YOLO-Master [19]. Table 1 shows the quantitative comparison of the EWH-YOLO with other detection methods. As presented in Table 1, $AP_{50:95}$ and AP_{50} of the EWH-YOLO are 50.8% and 81.8%, respectively, which achieves the highest detection accuracy. Specifically, compared with YOLO11-S, the EWH-YOLO achieves a 4.5% improvement in $AP_{50:95}$ and a 4.2% improvement in AP_{50}. Compared with GELAN-S, the proposed EWH-YOLO obtains an improvement of 5.6% and an improvement of 6.7% in $AP_{50:95}$ and AP_{50}. Compared with Hyper-YOLO, the EWH-YOLO improves $AP_{50:95}$ and AP_{50} by 3.9% and 3.3%, respectively. Figure 7 provides the PR curves of different methods. As shown in Figure 8, the area surrounded by the PR curve of the EWH-YOLO is the largest. Therefore, the EWH-YOLO significantly enhances the detection accuracy of small UAVs.

Table 1. Comparison of different methods on the self-constructed dataset.

Methods	$AP_{50:95}$ (%)	AP_{50} (%)	Parameters (M)	Model Size (MB)	FLOPs (G)
GELAN-S	45.2	75.1	7.1	14.3	21.2
YOLOv9-S	45.9	75.6	7.1	14.3	21.2
YOLOv10-S	44.3	76.4	8.0	15.7	19.8
YOLO11-S	46.3	77.6	9.4	18.3	17.3
HyperYOLO-S	46.9	78.5	14.8	28.6	31.5
YOLO-Master-S	47.1	78.1	9.7	18.9	23.4
Ours	50.8	81.8	6.4	12.6	17.2

In terms of model complexity, Table 1 lists the network parameters, model size and FLOPs of the EWH-YOLO and other comparison methods. As presented in Table 1, compared with YOLO11-S, the EWH-YOLO reduces the parameters and model sizes by 31.9% and 31.1%, respectively, and the FLOPs is slightly lower. Compared with GELAN-S, the parameters, model size and FLOPs of the EWH-YOLO are decreased by 9.9%, 11.9% and 18.9%, respectively. Compared with Hyper-YOLO, the parameters, model size and FLOPs of the EWH-YOLO are decreased by 56.8%, 55.9% and 45.4%, respectively. Figure 9 provides the comparison of $AP_{50:95}$ with network parameters, model sizes and FLOPs. It can be seen that the EWH-YOLO achieves the highest $AP_{50:95}$ while using the lowest parameters and FLOPs. Overall, the EWH-YOLO has superior detection performance in terms of detection accuracy and model complexity.

Figure 10 presents the visualization results of the baseline and the EWH-YOLO. It can be observed that the baseline fails to detect the small UAVs under some practical scenarios. For example, in the first row, the small UAV is missed by the baseline. In contrast, the EWH-YOLO is able to detect all of the small UAVs. Figure 11 illustrates more detection results of the EWH-YOLO. The visualization results demonstrate that the EWH-YOLO effectively detects the small UAVs under various scenarios and reduces missed detections.

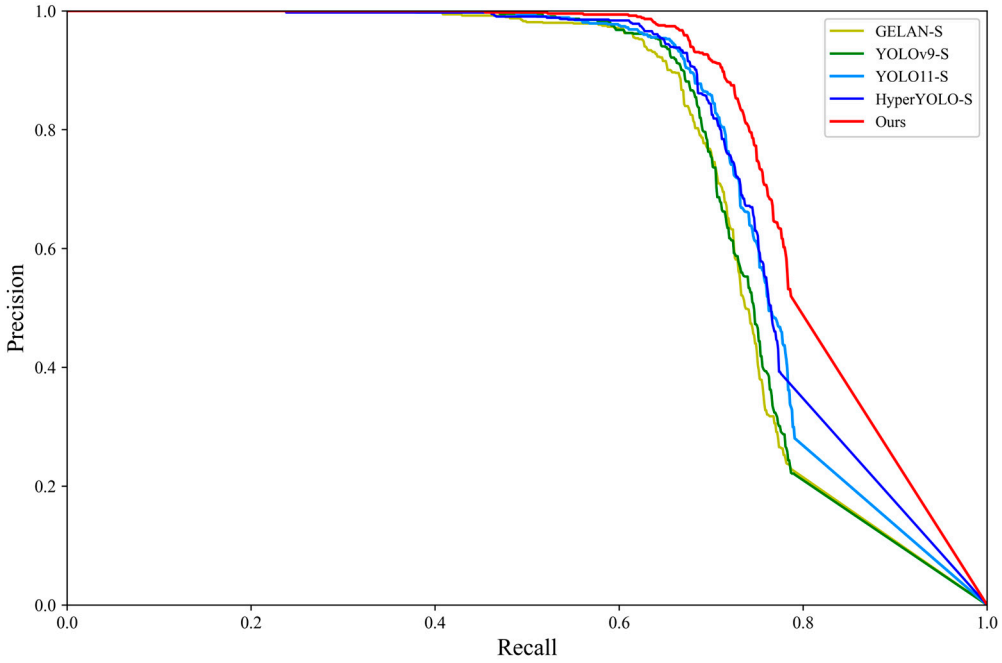


Figure 8. PR curves of GELAN-S, YOLOv9-S, YOLO11-S, HyperYOLO-S and the EWH-YOLO.

To further evaluate the detection performance of the EWH-YOLO, a public dataset [50] containing small UAVs is adopted to train and test the EWH-YOLO. Table 2 presents the detection results of different methods. Experimental results on the public dataset show that EWH-YOLO achieved the highest detection accuracy. As shown in Figure 12, the EWH-YOLO effectively detects the small UAVs in the images. In general, the proposed EWH-YOLO significantly improves the detection performance on the public dataset.

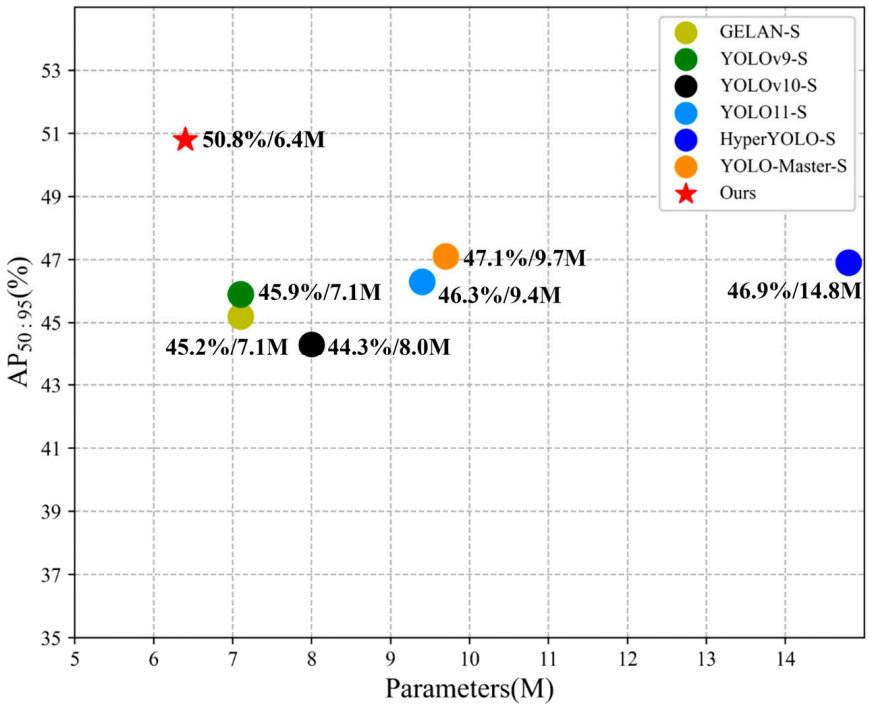


Figure 9. *Cont.*

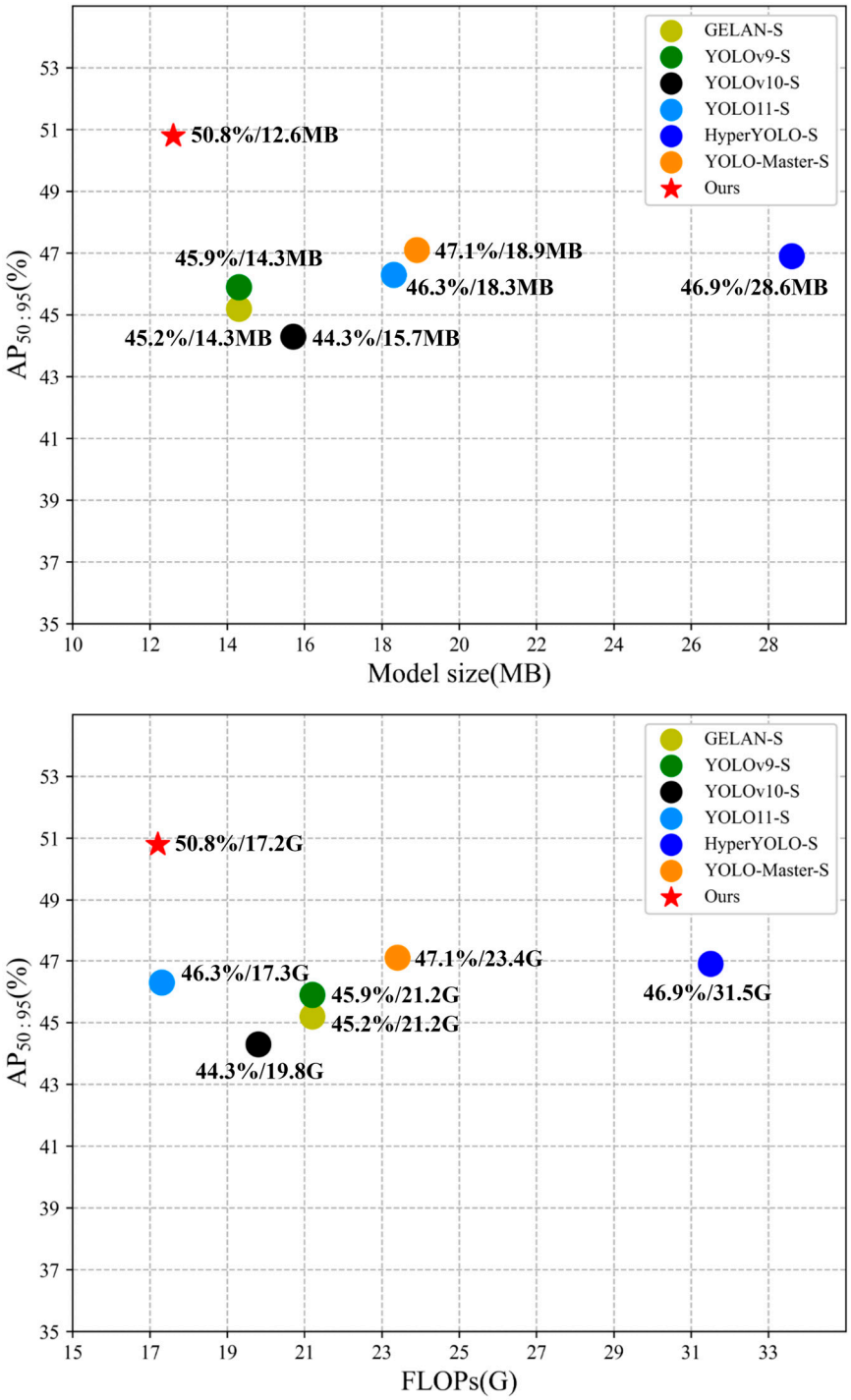


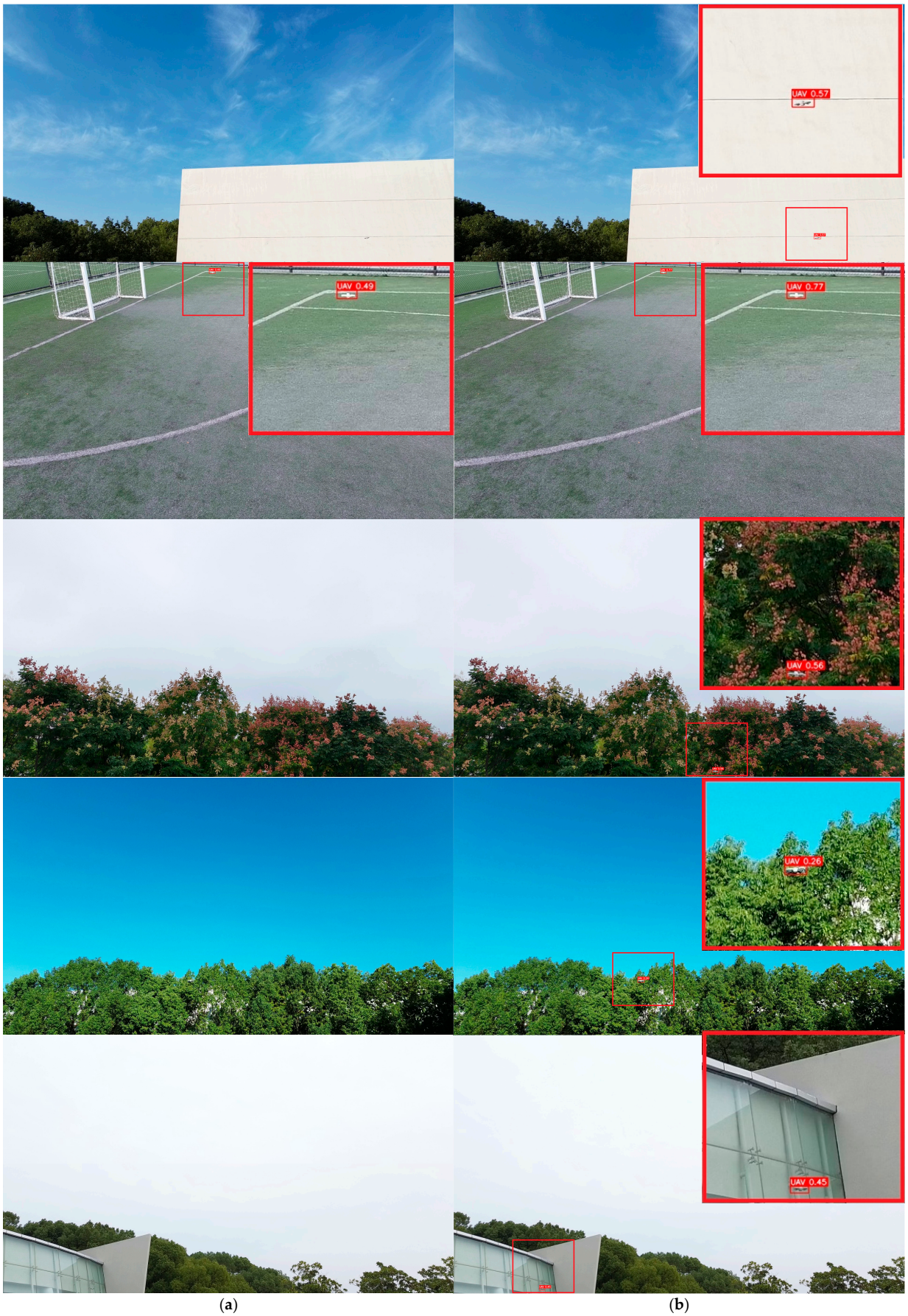
Figure 9. Comparison of $AP_{50:95}$ with parameters, model size and FLOPs.

Table 2. Comparison of different methods on the public dataset.

Methods	AP_{50} (%)	Parameters (M)	Model Size (MB)	FLOPs (G)
YOLO11-S	74.5	9.4	18.3	17.3
YOLO-Master-S	75.4	9.7	18.9	23.4
Ours	79.9	6.4	12.6	17.2

4.5. Ablation Experiments

To evaluate the effectiveness of each proposed module, ablation experiments are conducted on the created UAV dataset. The proposed EFEN, WBFFN and HBBRL are gradually introduced to the baseline.



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

Figure 10. Detection results of the baseline and the EWH-YOLO. (**a**) Detection results of the baseline. (**b**) Detection results of the EWH-YOLO.

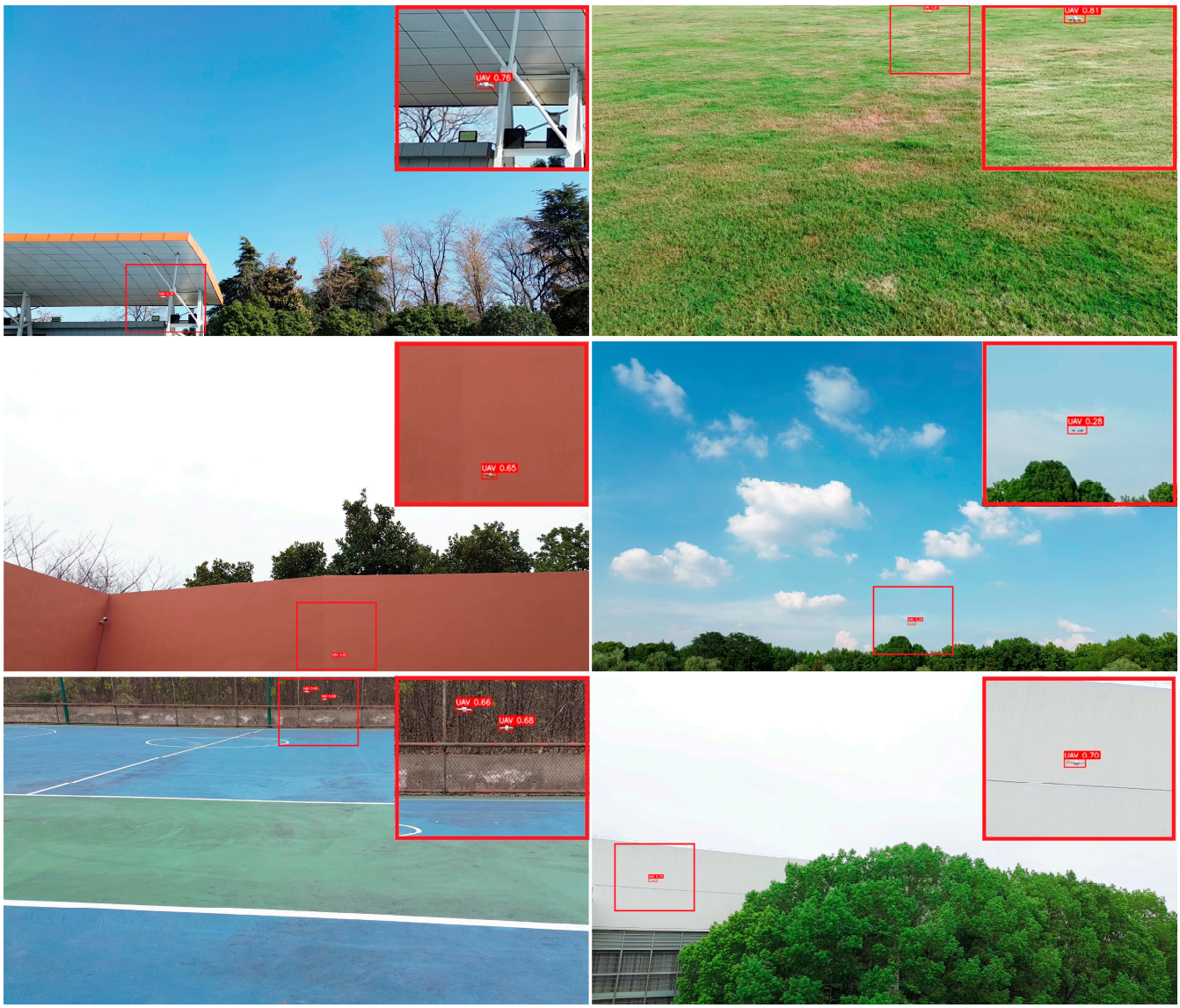


Figure 11. More detection results.

Table 3 shows the specific results of ablation experiments. First, the EFEN is adopted as the feature extraction network. According to Table 3, when adding the proposed EFEN to the baseline, the $AP_{50:95}$ increases from 46.3% to 47.3%, and the AP_{50} increases from 77.6% to 79.1%. The parameters decrease from 9.4M to 5.7M and the FPS decreases from 256 to 244. Then, the WBFFN is introduced to the baseline. After adding the WBFFN, the $AP_{50:95}$ increases from 47.3% to 48.7%, and the AP_{50} increases from 79.1% to 80.0%. The parameters increase from 5.7M to 6.4M and the FPS decreases from 244 to 213. Finally, the impact of the HBBRL is studied. When incorporating the HBBRL, the $AP_{50:95}$ increases from 48.7% to 50.8%, the AP_{50} increases from 80.0% to 81.8%, and the parameters and FPS remain unchanged. Despite a slight drop in FPS, the proposed EWH-YOLO effectively improves detection accuracy and reduces the network parameters.

In general, the results of ablation experiments confirm that each of the three proposed modules contributes positively to the baseline. Functionally, the EFEN is designed to extract multi-level features required for small UAV detection. The WBFFN is adopted to effectively integrate multi-level features. The HBBRL is used to enhance the location learning ability. In addition, the ablation results show steady performance growth when

adding them incrementally. Therefore, the three proposed modules are complementary, and their combination achieves synergistic improvements without performance conflicts.



Figure 12. Detection results on the public dataset.

Table 3. The results of ablation experiments.

Baseline	EFEN	WBFFN	HBBRL	$AP_{50:95}$ (%)	AP_{50} (%)	Parameters (M)	FPS
√				46.3	77.6	9.4	256
√	√			47.3	79.1	5.7	244
√	√	√		48.7	80.0	6.4	213
√	√	√	√	50.8	81.8	6.4	213

To further validate the performance of the EFEN, we compare the proposed EFEN with other feature extraction networks, including ELAN, C3k2, and C3k2-Ghost. Table 4 presents the detection results of different feature extraction networks. The results indicate that the EFEN achieves a more balanced detection performance.

Table 4. Comparison of different feature extraction networks.

Methods	$AP_{50:95}$ (%)	Parameters (M)
C3k2(YOLO11)	46.3	9.4
C3k2-Ghost	45.2	7.8
ELAN	47.2	7.7
EFEN	47.3	5.7

To further validate the performance of the HBBRL, we compare the HBBRL with other bounding box regression losses. As shown in Table 5, the HBBRL achieves better detection accuracy. The results indicate that the HBBRL effectively improves the detection accuracy of small UAVs.

Table 5. Comparison of different bounding box regression losses.

Methods	$AP_{50:95}$ (%)	Parameters (M)
CIoU	48.7	6.4
DIoU	48.5	6.4
EIoU	49.1	6.4
HBBRL	50.8	6.4

5. Conclusions

For the detection of small UAVs, this paper proposes a novel deep convolutional neural network with efficient feature extraction network, weighted bidirectional feature fusion network and hybrid bounding box regression loss. To begin with, an efficient feature extraction network is constructed based on the RGM-ELAN. It effectively obtains the multi-level features of small UAVs while reducing network parameters and computational complexity. Then, a weighted bidirectional feature fusion network is proposed to obtain more representative low-level and high-level features in the output feature maps. Moreover, a hybrid bounding box regression loss is adopted to evaluate the difference between the true bounding box and the predicted bounding box during training and enhance the location learning ability of small UAVs. Additionally, a self-constructed dataset containing small UAVs in various practical scenarios is used to verify the detection performance. The experimental results show that the EWH-YOLO achieves competitive advantages over state-of-the-art detectors. In future work, we plan to deploy the EWH-YOLO on the embedded platforms.

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

Funding: This work was funded by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (grant number KYCX23_0389).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
R-CNN	Region-based Convolutional Neural Network
YOLO	You Only Look Once
EFEN	Efficient Feature Extraction Network
WBFFN	Weighted Bidirectional Feature Fusion Network
HBBRL	Hybrid Bounding Box Regression Loss
RGM	Residual Ghost Module
WFF	Weighted Feature Fusion
DM	Down-sampling Module
UM	Up-sampling Module
SPPF	Spatial Pyramid Pooling-Fast
ELAN	Efficient Layer Aggregation Network
FLOPs	Floating-point Operations
PAN	Path Aggregation Network
EIoU	Efficient Intersection over Union
NWD	Normalized Wasserstein Distance

SGD	Stochastic Gradient Descent
AP	Average Precision

References

1. Liu, R.; Hu, W.; Pedrycz, W.; Song, Y.; Xing, L.; Yu, Y. A Reinforcement Learning-Based Artificial Bee Colony Algorithm for Resolving the UAV Logistics Path Planning Problem in a 3D Space. *Swarm Evol. Comput.* **2026**, *102*, 102328. [CrossRef]
2. Joshi, P.; Sandhu, K.S.; Singh Dhillon, G.; Chen, J.; Bohara, K. Detection and Monitoring Wheat Diseases Using Unmanned Aerial Vehicles (UAVs). *Comput. Electron. Agric.* **2024**, *224*, 109158. [CrossRef]
3. Gao, F.; Liu, J.; Feng, Q.; Feng, R.; Yu, B. Task Allocation and Route Planning for UAV-Based Hybrid Inspection of Transmission Towers, Substations, and Distribution Poles from a Fixed Nest. *Expert Syst. Appl.* **2026**, *313*, 131625. [CrossRef]
4. Kamath, V.; Renuka, A. Deep Learning Based Object Detection for Resource Constrained Devices: Systematic Review, Future Trends and Challenges Ahead. *Neurocomputing* **2023**, *531*, 34–60. [CrossRef]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Columbus, OH, USA, 2014; pp. 580–587.
6. Girshick, R. Fast R-CNN. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*; IEEE: Santiago, Chile, 2015; pp. 1440–1448.
7. 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]
8. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Honolulu, HI, USA, 2017; pp. 936–944.
9. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards Balanced Learning for Object Detection. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Long Beach, CA, USA, 2019; pp. 821–830.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Las Vegas, NV, USA, 2016; pp. 779–788.
11. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Honolulu, HI, USA, 2017; pp. 6517–6525.
12. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
13. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
14. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Vancouver, BC, Canada, 2023; pp. 7464–7475.
15. Wang, C.-Y.; Yeh, I.-H.; Mark Liao, H.-Y. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *Computer Vision—ECCV 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer Nature: Cham, Switzerland, 2025; Volume 15089, pp. 1–21, ISBN 978-3-031-72750-4.
16. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011. [CrossRef]
17. Jocher, G. Ultralytics YOLO11. 2024. Available online: https://github.com/ultralytics (accessed on 2 September 2026).
18. Feng, Y.; Huang, J.; Du, S.; Ying, S.; Yong, J.-H.; Li, Y.; Ding, G.; Ji, R.; Gao, Y. Hyper-YOLO: When Visual Object Detection Meets Hypergraph Computation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 2388–2401. [CrossRef] [PubMed]
19. Lin, X.; Peng, J.; Gan, Z.; Zhu, J.; Liu, J. YOLO-Master: MOE-Accelerated with Specialized Transformers for Enhanced Real-Time Detection. *arXiv* **2025**, arXiv:2512.23273.
20. Huang, L.; Qiu, M.; Xu, A.; Sun, Y.; Zhu, J. UAV Imagery for Automatic Multi-Element Recognition and Detection of Road Traffic Elements. *Aerospace* **2022**, *9*, 198. [CrossRef]
21. Ojdanić, D.; Naverschnigg, C.; Sinn, A.; Zelinskyi, D.; Schitter, G. Parallel Architecture for Low Latency UAV Detection and Tracking Using Robotic Telescopes. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 5515–5524. [CrossRef]
22. Zhang, H.; Zhang, W.; Wang, W.; Li, X.; Zhang, A. Research on Traffic Sign Detection Algorithm Based on Improved SSD in Complex Environments. *Meas. Sci. Technol.* **2024**, *35*, 115404. [CrossRef]
23. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*; IEEE: Kauai, HI, USA, 2001; Volume 1.
24. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*; IEEE: San Diego, CA, USA, 2005; Volume 1, pp. 886–893.
25. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]

26. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
27. 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 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*; IEEE: Daejeon, Republic of Korea, 2016; pp. 1556–1561.
28. Yang, R.; Shan, P.; He, Y.; Xiao, H.; Zhang, L.; Zhao, Y.; Fu, Q. A Lightweight Bionic Flapping Wing Drone Recognition Network Based on Data Enhancement. *Measurement* **2025**, *239*, 115476. [CrossRef]
29. Arsenos, A.; Karampinis, V.; Petrongonas, E.; Skliros, C.; Kollias, D.; Kollias, S.; Voulodimos, A. Common Corruptions for Evaluating and Enhancing Robustness in Air-to-Air Visual Object Detection. *IEEE Robot. Autom. Lett.* **2024**, *9*, 6688–6695. [CrossRef]
30. Lee, Z.W.; Chin, W.H.; Ho, H.W. Air-to-Air Micro Air Vehicle Interceptor with an Embedded Mechanism and Deep Learning. *Aerosp. Sci. Technol.* **2023**, *135*, 108192. [CrossRef]
31. Opromolla, R.; Fasano, G. Visual-Based Obstacle Detection and Tracking, and Conflict Detection for Small UAS Sense and Avoid. *Aerosp. Sci. Technol.* **2021**, *119*, 107167. [CrossRef]
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
33. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Las Vegas, NV, USA, 2016; pp. 770–778.
35. Wang, C.-Y.; Mark Liao, H.-Y.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*; IEEE: Seattle, WA, USA, 2020; pp. 1571–1580.
36. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2016; Volume 9905, pp. 21–37, ISBN 978-3-319-46447-3.
37. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE: Salt Lake City, UT, USA, 2018; pp. 8759–8768.
38. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. UnitBox: An Advanced Object Detection Network. In *Proceedings of the 24th ACM International Conference on Multimedia*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 516–520.
39. 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)*; IEEE: Long Beach, CA, USA, 2019; pp. 658–666.
40. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000. [CrossRef]
41. Wang, C.-Y.; Liao, H.-Y.M.; Yeh, I.-H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
42. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Seattle, WA, USA, 2020; pp. 1577–1586.
43. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Seattle, WA, USA, 2020; pp. 10778–10787.
44. Zhang, Y.-F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and Efficient IOU Loss for Accurate Bounding Box Regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
45. Wang, J.; Xu, C.; Yang, W.; Yu, L. A Normalized Gaussian Wasserstein Distance for Tiny Object Detection. *arXiv* **2022**, arXiv:2110.13389.
46. Ying, X.; Xiao, C.; An, W.; Li, R.; He, X.; Li, B.; Cao, X.; Li, Z.; Wang, Y.; Hu, M.; et al. Visible-Thermal Tiny Object Detection: A Benchmark Dataset and Baselines. *IEEE Trans. Pattern Anal. Mach. Intell.* **2025**, *47*, 6088–6096. [CrossRef] [PubMed]
47. Tzutalin. LabelImg. 2015. Available online: https://github.com/tzutalin/labelImg (accessed on 2 September 2026).
48. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision (ECCV 2014), Zurich, Switzerland, 6–12 September 2014*; Springer: Cham, Switzerland, 2015; pp. 740–755.

49. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*; Curran Associates: Red Hook, NY, USA, 2019.
50. Sun, H.; Yang, J.; Shen, J.; Liang, D.; Ning-Zhong, L.; Zhou, H. TIB-Net: Drone Detection Network With Tiny Iterative Backbone. *IEEE Access* **2020**, *8*, 130697–130707. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.