algorithms

MDPI

Article

Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery

Linhua Zhang [1], Ning Xiong [2], Xinghao Pan [3], Xiaodong Yue [4], Peng Wu [5,*] and Caiping Guo [1]

[1] Department of Computer Engineering, Taiyuan Institute of Technology, Taiyuan 030008, China; zhanglh@tit.edu.cn (L.Z.); guocaiping@tit.edu.cn (C.G.)
[2] School of Innovation, Design and Engineering, Malardalen University, 72123 Vasteras, Sweden; ning.xiong@mdh.se
[3] School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, China; panxinghaovip@163.com
[4] Artificial Intelligence Institute of Shanghai University, Shanghai University, Shanghai 200444, China; yswantfly@shu.edu.cn
[5] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: 14112078@bjtu.edu.cn

Abstract: In unmanned aerial vehicle photographs, object detection algorithms encounter challenges in enhancing both speed and accuracy for objects of different sizes, primarily due to complex backgrounds and small objects. This study introduces the PDWT-YOLO algorithm, based on the YOLOv7-tiny model, to improve the effectiveness of object detection across all sizes. The proposed method enhances the detection of small objects by incorporating a dedicated small-object detection layer, while reducing the conflict between classification and regression tasks through the replacement of the YOLOv7-tiny model's detection head (IDetect) with a decoupled head. Moreover, network convergence is accelerated, and regression accuracy is improved by replacing the Complete Intersection over Union (CIoU) loss function with a Wise Intersection over Union (WIoU) focusing mechanism in the loss function. To assess the proposed model's effectiveness, it was trained and tested on the VisDrone-2019 dataset comprising images captured by various drones across diverse scenarios, weather conditions, and lighting conditions. The experiments show that mAP@0.5:0.95 and mAP@0.5 increased by 5% and 6.7%, respectively, with acceptable running speed compared with the original YOLOv7-tiny model. Furthermore, this method shows improvement over other datasets, confirming that PDWT-YOLO is effective for multiscale object detection.

Keywords: small-object detection; decoupled head; WIoU; YOLOv7-tiny model

Citation: Zhang, L.; Xiong, N.; Pan, X.; Yue, X.; Wu, P.; Guo, C. Improved Object Detection Method Utilizing YOLOv7-Tiny for Unmanned Aerial Vehicle Photographic Imagery. *Algorithms* **2023**, *16*, 520. https://doi.org/10.3390/a16110520

Academic Editor: Frank Werner

Received: 20 October 2023
Revised: 11 November 2023
Accepted: 13 November 2023
Published: 14 November 2023

1. Introduction

Object detection [1] is an important task in the field of computer vision which aims to identify and locate target objects automatically in images or videos. Unlike simple classification tasks, object detection requires identifying multitargets in the image and providing a bounding box for each target to indicate its precise location. Object detection technology has been widely used in fields such as self-driving cars, video surveillance, medical image analysis, and drone image analysis [2]. In these application fields, unmanned aerial vehicle (UAV) object detection has received widespread attention in recent years, and has a significant impact on both military and civilian applications [3]. However, complex scenes, variably sized small objects, occlusions, and variable illumination bring a variety of challenges and requirements to UAV object detection technology. Therefore, research on this technology is of considerable significance in terms of improving the real-time accuracy of UAV object detection.

Recently, the rapid progress in deep learning, as indicated in [4], has greatly expedited its extensive adoption in UAV object detection. Deep learning models can be broadly

classified into one- or two-stage approaches, a categorization based on their algorithmic structures.

The most classic two-stage models are the region-based convolutional neural network (R-CNN) algorithm series. In 2014, Girshick et al. [5] introduced the R-CNN, an algorithmic method of region selection, followed by classification and detection. In 2015, Girshick [6] improved the R-CNN and proposed a Fast R-CNN. And Ren et al. [7] made a further improvement and introduced Faster R-CNN. This further improved detection accuracy and efficiency. In 2018, Zhao et al. [8] introduced Cascade R-CNN.

Conversely, one-stage models do not require candidate region selection. These end-to-end algorithms treat object detection as a direct regression task. Representative algorithms include the You Only Look Once (YOLO) series [9–19], the Single Shot MultiBox Detector (SSD) [20], and RetinaNet [21]. These algorithms have fewer parameters, and the detection processes are considerably shorter; nevertheless, they tend to have a slightly lower accuracy than two-stage models.

With continuous model development, object detection methods have evolved to meet various scene-specific needs. However, achieving real-time and highly accurate detection remains challenging, particularly because of the high flight speeds of UAVs. To address this issue, Zhang et al. [22] proposed a UAV detection method based on YOLOv3 and a pruning algorithm. These measures were aimed at enhancing the detection speed. The proposed UAV detection model achieved a significant reduction in model size of 95.16% and inference time of 51.37%. However, this optimization did not improve the accuracy. Wang et al. [23] proposed replacing Visual Geometry Group 16 (VGG-16) with Residual Network (ResNet) specifically for UAV vehicle detection scenarios. The optimized Faster R-CNN model was effective in handling small-object vehicles while reducing false alarms and missed detections. This led to an impressive vehicle detection accuracy of 96.83%. Nonetheless, this improvement in accuracy came at the expense of decreasing the detection speed. Huang et al. [24] proposed an optimized cascaded R-CNN. This enhancement involved adding superclass detection to the original algorithm. Additionally, it incorporated the fusion of regression confidence and modifications to the loss function, all aimed at enhancing object detection capability. This method effectively improves the detection performance for aerial targets. However, because of the large baseline, the model was limited by slow reasoning speed and weak mobility.

In this study, to address the challenge of detecting small objects with complex backgrounds while satisfying the high-speed detection requirements of edge computing, the PDWT-YOLO algorithm was introduced as an improvement to YOLOv7-tiny. Experiments were conducted using the public dataset VisDrone-2019 [25], and the results confirmed that the proposed algorithm enhanced detection accuracy while simultaneously maintaining detection speed. The model shows the following improvements compared with YOLOv7-tiny:

- A small-object detection layer was added that used a 160 $\times$ 160 resolution feature map to detect small objects. This adjustment enhanced the detection performance for small objects.
- The contradiction between small-object classification and regression was weakened by introducing a decoupled head [26] to replace the detection head IDetect [27] in YOLOv7-tiny, which improved detection accuracy.
- The WIoU loss function [28] that expedites network convergence and enhances the regression accuracy was used instead of CIoU thus presenting a balanced regression approach for both high- and low-quality samples. Compared with CIoU [29], WIoU emphasizes anchor boxes of average quality, resulting in an overall performance enhancement for the detector. This function is suited to handling small object boxes and overlapping occluded object boxes, making it beneficial for small-object detection.

2. Related Work

In this paper, after fully studying and analyzing the structure of YOLOv7-tiny and various improved algorithms, we put forward an improved object detection method based on YOLOv7-tiny. In Section 2.1, we introduce the development of YOLO series algorithms and the structure of the YOLOv7-tiny network. In most of the object detection methods based on deep learning, the detection head and the IoU loss function are also two important parts. They are respectively used for classification, positioning, and measuring the overlap between the prediction box and the real box. Hence, we introduce the mainstream target detection heads and their respective advantages and disadvantages in Section 2.2, and various IoU loss functions in Section 2.3.

2.1. YOLOv7-Tiny Network Structure

YOLO is an object detection model that changes the traditional approach to object detection. It applies a single CNN [30] to process the entire image, divides the image into grids, and makes predictions. These predictions include the class probabilities and bounding box coordinates for any object present within each grid cell. The method enables YOLO to efficiently and accurately detect objects in images in real time.

YOLOv1 [9] was introduced in 2016 by Redmon et al. It is an end-to-end object recognition and detection method that predicts the probabilities of object categories in a complete image. In 2017, Redmon and team unveiled YOLOv2 [10], which improved on some shortcomings of YOLOv1 and significantly improved its accuracy and number of object detections. YOLOv2 adopted the network structure of DarkNet-19 [31] instead of the GoogLeNet [32] network structure of YOLOv1. The network structure had no fully connected layers, and there were five down samplings, all of which were convolution operations. The 1×1 convolution operation saved the parameters. YOLOv3 [11], introduced by Redmon et al. in 2018, represented a significant advancement in the YOLO series. Compared with its predecessors, YOLOv3 placed a primary emphasis on network architecture improvements, one of which was the adoption of the DarkNet-53 network structure, which was deeper and more complex than the earlier structures. This structural enhancement simultaneously improved the speed and accuracy of object detection. YOLOv3 introduced the idea of ResNet [33], stacking more layers for feature extraction and using spatial pyramid pooling networks [34] to achieve multi-size input and same-size output. In April 2020, Bochkovskiy et al. [12] improved YOLOv3 and proposed the more powerful YOLOv4 algorithm. The network structure adopted was Cross Stage Partial Networks 53 (CSP DarkNet-53). In YOLOv4, a spatial pyramid pooling network (SPP-Net) was added to enable the model to adapt to inputs of different sizes. A Path Aggregation Network (PANet) [35] was incorporated to fully exploit feature fusion. In June 2020, Redmon et al. launched YOLOv5 [13–15]. The Leaky ReLU activation function, commonly used in neural network architectures, was applied in the middle or hidden layers, and the final detection-layer activation function was sigmoid.

In 2020, Bochkovskiy et al. proposed the YOLOv7 [16,17] algorithm. YOLOv7 continued optimization based on YOLOv5, and had obvious advantages in detection accuracy and speed. First, YOLOv7 incorporated a redesigned extended efficient aggregation network to enhance its feature extraction capabilities. The max-pooling convolution (MPConv) module was introduced for the down sampling operation. This module added a convolution operation based on pooling and realized down sampling through the dual operation of pooling and convolution, which reduced the loss of features. Improved SPP was the final part of the backbone network, and a set of convolution operations was integrated into multiple parallel pooling operations to avoid problems such as image distortion. In the neck network, the PANet structure was still used for network aggregation to ensure the effective integration of different feature layers. In the final prediction head network, REPConv [36] was used to adjust the number of channels. REPconv had different structures during the training and inference stages. Using the concept of reparameterization, the structure of REPConv was simplified in the inference stage without losing accuracy. Although the accuracy of

YOLOv7 has improved over the series, the network structure is too complex, the number of parameters too large, and the equipment performance requirements too high for it to be used for edge-terminal equipment.

YOLOv7-tiny [18,19] simplifies the structure based on YOLOv7, which is a network model designed for edge GPUs. It consists of a backbone, neck, and head, as shown in Figure 1. In the backbone, a simpler ELAN-T is used instead of an extended efficient layer aggregation network (E-ELAN), and the convolution operation in MPConv is cancelled. Only pooling is used for down sampling, which retains the optimized SPP structure and inputs richer feature maps for the neck layer. At the neck, the PANet structure is retained for feature aggregation. At the head, a standard convolution is used to adjust the number of channels instead of REPConv. YOLOv7-tiny sacrifices a certain degree of accuracy, but has advantages in terms of speed and weight compared with YOLOv7.

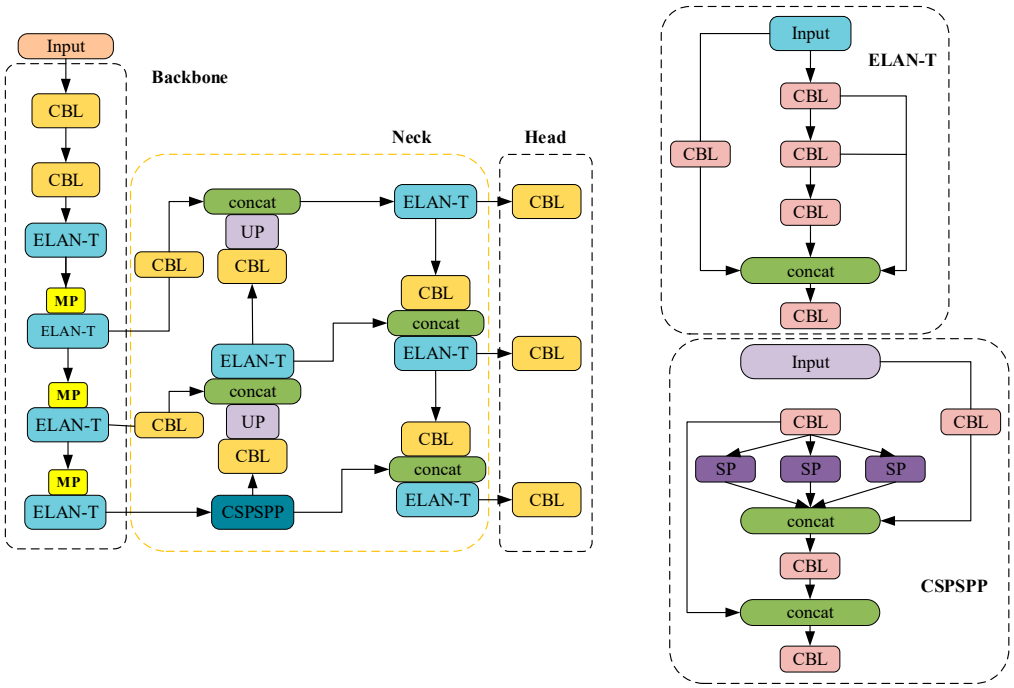


Figure 1. YOLOv7-tiny network structure.

2.2. Detection Head

The object detection head in the object detection model is responsible for detecting objects within the feature pyramid. It typically comprises a series of convolutional, pooled, and fully connected layers. To achieve a better detection performance, studies have proposed different detection heads [13,16,26] based on model characteristics. In the YOLOv5 model, the coupled head plays a crucial role in multiscale object detection; it uses a feature map extracted from the backbone network. The design idea of the module is simple, but it requires many parameters and computing resources, and is easy to overfit. The coupled head includes anchors, classification, and object detection.

Anchors are used to define object boxes of different sizes and aspect ratios in object detection models. These anchor boxes provide reference points for the model to predict the bounding box coordinates and class probabilities. During the preprocessing phase before model training, K-means clustering [37] is applied to group similar object boxes into clusters, and the centroids of these clusters become the anchor box dimensions.

Classification determines whether a detection box contains an object. The features extracted from the detection box typically pass through a fully connected layer, followed by a softmax activation function. The softmax function assigns class probabilities to each class, and the box is assigned to the class with the highest probability if it exceeds a certain threshold.

Object detection regression is used to refine the position and size of each detection box. Typically, this regression is performed using a fully connected layer that learns to predict the necessary adjustments to the initial bounding box coordinates.

YOLOv7 introduced an auxiliary head to the head side for training. When the auxiliary head is used for training, the loss of the auxiliary and detection head is integrated, and the training of the model is deeply supervised. This is equivalent to performing high-level local model ensemble operations in a network to enhance the overall performance of the model.

In YOLOX, the use of a decoupled head for classification and localization was designed to effectively reduce the number of parameters and computational complexity, while simultaneously enhancing the model's generalization ability and robustness. This architectural choice improved the efficiency and overall performance of the YOLOX object detection model. For each level of feature pyramid network (FPN) features, a 1×1 convolutional layer was initially applied to reduce the number of feature channels to 256. This reduction in feature channels helped manage computational complexity while preserving relevant information. Two parallel branches were introduced, each consisting of two 3×3 convolutional layers. These branches were responsible for the classification and localization. Finally, an additional Intersection over Union (IoU) branch was added to the positioning branch.

2.3. IoU

IoU is a metric for assessing object detection algorithms. It quantifies the overlap between predicted and ground truth bounding boxes by dividing their intersection area by their union area. IoU values range from 0 to 1, where 0 indicates no overlap and 1 signifies perfect overlap. Commonly used in tasks such as image segmentation, IoU measures localization accuracy, with higher values indicating more precise object localization. Higher IoU values typically signify superior precision in object localization.

The introduction of Generalized Intersection over Union (GIoU) [38] has transformed the evaluation of object detection, as it takes into account both the overlap and alignment between predicted and ground truth bounding boxes. GIoU offers a more nuanced assessment of object localization accuracy, ensuring a comprehensive evaluation while avoiding unnecessary repetition.

Distance Intersection over Union (DIoU) [39] extends its scope beyond measuring the overlap between bounding boxes. It also considers the distance between the centers of these boxes. This innovative approach provides a more precise and refined assessment of object localization, reducing redundancy in the evaluation process. CIoU introduces a significant innovation in the field of object detection evaluation by incorporating additional geometric insights. In addition to measuring intersection and union, CIoU considers the bounding box aspect ratio and the difference in diagonal lengths. This novel approach ensures a more holistic and precise evaluation of object localization accuracy, further minimizing redundancy in the assessment process. CIoU builds upon the foundation laid by DIoU by extending the IoU metric.

Efficient Intersection over Union (EIoU) [40] directly takes into account the differences in length and width between the predicted and ground truth bounding boxes, in contrast to CIoU. On the other hand, Scylla Intersection over Union (SIoU) [41] places increased emphasis on the regression angle of the bounding box, enhancing the evaluation process.

Tong et al. [28] have noted that the IoU metric can be sensitive to the exact position of bounding boxes, particularly when dealing with small objects or objects requiring precise localization. In response to this challenge, an evaluation metric tailored for small objects was introduced. This metric is known as WIoU, and it was developed to address the balance issue between good- and poor-quality samples in bounding box regression (BBR). Through the use of the Wise gradient gain allocation strategy within a dynamic nonmonotone focusing mechanism, WIoU v3 achieves superior performance.

3. Methodology

In this study, three innovations were introduced, a novel model was established, and enhancements were made to the original algorithm to enhance the detection performance for UAV image objects. Using the P2 small-object detection layer, the size of the detection feature layer was expanded, the model recall rate was improved, and the detection performance for small objects in the UAV images was improved. The use of a decoupled head to weaken the contradiction between classification and regression tasks improved detection accuracy, accelerated network convergence, and ensured information validity. WIoU was used to solve the balance problem between samples of good and poor quality. The network structure of PDWT-YOLO is shown in Figure 2; the improvements are shown within the red dashed lines.

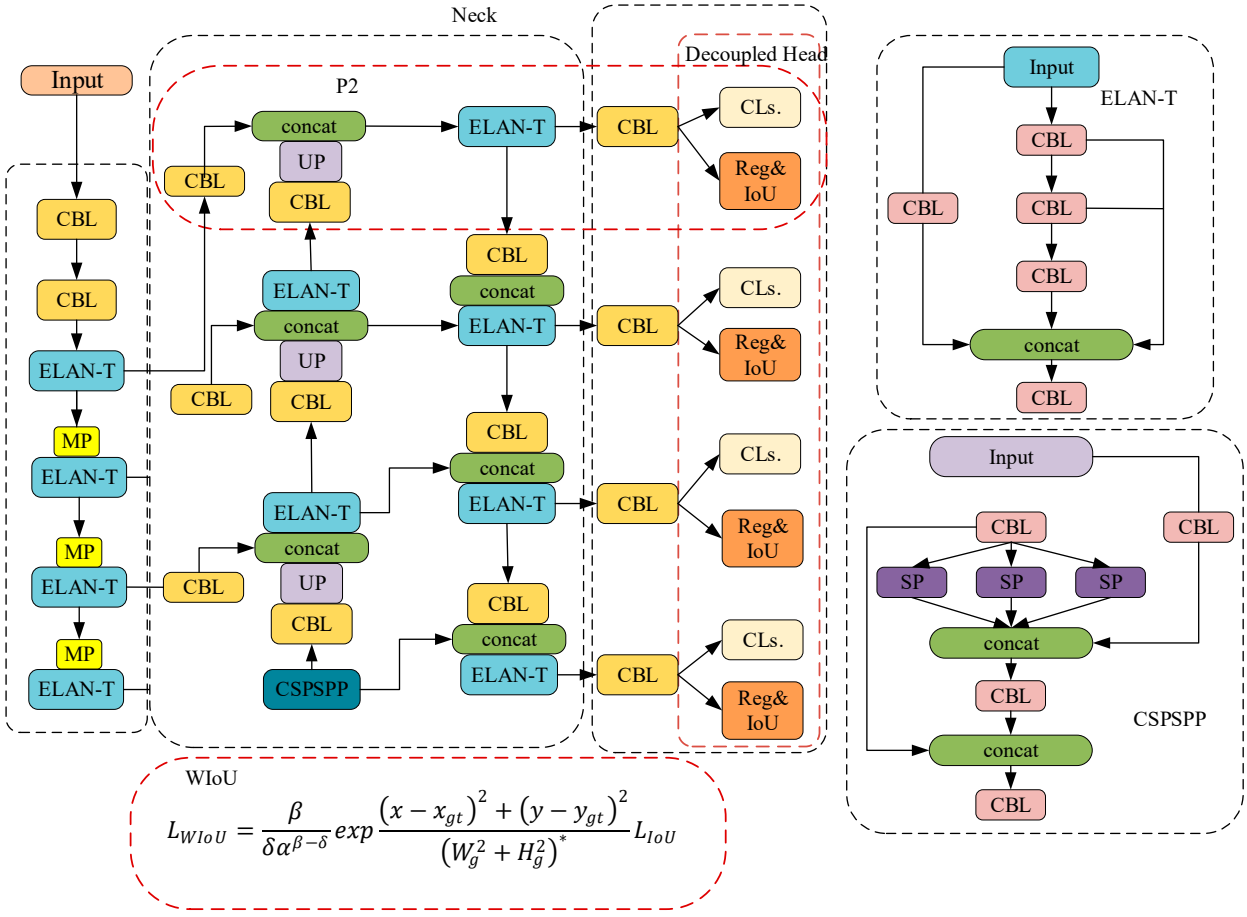


Figure 2. PDWT-YOLO network structure.

3.1. Small-Object Detection Layer

Achieving high accuracy in detecting small objects within UAV images poses a common challenge in computer vision, especially in object detection tasks. Although YOLOv7-tiny may perform well in different application scenarios, it still struggles with small-object detection owing to the use of a convolution-based feature extraction mechanism. As the network depth increases, the resolution of the feature map becomes lower step by step. Moreover, the small features of the target are further weakened because of the pooling layer and convolution kernel operations. These reasons above may result in falsely detecting small targets or missing small targets. The PDWT-YOLO model consists of four detection heads used to detect features of different sizes: 160 × 160, 80 × 80, 40 × 40, and 20 × 20. As illustrated by the role of P2 in Figure 2, a new P2 (160 × 160) small-object detection layer was added, and the resolution of the detection feature layer was expanded. The new small target detection layer was added to generate a feature map with higher resolution.

Hence, the network can maintain more detailed features in the image, which can enable the network to detect more small targets. At the same time, the pyramid structure (as shown in the neck part of Figure 2) can well integrate the shallow features and deep features under various resolutions, which can not only improve the detection effect for small targets, but also not affect the detection accuracy for large and medium targets. Therefore, after using this improved method, the detection performance (such as the recall rate) of the model is expected to improve soundly.

3.2. Decoupled Head

The main function of the detection head is to convert the extracted feature map from the network into the target category, location, confidence, and other information. In the YOLOv7-tiny model, the classification and location regression tasks share a fully connected layer. But classification pays more attention to calculating the most probable category, whereas regression focuses on calculating positions and boundary boxes of targets. The two tasks are different, and may lead to mutual interference if there is only a single head. To solve the contradiction between classification tasks and regression tasks with a single head (which leads to problems of low detection accuracy and slow convergence speed), a decoupled head (including a classification part and a regression part) was added to replace the detection head in YOLOv7-tiny. A decoupled head separates the two tasks of classification and location regression, since two different head structures can optimize their own tasks without affecting each other. Furthermore, the weight of each task can also be adjusted finely. The decoupled head structure, depicted in Figure 3, utilizes an initial 1×1 convolution for dimensionality reduction, followed by two 3×3 convolutions for each of the two parallel classification and regression branches. In the classification branch, a 1×1 convolution was used for the classification operation. In the regression branch, a 1×1 convolution was used for the positioning and confidence operations in the two parallel positioning and confidence branches, respectively. Finally, the anchor-based [42] method was used to extract the object box, compare it with the marked ground truth to determine the difference between the two, and finally produce a detection result. As shown in the decoupled head section of Figure 2, the output of the CBL module in the head is divided into CLs, Reg, and IoU. Our method separates classification and regression tasks with a decoupled head, thereby improving the detection accuracy, accelerating network convergence, and improving the detection results.

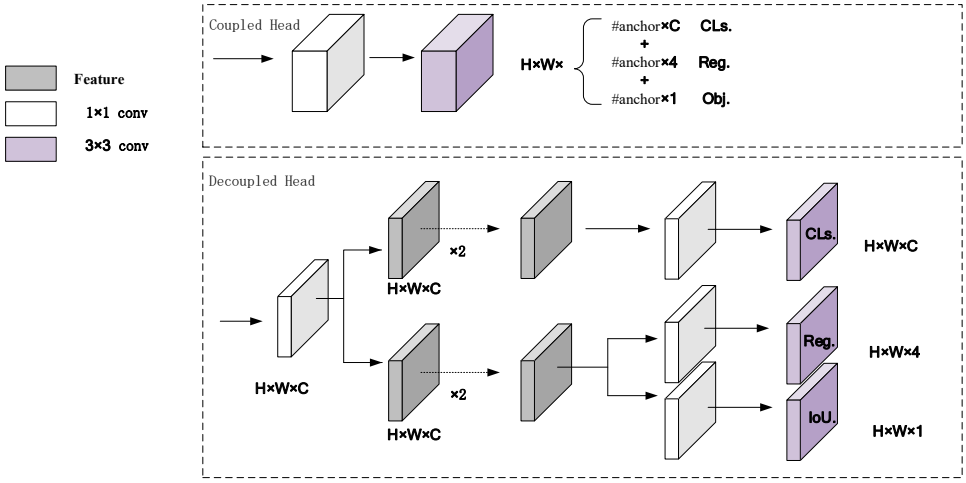


Figure 3. Coupled and decoupled heads compared.

3.3. WIoU

Within the YOLOv7-tiny network, the coordinate loss for predicting the bounding box adopts CIoU. However, CIoU exhibits a significant limitation as it is sensitive to the scale of objects, particularly when dealing with objects of vastly different sizes. This sensitivity

arises from its reliance on the diagonal length of the minimum bounding box, which may not accurately represent the scale of objects with varying shapes. In this study, the scale of the target varies greatly in the UAV image, and there are many small targets. Moreover, CIoU is quite sensitive to input parameters. Hence, large differences in target scales can cause training instability and unsatisfactory performance.

In this study, WIoU replaced CIoU as an improved loss function, because WIoU offers a notable advantage in object detection tasks by assigning different weights to different object classes. This approach allows for a more fine-grained evaluation, emphasizing the accuracy of specific classes over others. By customizing the importance of different classes, WIoU provides a nuanced assessment of object detection performance. This customization minimizes repetition in the evaluation process. The calculation is shown in Equation (1):

$$L_{WIoU} = \frac{\beta}{\delta \alpha^{\beta-\delta}} exp \frac{(x - x_{gt})^2 + (y - y_{gt})^2}{\left(W_g^2 + H_g^2\right)^*} L_{IoU} \quad (1)$$

where 'x' and 'y' denote the center point coordinates of the prediction box, while 'x_{gt}' and 'y_{gt}' represent the center point coordinates of the ground truth box. Furthermore, 'W' and 'H' represent the width and height, respectively, of the minimum bounding box of the ground truth. Additionally, 'α' and 'δ' denote the learning parameters, and 'β' denotes the quality of the bounding box, with lower values indicating higher quality.

4. Experiments

4.1. Dataset

For our study, we utilized still images sourced from the VisDrone-2019 dataset [25], which was publicly released by the AISKYEYE team at Tianjin University. The dataset was compiled using images from different types of drones under a wide range of weather, scene, and lighting conditions, and included 10,209 still images. The highest resolution of a still image was 2000 × 1500 pixels, and the lowest was 960 × 540 pixels. Sample images from the dataset are shown in Figure 4.

Figure 4. Sample images from VisDrone-2019 dataset.

The VisDrone-2019 dataset contains 6471, 548, and 3190 images for training, validation, and testing, respectively. The category distribution of the labels in this dataset is shown in Figure 5.

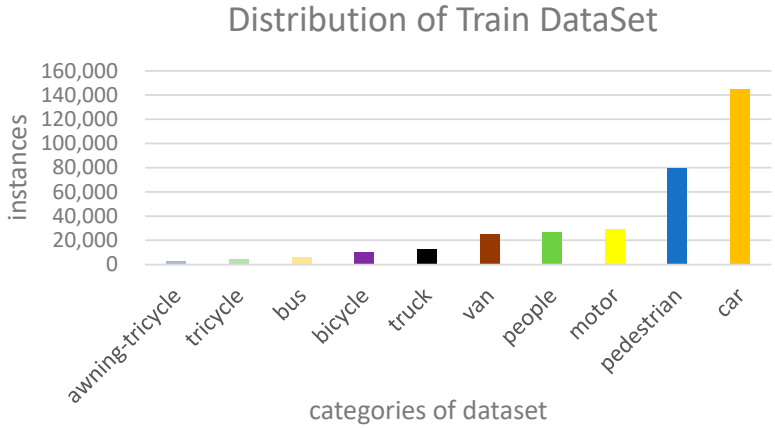


Figure 5. Label distribution of dataset.

4.2. Experimental Platform

An Ubuntu 18.04.4 LTS system was used to validate the proposed object detector. Four GPUs (NVIDIA GeForce RTX 3090 24G (Nvida, Santa Clara, CA, USA)) were used for training and testing. The system configuration included an Intel Xeon Silver 4210 CPU (Intel, Santa Clara, CA, USA) clocked at 2.40 GHz, Python version 3.8, CUDA toolkit version 11.4, and PyTorch version 1.8.0. Throughout the model training phase, input images were standardized to 640 × 640 pixels, and optimization used gradient descent. Key parameters included an initial learning rate of 0.01, a learning rate factor of 0.1, a momentum value of 0.937, and 250 training iterations. All model-related code can be accessed at https://github.com/1thinker1/PDWT-YOLO (accessed on 12 June 2023).

4.3. Evaluation Criteria

A cross-validation was performed to evaluate the experimental results. After training and validation using the corresponding datasets, the final performance evaluation of the model was conducted using the test dataset. In the experiment, the time from the input to the output of the model (inference time), sum of all weights and biases in the neural network (params), billion floating-point operations/s (GFLOPS), and average accuracy/precision (AP) were used as performance indicators to evaluate the detector.

True positives (TP) are correct identification of positive examples, while false positives (FP) are incorrect identification of negative examples as positive. False negatives (FN) are identification of positive examples as negative examples. Accuracy (P) evaluates the object detection precision of the model and is computed using Equation (2):

$$P = \frac{TP}{TP + FP} \tag{2}$$

The recall rate (R) is the proportion of correctly predicted objects. The recall rate can be calculated using Equation (3).

$$R = \frac{TP}{TP + FN} \tag{3}$$

AP represents the area under the curve formed by accuracy and recall rates. The average accuracy, calculated using Equation (4), indicates the average precision across all samples.

$$AP = \int_0^1 P(r)dr \tag{4}$$

The mAP@0.5 stands for the average accuracy across all categories with an IoU threshold of 0.5. Meanwhile, mAP@0.5:0.95 represents the average accuracy across all categories within the IoU range of 0.5 to 0.95. Thus, mAP@0.5 and mAP@0.5:0.95 are the two most commonly used indicators for evaluating the performance of target detection algorithms.

To enhance the description of the detector's capability in multiscale object detection, COCO evaluation indicators, such as average precision for small objects (APS), average precision for medium objects (APM), and average precision for large objects (APL) were used. An APS is a small object with an AP value area of less than 32×32. An APM is an object with an AP value between 32×32 and 96×96. An APL is an object with an AP value greater than 96×96.

4.4. Experimental Results

PDWT-YOLO was compared with other models using the VisDrone-2019 dataset to verify its effectiveness. As Table 1 shows, the values of APM and APL for PDWT-YOLO are lower than those of the Faster R-CNN; however, the number of parameters is also considerably smaller. In addition, Cascade R-CNN has a higher mAP@0.5:0.95; however, it has a params indicator that is considerably larger than that of PDWT-YOLO. Although the params indicator for PDWT-YOLO is slightly higher, the mAP@0.5:0.95 and mAP@0.5 increased by 5% and 6.7%, respectively, compared with the baseline YOLOv7-tiny. The PDWT-YOLO outperformed CenterNet, YOLOv3, YOLOv5l, and YOLOv7-tiny in terms of indicators. In general, PDWT-YOLO is superior to these mainstream models in terms of multiscale object detection performance, which also shows that the model can effectively integrate features of different scales and understand features at different scales.

Table 1. Comparison of different models using VisDrone-2019 dataset.

Method	mAP@0.5:0.95	mAP@0.5	Params (M)	APS	APM	APL
Faster R-CNN	21.9	37.1	137.1	13.1	33.6	37.1
Cascade R-CNN	**24.5**	39	673	**15.2**	**36.7**	39.2
CenterNet	18.7	33.6	104.8	9.8	29.3	38.7
YOLOv3	16.4	31.4	59.13	8.3	26.7	36.5
YOLOX	22.4	39.1	8.9	13.7	33.1	**41.3**
YOLOv5l	20.5	36.2	46.1	12.4	29.9	36.4
YOLOv7-tiny	17.5	34.5	**6.2**	10.4	26.5	36.5
PDWT-YOLO	22.5	**41.2**	6.44	15.1	31.8	36.6

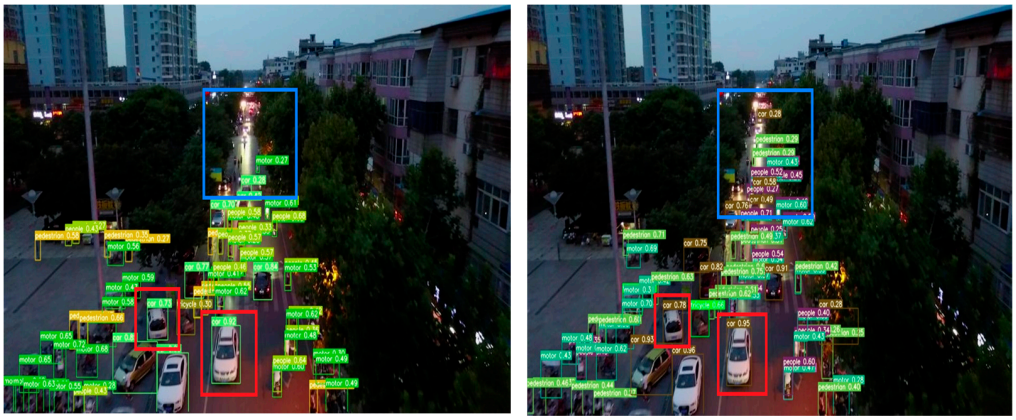
We listed mAP@0.5 for each category, which can describe in detail what has been improved with our model. As shown in Table 2, the mAP@0.5 of our model is compared with those of other models. Improvements were achieved in each category, particularly in the pedestrian and van categories, which increased by 11% and 8%, respectively. These results show that the detection performance of our method is well improved for most categories of targets.

Table 2. Results for individual categories on the VisDrone-2019.

Method	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awning Tricycle	Bus	Motor	mAP@0.5
YOLOv3	12.8	7.8	4.0	43.0	23.5	16.5	9.5	5.1	29.0	12.5	31.4
YOLOv5l	44.4	36.8	**15.6**	73.9	39.2	**36.2**	22.6	11.9	50.5	42.8	36.2
YOLOv7-tiny	37.7	35.9	11.0	74.5	35.2	27.6	22.4	8.3	48.5	43.6	34.5
PDWT-YOLO	**48.7**	**41.6**	14.7	**82.0**	**43.2**	35.4	**26.8**	**14.2**	**56.4**	**49.3**	**41.2**

Figure 6 shows a comparison of the detection effects of YOLOv7-tiny (Figure 6a) and PDWT-YOLO (Figure 6b) on the VisDrone-2019 test dataset. As shown in the blue box of Figure 6b, we can directly observe that PDWT-YOLO successfully detected more

small objects than those shown in Figure 6a, particularly in the far field of view, which is equivalent to reducing the likelihood of small objects being missed or incorrectly detected. In addition, the detection confidence and the detection accuracy for the object were also improved. For example, the detection confidence ratings for the two white cars in the red boxes in Figure 6b improved, respectively, from 0.73 to 0.78 and 0.92 to 0.95 compared with those in Figure 6a. Hence, it can be concluded that our model improved the ability to detect small objects well.



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

Figure 6. Dataset detection effect comparison. (**a**) YOLOv7-tiny, and (**b**) PDWT-YOLO.

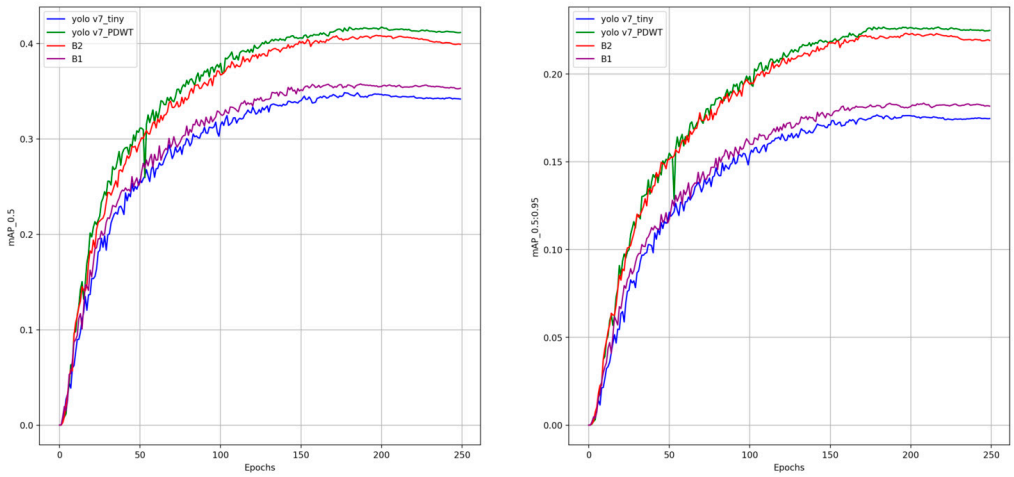
4.5. Ablation Experiment

Ablation experiments were conducted on the PDWT-YOLO model using the VisDrone-2019 dataset, adding the different improvement methods mentioned above, individually or in combination, to verify the effectiveness of each individual method. To ensure accuracy, all the experiments were conducted using the same parameters and environments. This experiment involved adding different improvements to the YOLOv7-tiny model, including adding a decoupled head (B1 in Table 3), continuing to add the P2 layer small-object detection head (B2 in Table 3), and then adding WIoU (PDWT-YOLO in Table 3). The experimental results show that each individual addition improved the mAP@0.5 and mAP@0.5:0.95, and the three indicators of inference time, params, and FLOPS did not increase significantly. The mAP@0.5, mAP@0.5:0.95, APS, and APM of the B1 model increased by 1%, 0.7%, 0.9%, and 0.7%, respectively, compared with the basic model. The mAP@0.5 and mAP@0.5:0.95 of the B2 model increased by 4.7% and 4%, respectively, and the APS, APM, and APL also increased compared with the B1 model. The performance of the PDWT-YOLO was better than that of the B2 model; the mAP@0.5, mAP@0.5:0.95, APS, and APL were, respectively, 1%, 0.3%, 0.4%, and 0.4% higher than those of the B2 model. Although its APL was lower than that of the B2 model, the detection ability of the comprehensive-scale features was better. This indicates that WIoU further improved the accuracy of the target detection model. The ablation experiment shows that our three improvements (independently or jointly) can enhance performance in object detection.

Table 3. Ablation experiments with VisDrone-2019 test dataset.

Method	Decoupled Head	P2	WIoU	mAP@0.5	mAP@ 0.5:0.95	Inference Time (ms)	Params	GFLOPS	APS	APM	APL
YOLOv7-tiny				34.5	17.5	**2.9**	6.2	**13.1**	10.4	26.5	36.5
B1	√			35.5	18.2	3.2	**5.8**	18.9	11.3	27.2	34.2
B2	√	√		40.2	22.2	4.3	6.44	24.2	14.7	31.4	**38**
PDWT-YOLO	√	√	√	**41.2**	**22.5**	4.4	6.44	24.2	**15.1**	**31.8**	36.6

The entire change processes of mAP@0.5 and mAP@0.5 are plotted in Figure 7, and each curve can converge at about 200 epochs. Evidently, each of the additions enhances the performance of the model compared with YOLOv7-tiny.



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

Figure 7. Training comparison of YOLOv7-tiny after adding a decoupled head (B1), P2 layer small-object detection head (B2), and WIoU (PDWT-YOLO). (**a**) mAP@0.5, and (**b**) mAP@0.5:0.95.

4.6. IoU Contrast Experiment

Based on the addition of a decoupled head and P2 layer, the impact of different IoU categories on the PDWT-YOLO was explored. As shown in Table 4, the performance of WIoU is better than that of the other IoUs.

Table 4. IoU contrast experiment.

IoU	mAP@0.5:0.95	mAP@0.5
CIoU	22.2	40.6
GIoU	22.0	40.2
DIoU	22.4	40.6
WIoU	**22.5**	**41.2**

4.7. Extended Experiment

To verify the generalizability of the PDWT-YOLO model, training and validation were conducted on a universal HIT-UAV dataset. The training set contained 2008 images, the validation set 287, and the test set 571 images. The mAP@0.5 values for each category are listed in Table 5.

Table 5. Results for individual categories on the HIT-UAV.

Method	Person	Car	Bicycle	Other Vehicle	Do Not Care	mAP@0.5
YOLOv7-tiny	89.1	**97.1**	88.3	67.1	52.1	78.8
PDWT-YOLO	**92.7**	96.6	**89.5**	**67.2**	**65.7**	**82.3**

The outcomes of the extended experiment demonstrate that the PDWT-YOLO model enhances mAP@0.5 in all image categories, with the exception of cars. The introduction of the P2 layer for feature fusion notably improved the representation of small- and medium-sized objects. These findings underscore the model's generalization capabilities, affirming its suitability for UAV image detection.

5. Discussion

Historically, drones have predominantly relied on manual intervention for specific tasks. However, with the rapid advancements in artificial intelligence technology, drones can now autonomously execute tasks such as object recognition and flight. While various object detection algorithms have made substantial improvements in performance, striking the right balance between accuracy and speed remains a challenge due to variations in object scale, the presence of numerous small objects, and the complexity of scenes within drone images.

The PDWT-YOLO algorithm showcased exceptional performance on the VisDrone-2019 dataset when compared with state-of-the-art algorithms, including Faster R-CNN, Cascade R-CNN, CenterNet, and the YOLO series, as demonstrated in Table 1. It achieved the highest mAP@0.5 with the second-lowest number of required parameters. Notably, PDWT-YOLO outperformed YOLOv7-tiny, which had fewer parameters but a lower mAP@0.5 score. PDWT-YOLO particularly excelled in enhancing the detection accuracy for small objects, as is evident in Tables 2 and 3 and Figure 6. Furthermore, it exhibited superior accuracy on the HIT-UAV drone dataset, which encompasses multiscale objects.

In the ablation experiment (Table 3), the addition of the decoupled head to PDWT-YOLO resulted in improvements across all performance indicators, with the exception of APL, which was reduced by 2.3%. Most notably, mAP@0.5 increased by 1%, indicating an enhanced ability to extract object position and category information from the feature map. This shows that the decoupled head achieved better performance by separating the classification and location regression of the model. Subsequent incorporation of small-object layers further improved detection indicators for objects of all sizes, with APS increasing by 3.4%, APM by 4.2%, and APL by 3.8%. It can be concluded that the P2 (the added small target layer based on FPN) makes the model more focused on small targets, and improves the detection performance of the model for multiscale targets. The addition of WIoU further heightened accuracy, resulting in a 1% increase in mAP@0.5 without adding computational complexity, thus ensuring both high accuracy and faster convergence. In comparison with the baseline algorithm, the PDWT-YOLO algorithm significantly boosted performance metrics, elevating mAP@0.5 by 6.7%, mAP@0.5:0.95 by 5%, APS by 4.7%, and APM by 5.3%. In summary, these three innovations in the PDWT-YOLO model have demonstrated the ability to enhance detection performance, whether implemented individually or in combination, and offer a substantial improvement in object detection for UAV applications.

6. Conclusions

The widespread integration of drones into everyday life has been significantly facilitated by deep learning technology. In this study, we introduced an enhanced object detection algorithm, PDWT-YOLO, designed specifically for UAVs. This model represents a substantial advancement in object detection for drone images characterized by diverse scales and complex backgrounds. This model mainly has the following three advantages compared with YOLOv7-tiny.

- To enhance the detection performance for small objects in drone images, a small-object detection layer was incorporated into the algorithm.
- Additionally, a decoupled head was added in place of the detection head IDetect in YOLOv7-tiny, which mitigated conflicts between classification and regression and improved detection accuracy.
- Finally, WIoU was used instead of CIoU in the loss function to improve the network convergence speed and improve regression accuracy.
- The experimental results demonstrate that PDWT-YOLO outperforms YOLOv7-tiny in object detection accuracy and has good network convergence performance for multiscale targets, especially for small objects. It can be inferred that the PDWT-YOLO model excels at extracting targets from intricate backgrounds with greater precision when compared with the YOLOv7-tiny model.

Author Contributions: Conceptualization, L.Z.; methodology, L.Z.; software, X.P.; validation, X.P.; investigation, N.X. and C.G.; resources, P.W.; data curation, L.Z. and N.X.; writing—original draft preparation, L.Z. and P.W.; writing—review and editing, N.X. and X.P.; visualization, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the General project of the Key R & D Plan of Shanxi Province, high-technology field (grant number 201903D121171) and the National Natural Science Foundation of China (serial no. 61976134).

Data Availability Statement: The datasets presented in this study are available through https://github.com/VisDrone/VisDrone-Dataset and https://github.com/suojiashun/HIT-UAV-Infrared-Thermal-Dataset (accessed on 12 June 2023).

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

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision
APL	Average precision for large objects
APM	Average precision for medium objects
APS	Average precision for small objects
CIoU	Complete Intersection over Union
CSP DarkNet	Cross Stage Partial Networks
DIoU	Distance Intersection over Union
E-ELAN	Extended efficient layer aggregation network
EIoU	Efficient Intersection over Union
FN	False negative
FP	False positive
FPN	Feature pyramid network
GFLOPs	Giga floating-point operations per second
GIoU	Generalized Intersection over Union
GPU	Graphics processing unit
IOU	Intersection over Union
MPConv	Max-pooling convolution
PANet	Path Aggregation Network
R-CNN	Region-based convolutional neural network
ResNet	Residual Network
SIoU	Scylla Intersection over Union
SSD	Single Shot MultiBox Detector
SPP-Net	Spatial pyramid pooling network
TP	True positive
UAV	Unmanned aerial vehicle
VGG	Visual Geometry Group Network
WIoU	Wise Intersection over Union
YOLO	You Only Look Once

References

1. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
2. Kaur, J.; Singh, W. Tools, techniques, datasets and application areas for object detection in an image: A review. *Multimed. Tools Appl.* **2022**, *81*, 38297–38351. [CrossRef]
3. Li, Y.; Zhang, Y.; Yu, J.-G.; Tan, Y.; Tian, J.; Ma, J. A Novel Spatio-Temporal Saliency Approach for Robust Dim Moving Target Detection from Airborne Infrared Image Sequences. *Inf. Sci.* **2016**, *369*, 548–563. [CrossRef]
4. Ahmed, M.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Survey and Performance Analysis of Deep Learning Based Object Detection in Challenging Environments. *Sensors* **2021**, *21*, 5116. [CrossRef] [PubMed]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [CrossRef]
6. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]

7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; Volume 28, pp. 91–99. [CrossRef]
8. Zhao, W.; Huang, H.; Li, D.; Chen, F.; Cheng, W. Pointer Defect Detection Based on Transfer Learning and Improved Cascade-RCNN. *Sensors* **2020**, *20*, 4939. [CrossRef]
9. 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), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
10. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525. [CrossRef]
11. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
12. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
13. 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 IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788. [CrossRef]
14. Ultralytics. Yolov5. [EB/OL]. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 November 2021).
15. Chen, Z.; Zhang, F.; Liu, H.; Wang, L.; Zhang, Q.; Guo, L. Real-Time Detection Algorithm of Helmet and Reflective Vest Based on Improved YOLOv5. *J. Real-Time Image Process* **2023**, *20*, 4. [CrossRef]
16. Wu, D.; Jiang, S.; Zhao, E.; Liu, Y.; Zhu, H.; Wang, W.; Wang, R. Detection of *Camellia oleifera* Fruit in Complex Scenes by Using YOLOv7 and Data Augmentation. *Appl. Sci.* **2022**, *12*, 11318. [CrossRef]
17. Jiang, K.; Xie, T.; Yan, R.; Wen, X.; Li, D.; Jiang, H.; Jiang, N.; Feng, L.; Duan, X.; Wang, J. An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation. *Agriculture* **2022**, *12*, 1659. [CrossRef]
18. Li, B.; Chen, Y.; Xu, H.; Fei, Z. Fast Vehicle Detection Algorithm on Lightweight YOLOv7-Tiny. *arXiv* **2023**, arXiv:2304.06002.
19. Kulyukin, V.A.; Kulyukin, A.V. Accuracy vs. Energy: An Assessment of Bee Object Inference in Videos from On-Hive Video Loggers with YOLOv3, YOLOv4-Tiny, and YOLOv7-Tiny. *Sensors* **2023**, *23*, 6791. [CrossRef] [PubMed]
20. 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, Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016*; Springer: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [CrossRef]
21. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. Automatic Ship Detection Based on RetinaNet Using Multi-Resolution Gaofen-3 Imagery. *Remote Sens.* **2019**, *11*, 531. [CrossRef]
22. Zhang, X.; Fan, K.; Hou, H.; Liu, C. Real-Time Detection of Drones Using Channel and Layer Pruning, Based on the YOLOv3-SPP3 Deep Learning Algorithm. *Micromachines* **2022**, *13*, 2199. [CrossRef] [PubMed]
23. Wang, L.; Liao, J.; Xu, C. Vehicle Detection Based on Drone Images with the Improved Faster R-CNN. In Proceedings of the 2019 11th International Conference on Machine Learning and Computing (ICMLC'19), Zhuhai, China, 22–24 February 2019; pp. 466–471. [CrossRef]
24. Huang, H.; Li, L.; Ma, H. An Improved Cascade R-CNN-Based Target Detection Algorithm for UAV Aerial Images. In Proceedings of the 2022 7th International Conference on Image, Vision and Computing (ICIVC), Xi'an, China, 26–28 July 2022; pp. 232–237. [CrossRef]
25. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q.; Zheng, J.; Peng, T.; Wang, X.; Zhang, Y.; et al. VisDrone-SOT2019: The Vision Meets Drone Single Object Tracking Challenge Results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 199–212.
26. Liu, C.; Xie, N.; Yang, X.; Chen, R.; Chang, X.; Zhong, R.Y.; Peng, S.; Liu, X. A Domestic Trash Detection Model Based on Improved YOLOX. *Sensors* **2022**, *22*, 6974. [CrossRef]
27. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
28. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
29. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2022**, *52*, 8574–8586. [CrossRef]
30. Shin, H.-C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Trans. Med. Imaging* **2016**, *35*, 1285–1298. [CrossRef]
31. Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X.; et al. PP-YOLOv2: A Practical Object Detector. *arXiv* **2021**, arXiv:2104.10419.
32. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A.; Liu, W.; et al. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]

33. 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), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
35. 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 (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. [CrossRef]
36. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737. [CrossRef]
37. Sinaga, K.P.; Yang, M.-S. Unsupervised K-Means Clustering Algorithm. *IEEE Access* **2020**, *8*, 80716–80727. [CrossRef]
38. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [CrossRef]
39. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [CrossRef]
40. 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]
41. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
42. Li, B.Y.; Liu, Y.; Wang, X.G. Gradient Harmonized Single-Stage Detector. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8577–8584. [CrossRef]