electronics

MDPI

Article

Object Detection Based on an Improved YOLOv7 Model for Unmanned Aerial-Vehicle Patrol Tasks in Controlled Areas

Dewei Zhao, Faming Shao, Li Yang, Xiannan Luo, Qiang Liu *, Heng Zhang and Zihan Zhang

College of Field Engineering, Army Engineering University, PLA, Nanjing 210007, China; zhaodewei533@126.com (D.Z.); shaofaming@aeu.edu.cn (F.S.); yangli563@163.com (L.Y.); lxn@163.com (X.L.); hengzhang4216@163.com (H.Z.); zzh2023@aeu.edu.cn (Z.Z.)
* Correspondence: 533@aeu.edu.cn; Tel.: +86-166-5168-3533

Abstract: When working with objects on a smaller scale, higher detection accuracy and faster detection speed are desirable features. Researchers aim to endow drones with these attributes in order to improve performance when patrolling in controlled areas for object detection. In this paper, we propose an improved YOLOv7 model. By incorporating the variability attention module into the backbone network of the original model, the association between distant pixels is increased, resulting in more effective feature extraction and, thus, improved model detection accuracy. By improving the original network model with deformable convolution modules and depthwise separable convolution modules, the model enhances the semantic information extraction of small objects and reduces the number of model parameters to a certain extent. Pretraining and fine-tuning techniques are used for training, and the model is retrained on the VisDrone2019 dataset. Using the VisDrone2019 dataset, the improved model achieves an mAP_{50} of 52.3% on the validation set. Through the visual comparative analysis of the detection results in our validation set, we find that the model shows a significant improvement in detecting small objects compared with previous iterations.

Keywords: drone patrol; control area; object detection; deformable attention; deformable convolution; depthwise separable convolution; YOLOv7

Citation: Zhao, D.; Shao, F.; Yang, L.; Luo, X.; Liu, Q.; Zhang, H.; Zhang, Z. Object Detection Based on an Improved YOLOv7 Model for Unmanned Aerial-Vehicle Patrol Tasks in Controlled Areas. *Electronics* **2023**, *12*, 4887. https://doi.org/10.3390/electronics12234887

Academic Editor: Hamid Reza Karimi

Received: 30 October 2023
Revised: 26 November 2023
Accepted: 28 November 2023
Published: 4 December 2023

1. Introduction

With the rapid development of modern drone technology, drones have replaced a great deal of human labor due to their lightweight, small, and inexpensive characteristics. There is also a major trend of using drone patrols to monitor areas in important locations such as factories, outdoor agricultural and sideline breeding production bases, primitive natural ecological protection areas, large-scale distributed warehouses, and various important experimental sites [1,2]. For autonomous cruising drones, the basic process of patrol-based area control through the use of patrols begins with the use of images drone-captured from the patrol area for detection. Then, the method provides an early warning of the detected targets in the images. Researchers adopt corresponding measures to perform processing based on the received warning information. Compared with manual patrols, this method greatly saves time and energy investment.

However, when comparing the targets captured by drones with those obtained using general fixed-height camera monitoring, drones have a higher shooting height, and the images captured by drones show the prominent features of smaller targets. How to improve the accuracy of small-object detection in drone images has always been a goal pursued by people.

The traditional object-detection technology, V-J detector [3,4], began to be used in 2001 and is mainly utilized for facial detection. By 2005, the HOG + SVM [5] method had emerged, which is mainly used for pedestrian detection. Afterward, Ross Girshick proposed the famous deformable component model [6,7] for object-detection tasks. In order to achieve good detection results, extensive time is required to design different models

for different applications, which is a huge workload. The basic process of the traditional algorithms is shown in Figure 1.

Figure 1. Basic process of object detection. The traditional process of completing an object-detection task is shown in Figure 1: the first step is to select the detection window, the second step is to extract the image features, and the third step is to design a classifier.

Object-detection technology is based on deep neural networks. In 2012, the application of the convolutional neural network AlexNet [8] in the field of object detection began to receive widespread attention. In 2013, the feature-extraction operator Overfeat [9] was proposed on the basis of convolutional neural networks. In 2014, Ross Girshick proposed a two-stage RCNN [10] detection algorithm for object detection, marking the beginning of two-stage object detection. After 2015, fast RCNN [11] and faster RCNN [12] versions of RCNN followed. Meanwhile, YOLO [13] had also been developed, with its arrival opening up a stage of detection algorithm development. In 2016, SSD [14] was proposed, and 2017 to 2018 saw pyramid networks [15] and retina nets [16] being successively proposed. At the same time, various variants of the YOLO series, such as YOLOv1~YOLOv6 [13,17–21], YOLOX [22], and YOLOv7 [23], have been proposed in recent years. The use of neural networks for object detection has induced a qualitative leap in accuracy, efficiency, and adaptability to different scenarios compared with traditional algorithms.

YOLOv7, an object-detection model with a high accuracy and detection rate, was proposed in July 2022. Our research has improved the existing YOLOv7 model to combine the characteristic applications of object detection for drone patrols in controlled areas. Our contribution has the following three aspects:

1. We propose improving the backbone network of YOLOv7 by applying a deformable attention module. Through the application of deformable attention, the constraint of kernel size in convolution is broken, and the correlation between distant feature points is enhanced. When extracting image features, richer semantics are formed, thereby improving the accuracy of detecting small objects in drone-captured images.
2. We further reduce the number of model parameters by utilizing deformable convolutions and deep separable convolutions. This improvement enhances the receptive field range of the convolution kernel in the model, reduces the total number of model parameters, enriches the semantic information of the feature map, and further improves the accuracy of small-object detection.
3. We propose the application of image-data augmentation, transfer learning, fine-tuning, and label-smoothing training in the model-training stage. For the VisDrone dataset used by our application target, the Mosaic image-data augmentation method was used to solve the limitation of small data volume. During the training process, strategies such as transfer learning, fine-tuning, and label smoothing are utilized to accelerate the model training speed and enhance the robustness of the model.

The overall structure of the article is as follows: The second part introduces the literature related to the research described in the article. The third part provides a specific introduction to the modification of the model and the methods used. The fourth part introduces and analyzes the dataset used and introduces the relevant metrics for the model training. The fifth part introduces the specific methods we used to train the model, analyzes the training results, and tests the effectiveness of the model improvement through ablation and comparative experiments. We visualize and discuss some experimental groups.

The sixth part summarizes the article and proposes further research on the application in the future.

2. Related Work

In this section, we mainly introduce relevant research on the use of deep neural networks for object-detection technology in recent years, the modules related to improving the performance of deep neural network models involved in model improvement in this article, and the relevant methods for improving the quality of model training.

2.1. Object-Detection Algorithm Based on Deep-Learning Networks

Object detection is an important task in the field of computer vision, with a primary task of classifying and locating targets. Deep-learning networks, with their outstanding ability to automatically extract features from a large amount of data through supervised methods, have replaced the traditional target detection approach of manual feature extraction and classification. The direction of object detection based on deep learning can be roughly divided into three categories: two-stage object detection, single-stage object detection, and transformer-based object detection [24,25].

The two-stage object-detection algorithm selects instance bounding boxes based on the image, and then performs secondary correction based on the bounding-box area to obtain detection results. The detection accuracy is high, but its speed is slow. This type of algorithm starts with the RCNN [10], and is subsequently improved using the fast RCNN [11] and the faster RCNN [12]. Algorithms such as FPN [15] and mask RCNN [26] offer improvements to address the shortcomings of the faster RCNN, further enriching its components and improving its performance. However, this type of method has the disadvantage of not being able to achieve end-to-end training and slow prediction time, and it cannot handle large datasets.

Compared with the two-stage object-detection algorithm, the single-stage object-detection algorithm directly calculates the image to generate detection results. Despite its accuracy limitations, this method quickly generates results end-to-end and is currently a widely used algorithm. This type of algorithm originated from YOLOv1 [13], which was subsequently improved by SSD [14] and Retinanet [16]. In the following period, YOLOv1 was successively improved, resulting in the versions YOLOv2 [17], YOLOv3 [18], YOLOv4 [19], YOLOv5 [20], YOLOv6 [21], YOLOX [22], and YOLOv7 [23]. These improved versions gradually enhanced the prediction accuracy of the single-stage target detection algorithm and also accelerated the inference speed.

The transformer-based method [27] mainly utilizes the attention mechanism to model the relationship between targets, incorporating relationship information into features and thus achieving feature enhancement. Relationship Net [28] and DETR [29] have proposed a new object-detection architecture based on a transformer with the aim of ushering in a new era of object detection. However, due to the large overall training parameters and long training practice of these models, further development is still needed—a hot topic in current research.

2.2. Module for Improving the Performance of Deep Neural Network Models

In this article, deformable convolution [30], deformable attention [31], depthwise separable convolution [32], and a new activation Mish function [33] are used to improve the overall performance of the model. Among these, deformable convolution introduces learnable offsets compared with traditional convolution operations, increasing the receptive field range of large convolutional kernels and enhancing the ability to extract semantic information. Deformable attention utilizes the idea of deformable convolution, employing the location of learnable sampling points, not to mention efficient attention mechanisms, for local and sparse processing to accelerate network convergence and to solve the problem of limited feature resolution in processing. Depthwise, separable convolution generates feature maps for channel volumes through convolution kernels of different layers, resulting

in significantly smaller model parameters. Mish is a self-regularized non-monotonic neural activation function, and a smooth activation function allows gradient information to be better transmitted via deep neural networks, resulting in better accuracy and better generalization ability of the model.

There are many improvement measures in neural networks for target detection tasks, such as using a data augmentation network based on cropping [34] to improve detection performance and changing the internal structure of the transformer to improve performance [35]. This article uses the popular YOLOv7 network [36–39] as the basis based on the characteristics of the task. Unlike other improvement methods, we use modules with sparse network parameters such as the attention mechanism and deformable convolution to improve the network and explore an improvement method that improves accuracy and reduces the number of parameters.

2.3. Related Strategies for Improving Model Training Effectiveness

Image-data augmentation [40] refers to the use of a certain image-processing method to process data that change the shape, color, chromaticity, angle, position, etc., of a graph, thereby increasing the capacity of the dataset and enhancing the model's generalization ability. Transfer learning [41] and fine-tuning [42] methods use the weights of previously trained models to initialize the weights of the new model put forward for training. The result is that the training stands at a starting point explored in the early stage. This approaches the optimization goal, thereby accelerating the convergence of the model. The label-smoothing training method is a regularization measure that enhances the generalization ability of the model, and it is a popular method in multi-objective loss calculation in model training.

3. Methodology

In this section, we provide a detailed description of the improvements made to the YOLOv7 [23] model, as well as the advantages and roles of the modules used in the model improvements.

3.1. Framework Structure of the Proposed Model

Due to the fast movement speed and high flying altitude of drone patrols compared with regular fixed-altitude cameras, it is necessary to meet certain requirements for image processing and detection speed and accuracy. YOLOv7 is an algorithm proposed in recent years that can perform real-time image processing, and its speed is higher than that required for ordinary object-detection algorithms, with the same processing accuracy. We chose the YOLOv7 model for this purpose. In response to the fact that there are many small targets in the application field of drone patrols in the control area, as well as some shortcomings in the accuracy of YOLOv7′s small-target detection model, we combined relevant knowledge to improve it.

The left side of Figure 2 represents the backbone network part of the model, an area primarily responsible for the feature extraction of input graphics, obtaining certain semantic information from the input images, and preparing for the object-detection task in the early stage. The red, yellow, and blue rectangular blocks on the bottom left represent the R, G, and B channels of the original input image. The lines in the figure represent the corresponding operation and processing procedure, and the text next to the lines signifies the abbreviation of the corresponding processing module. The yellow cubes of different sizes in the figure are feature maps generated after processing by the processing module. The numerical annotation form of the feature map is $H \times W \times C$, where H, W, and C represent the pixel height, pixel width, and number of channels of the feature map, respectively. Their order is consistent with the coordinate direction at the bottom left corner of the map. In the middle section of Figure 2—dubbed the "Neck"—we find a network that integrates features from different layers of feature map outputs from the backbone network. On the right side, we see the section referred to as the "Head", performing detection operations on feature

maps. During the training process of the model, this area is responsible for calculating the detection errors and losses and for outputting the detection results during the detection application process. The formula at the bottom right corner of the figure is a graphical explanation for the combination of a number of modules.

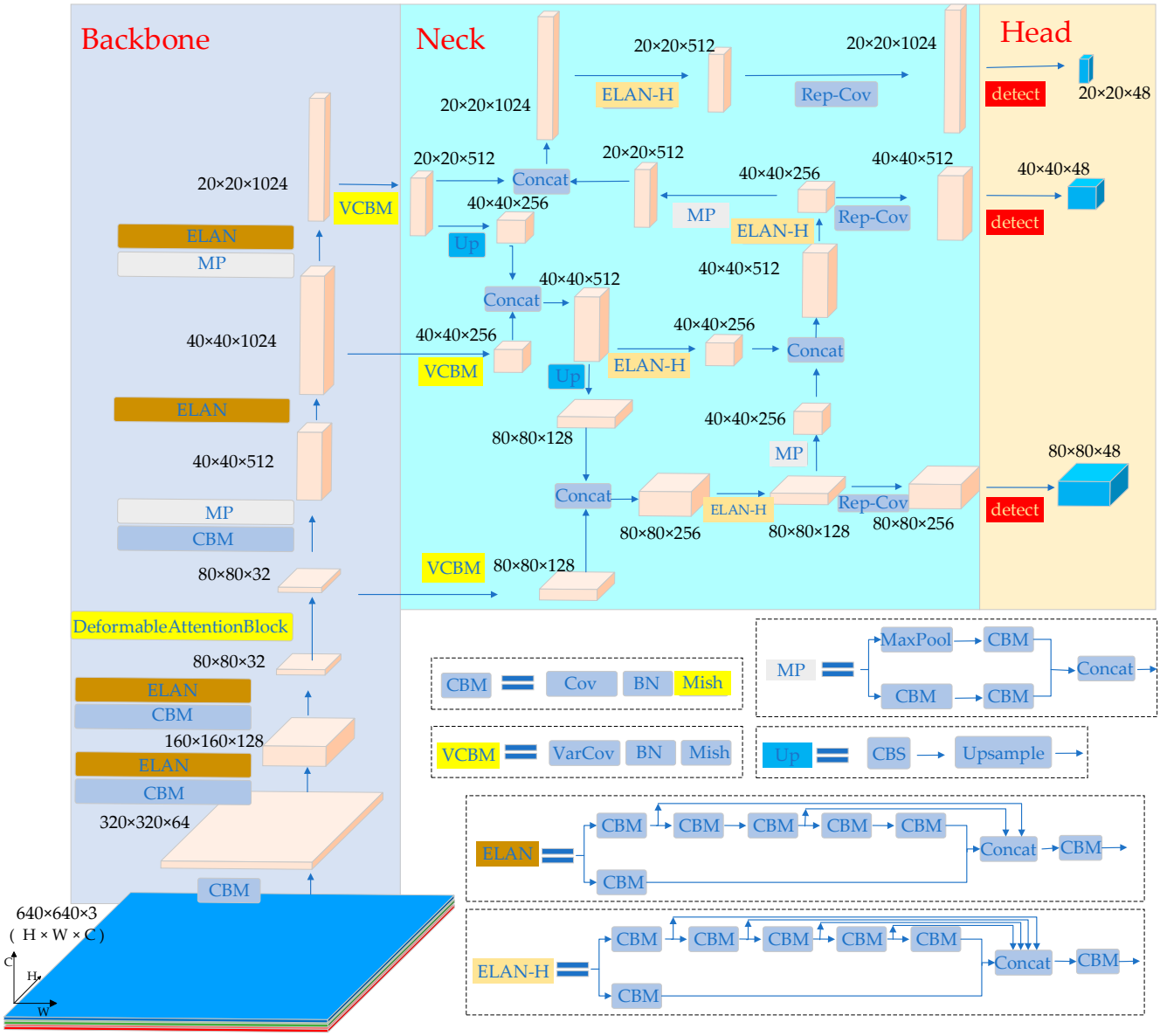


Figure 2. Architecture of our improved model.

Regarding the YOLOv7 original framework, our improvements to the model primarily focus on three aspects: Firstly, in the backbone network section, we introduce a deformable attention module to replace the third processing module in the original model. At the same time, considering that the parameter quantity of the deformable attention module is related to the number of input and output channels, we reduce this number in the third module of the backbone network, while maintaining the same pixel height and width during the input and output. Secondly, in the input part of the neck, we introduce a deformable convolution module to replace the original processing module. By using deformable convolution, we reduce the number of model parameters and increase the ability to extract semantic feature information. Thirdly, we use variable depthwise separable convolutions to replace the ordinary convolutions in the original model throughout the entire network, while using

Mish activation functions to replace the SiLU activation functions in the original model. The modified parts are compared with the original network and highlighted in yellow in Figure 2.

3.2. The Modules We Used

3.2.1. Deformable Attention Module

In Figure 3, Query, Value, and Key represent the basic elements required to perform the attention mechanism operations. In Figure 3, Offset describes a set of paranoid parameters used to calculate Value. The values of Offset and Key are obtained through linear transformation using the Query feature, and the specific value of Value is obtained via Offset. On the right side of the figure, Query, Value, and Key obtain the final result for the input feature x. For Figure 3, the mathematical equation for deformable attention is Equation (1) [31]:

$$DeAttn(z_q, x) = \sum_{m=1}^{M} W_m \left[\sum_{k=\Omega_k} A_{mqk} \cdot W'_m x(p_q + \Delta p_{mqk}) \right] \quad (1)$$

In Formula (1), z_q represents the feature vector of Query, x represents the input feature map, M represents the total number of attention heads, m represents the m-th attention head, and W_m represents the weight of a linear transformation. Additionally, Ω_k represents the entire set of k values; A_{mqk} represents the weight of the k-th Key elements, which is a normalized weight vector; W'_m prime represents the encoding of the Key element; p_q represents the position without bias changes; and Δp_{mqk} represents the specific offset value. The engineering implementation of the formula utilizes two fully connected neural networks to generate the offset value and attention weights. These two fully connected networks share weights for all the input features, and the weights of the fully connected network are obtained through learning.

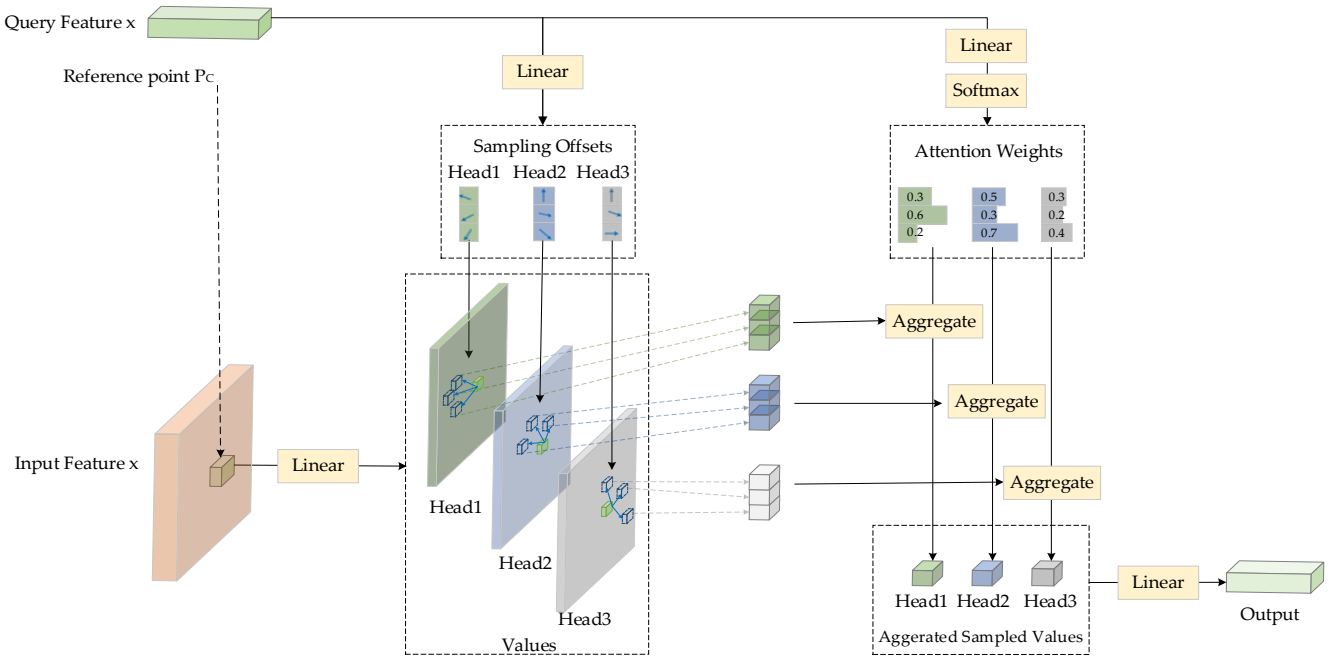


Figure 3. Deformable attention module.

By combining Figure 3 with Formula (1), we concluded that deformable attention is an attention mechanism based on sparse spatial sampling. In traditional attention mechanisms, the number of Keys and Queries is equal to the number of all Value figures. The difference between this model and those used in traditional attention is that the deformable attention is based on different Value values. The numbers and values of Keys and Queries required for this are generated via network learning, and the original dense

network structure changes. This greatly reduces the computational complexity required for the attention mechanisms and reduces the number of model parameters. The learned Key and Query not only reduce the redundancy of output features, but also further enhance the feature-extraction capabilities.

3.2.2. Deformable Convolution Module

Figure 4 is a schematic diagram of variable convolution, with an input feature map on the left and an output feature map on the right. In the left feature map, the blue lines represent small squares, thus signifying ordinary convolutional kernels. In the figure, the ordinary convolutional kernel is a rectangle of 3×3. The green square is a deformed convolutional kernel that has an irregular scattered shape. The upper part of the figure represents the offset generated via convolution, and the ordinary convolutional kernel shifts relative to its original position under the effect of the offset. The mathematical expression of deformable convolution is Equation (2):

$$y(P_0) = \sum_{P_n \in \mathcal{R}} w(P_n) \cdot X(P_0 + P_n + \Delta P_n) \tag{2}$$

where P_0 represents the center position of a point in the input feature map, P_n represents the relative position of the convolutional kernel relative to the center point P_0, ΔP_n represents the calculated offset of each point in the convolutional kernel relative to their original positions, and w represents the weight of the convolution kernel.

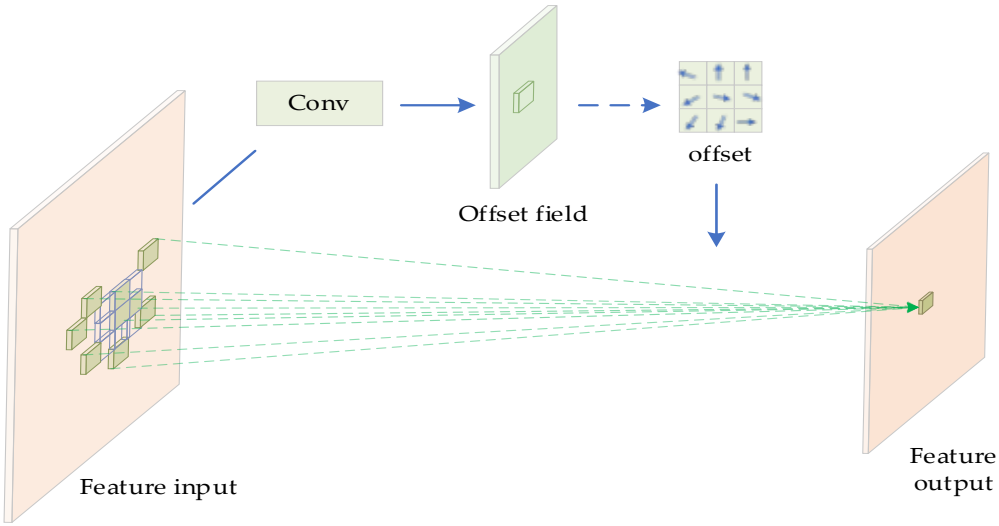


Figure 4. Deformable convolutional module.

Figure 4 shows that, compared with ordinary convolution kernels, deformable convolution kernels expand the receptive field and provide richer semantic information in the output feature maps. Features that originally required multiple layers of convolution to generate now require only one layer. This convolution method has a promoting effect on the detection of small objects.

3.2.3. Depthwise Separable Convolution Module

The left half of Figure 5 represents the depth convolution portion, which, in contrast to ordinary convolution kernels, produces feature maps of multiple channels via only one convolution operation. However, this operation performs a convolution procedure on each channel of the input layer independently and without effectively utilizing the feature information of different channels at the same spatial position. The right half of the figure utilizes the convolution kernel of 1×1 to obtain new features, fully integrating the information between multiple channels. However, there is no information exchange

between the internal points of the feature map with a single channel. The combination of the two convolution methods in Figure 5 fully utilizes the unique advantages of each convolution method in a complementary fashion. This convolution operation greatly reduces the number of model parameters compared with ordinary convolution operations.

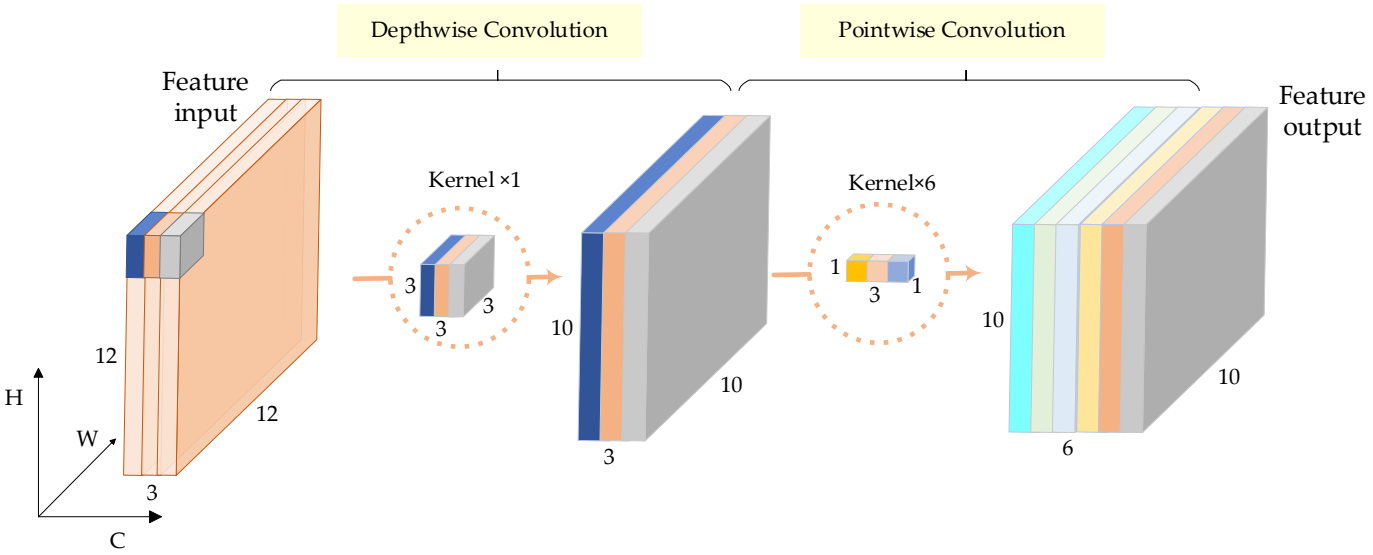


Figure 5. Depthwise Separable Convolutional Module.

3.2.4. Mish Activation Function

Equation (3) is the mathematical expression for the Mish activation function [33], and the function image drawn based on the mathematical expression is shown in Figure 6.

$$y(x) = x * tanh(ln(1 + e^x)) \quad (3)$$

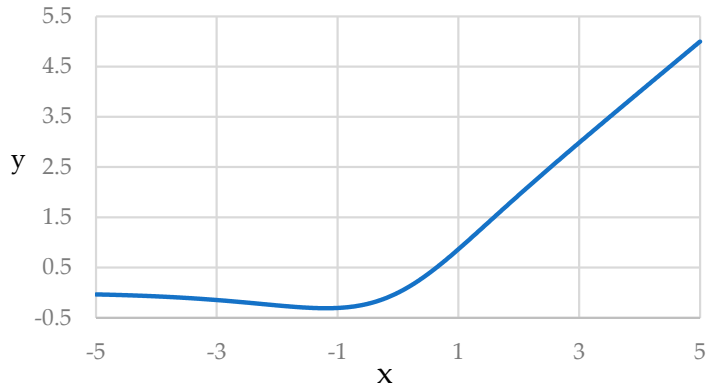


Figure 6. Mish activation function curve.

The curve of the Mish activation function for x within the range $[-5, 5]$ is shown in Figure 6. The Mish function increases without boundaries along the positive direction of the x-axis as x increases, avoiding the saturation caused by capping. Along the negative direction of the x-axis, the function value gradually approaches zero. This allows for better gradient flow with slight negative values, avoiding hard zero boundaries similar to those in the ReLU activation function [43]. The smooth activation function of the Mish function improves information penetration into the neural network, resulting in better accuracy and model generalization as a whole.

4. Dataset and Training Metrics

In this section, we will introduce and analyze the dataset used before explaining the relevant metrics of the training model.

4.1. Dataset Used

According to the drone patrol mission, primary drone targets include people, vehicles, and other types of transportation. The categories in the VisDrone2019 dataset are similar to those employed in our mission, which is the main reason why we chose the VisDrone2019 dataset for training. The VisDrone2019 dataset contains 10,209 static images (6471 for training, 548 for validation, and 3190 for testing) captured using drone platforms at different locations and heights. There are 10 predefined categories of objects in the dataset (pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle). Their statistical results are shown in Figure 7, and the bounding boxes of the different categories of objects in each image are manually annotated [44]. We conducted statistical analysis on the number of different types of objects used in the model training section of the dataset, and their distribution is shown in Figure 8.

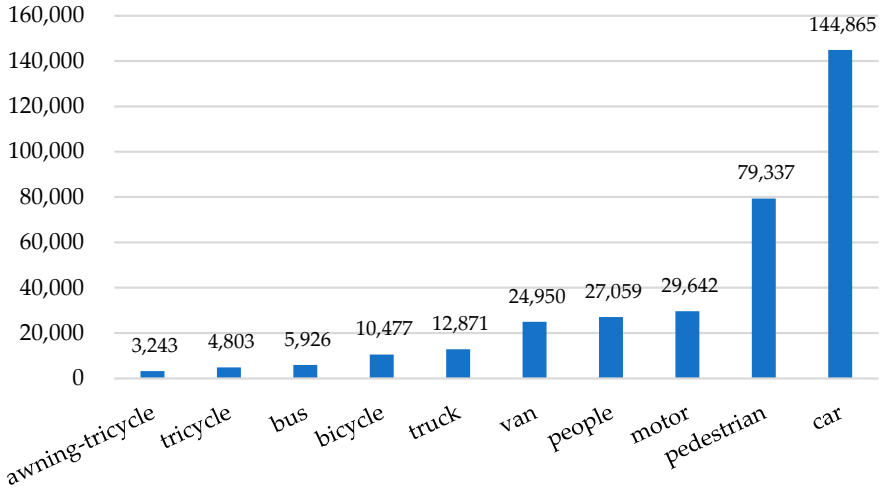
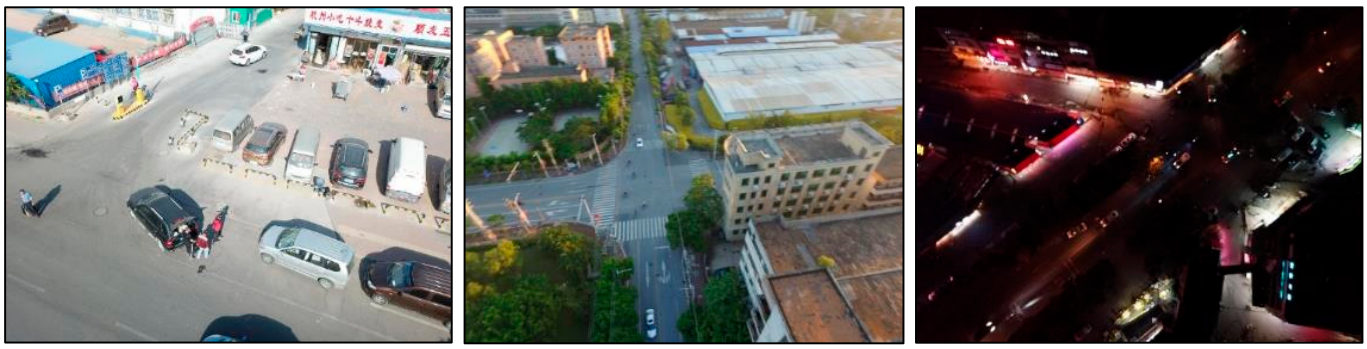


Figure 7. Quantity statistics of each category in the dataset. The objects labeled in the VisDrone2019 training dataset are divided into 10 types of objects. The blue column in the figure represents the total number of times the object corresponding to the label below appears in the VisDrone2019 training dataset. The specific values are labeled above the blue column.



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

Figure 8. *Cont.*

(**d**) (**e**) (**f**)

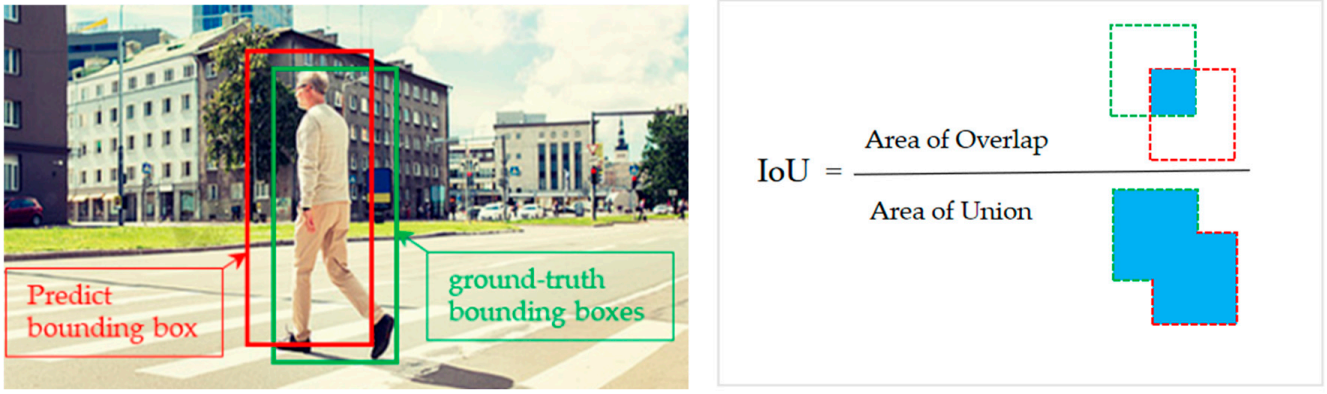
Figure 8. Partial image display of the VisDrone2019 dataset (see Supplementary Materials). Among them, subimages (**a**,**d**) show scenes captured using drones under strong lighting conditions, subimages (**b**,**e**) show scenes captured using drones under normal lighting intensity, and subimages (**c**,**f**) show scenes captured using drones under nighttime lighting conditions.

4.2. Related Metrics for Model Training

4.2.1. Intersection over Union

Intersection over Union (IoU) [43] is a standard for measuring the accuracy of corresponding object detection in a specific dataset. IoU is a simple measurement standard that can be used for tasks that derive a bounding box from the output.

In order to enable the use of IoU when detecting objects of any size or shape, as shown in Figure 9a, two conditions must first be met. Firstly, we require the artificially marked range of objects to be detected in images of the training set, known as "ground truth bounding boxes". Secondly, the range of results obtained using our algorithm is called the "predicted bounding boxes". The calculation method of IoU is shown in Figure 9b. In summary, this standard is used to measure the correlation between reality and the prediction. The higher the correlation is, the higher the value will be.



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

Figure 9. Schematic diagram of Intersection over Union. (**a**) The green markings are the correct ground-truth results artificially marked, while the red markings are the predicted results of the algorithm. (**b**) Graphical representation of IoU calculations.

4.2.2. Average Precision and Mean Average Precision

In Equation (4), P represents the precision, as shown in Figure 10, which refers to the proportion of positive predicted values to the ground truth. The larger the value is, the better the result will be. When the P value is equal to 1, an ideal state has been reached.

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{5}$$

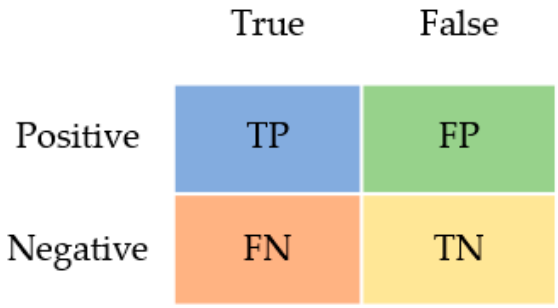


Figure 10. Graphical representation of the relevant metrics. True positives (TPs) represent the number of instances that have been correctly divided into positive cases; that is, the number of instances (samples) that are actually positive cases and are classified as positive cases by the classifier. False positives (FPs) represent the number of instances that were mistakenly classified as positive; that is, the number of instances that were actually negative but were classified as positive by the classifier. False negatives (FNs) represent the number of instances that were mistakenly classified as negative; that is, the number of instances that were actually positive but were classified as negative by the classifier. True negatives (TNs) represent the number of instances that were correctly divided into negative cases; that is, the number of instances that are actually negative cases and have been classified as negative cases by the classifier.

In Equation (5), R represents the recall, as shown in Figure 10, which refers to the proportion of all true samples that are predicted to be positive. The larger the value is, the better the result will be. When the R value is equal to 1 there is an ideal state.

Average precision (AP) [45], which represents the area under the P-R curve, is mathematically expressed as Equation (6):

$$AP = \int_0^1 P(R)d(R) \tag{6}$$

Mean average precision (mAP) [45], which represents the average value of multiple categories of AP, is mathematically expressed as Equation (7):

$$mAP = \frac{1}{\mathrm{classes}} \int_0^1 P(R)d(R) \tag{7}$$

4.2.3. Calculation of Loss Function

In Figure 11, the upper part describes the image input model, which generates an output of $10 \times 10 \times 48$, with 10×10 representing the division of the input graph using a grid of 10 rows and 10 columns. The number 4"8 in the figure is equal to $3 \times (4 + 1 + 11)$, where 3 represents the generation of 3 different borders for each grid; 4 represents the x, y, h, and c of the border, and their specific meanings are explained in the lower right corner of the image. Due to the 11 categories of the dataset used, 11 represents the probability value of the grid, where the center point of the border is predicted to belong to each category.

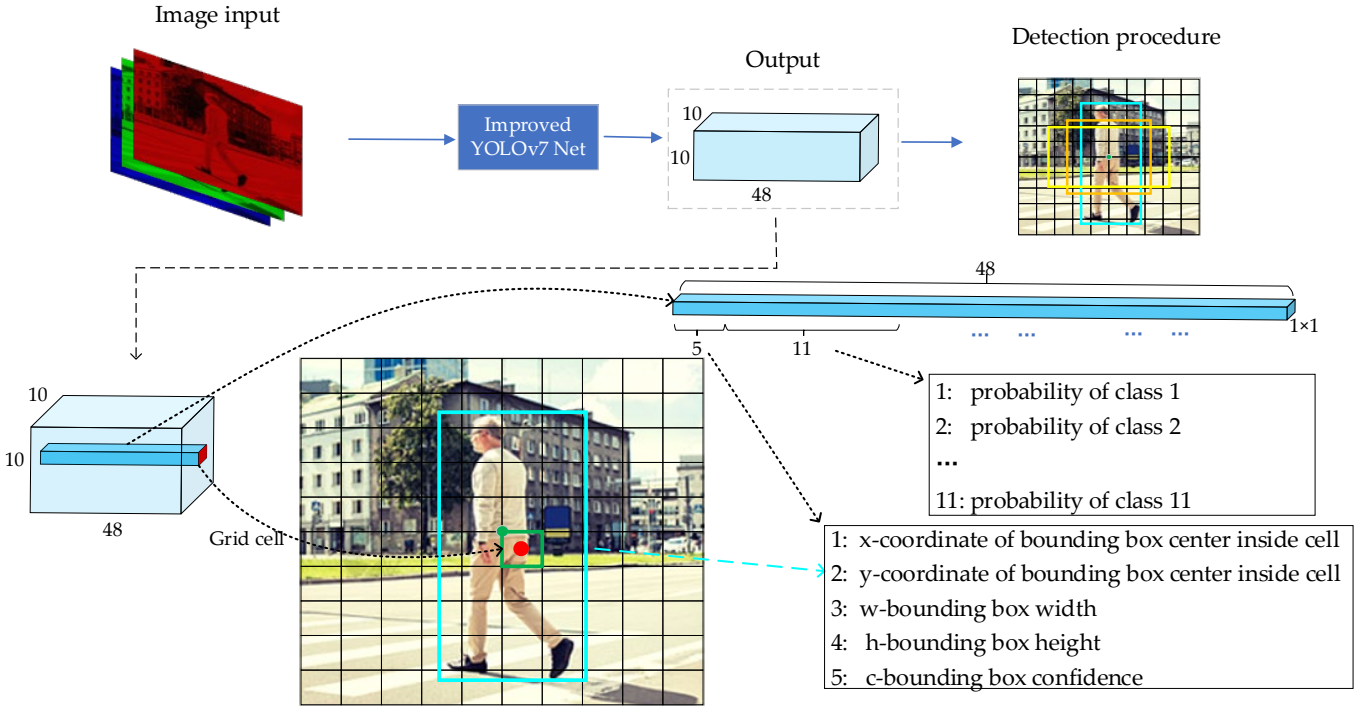


Figure 11. Schematic diagram of the network output.

According to the output of the network as shown in Figure 11, we used the EIoU [46] loss to calculate the border loss, L_{EIoU}, during the training process. We also calculated the label-smoothing loss, $L(p_i, \hat{p}_i)$, and the confidence loss. We summed them to obtain the total loss function, L_{net}. The mathematical expression for the EIoU loss is Equation (8):

$$L_{EIoU}^{ij} = 1 - IoU_{ij} + \frac{\rho^2(b_{ij}, b_i^{gt})}{(c_w^{ij})^2 + (c_h^{ij})^2} + \frac{\rho^2(w_{ij}, w_i^{gt})}{(c_w^{ij})^2} + \frac{\rho^2(h_{ij}, h_i^{gt})}{(c_h^{ij})^2} \tag{8}$$

In Formula (8), as shown in Figure 12, i represents the i-th grid on the graph, as shown in the image with a grid drawn at the bottom left in Figure 11, and j represents the j-th bounding box predicted for each grid. In this article, three bounding boxes are predicted for each grid i. The parameters b_{ij} and b_i^{gt} represent the center points of the predicted and actual bounding boxes for each grid, respectively, w_{ij} and w_i^{gt} represent the width of the predicted and actual bounding boxes for each grid, respectively, h_{ij} and h_i^{gt} represent the height of the predicted and actual bounding boxes for each grid, respectively, and ρ^2 represents the Euclidean distance between the two points. The parameter c represents the diagonal length of the minimum bounding rectangle that contains both the predicted bounding box and the actual bounding box. Among them, c_w^{ij} and c_h^{ij} are the width and height of the minimum bounding rectangle covering the two boxes. The advantage of EIoU loss is that, in the model, the aspect-ratio loss term is split into the difference between the width and height of the predicted bounding box and the width and height of the minimum bounding rectangle. This accelerates the convergence and improves the regression accuracy compared with ordinary bounding-box losses.

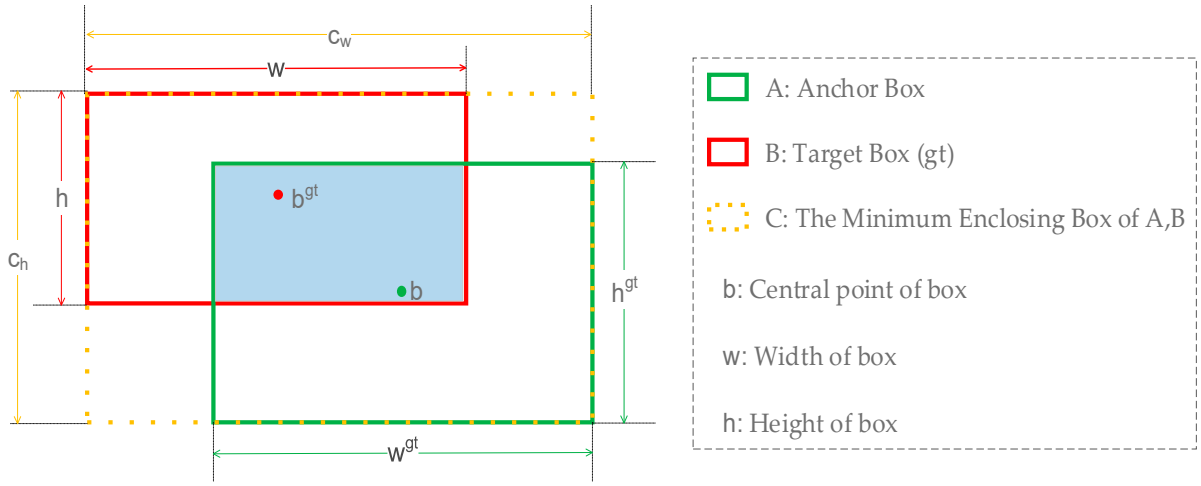


Figure 12. Schematic representation of notations in the EIoU loss equation.

The specific mathematical expressions for the label-smoothing loss [47] are Equations (9) and (10):

$$\hat{p}_{ik} = \begin{cases} 1-\varepsilon & if\ k = i, \\ \varepsilon/(K-1) & otherwise. \end{cases} \tag{9}$$

$$L(p_{ij}, \hat{p}_i) = -\sum_{k=0}^{K} \hat{p}_{ik} log p_{ijk} \tag{10}$$

In Equation (9), as shown in Figure 13, $\hat{p}_{ik}$ represents the label value of the k-th category corresponding to the i-th grid, which is the probability value of the k-th category, and ε is a small constant with a value range between 0 and 1. In this article, we take ε as 0.2 and K as the total number of categories. The K value is 11. In Equation (10), p_{ijk} represents the predicted value of the k-th category corresponding to the j-th predicted bounding box of the i-th grid.

	One-hot			Label smoothing		
	Label 1	Label 2	Label 3	Label 1	Label 2	Label 3
Class 1	1	0	0	0.80	0.02	0.02
Class 2	0	1	0	0.02	0.80	0.02
Class 3	0	0	0	0.02	0.02	0.02
Class 4	0	0	0	0.02	0.02	0.02
Class 5	0	0	0	0.02	0.02	0.02
Class 6	0	0	0	0.02	0.02	0.02
Class 7	0	0	1	0.02	0.02	0.80
Class 8	0	0	0	0.02	0.02	0.02
Class 9	0	0	0	0.02	0.02	0.02
Class10	0	0	0	0.02	0.02	0.02
Class11	0	0	0	0.02	0.02	0.02

Figure 13. Schematic diagram of one-hot and label smoothing. The numerical value in the figure represents the specific probability value of the label belonging to a certain category. The left side of the figure shows the label format in one-hot encoding format, while the right side of the figure shows the label format in label-smoothing encoding.

Compared with using one-hot encoding for multi-classification tasks, label-smoothing measures avoid the problem of manual labelling errors for specific objects, which can cause significant harm to the model. This is because, during the training process, a non- class sample is forcibly learned and its probability is very high, affecting the estimation of a posterior probability. Additionally, there is sometimes an incomplete correlation between classes. If the probability of an encouraging output varies too much, this can lead to a certain degree of overfitting. This means that, to some extent, the loss of label smoothing avoids overfitting and alleviates the impact of incorrect labels.

The mathematical expression for the total loss, L_{net}, of the model is Equation (11):

$$L_{net} = \lambda_1 \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} L_{EIoU}^{ij} + \lambda_2 \sum_{i=0}^{S^2} I_{ij}^{obj} L(p_{ij}, \hat{p}_i) + \lambda_3 (\sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{obj} (c_i - \hat{c}_i)^2 + \lambda_g \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{noobj} (c_i - \hat{c}_i)^2) \quad (11)$$

In Equation (11), λ_1, λ_2, and λ_3 are the proportional coefficients of the bounding-box loss, confidence loss, and category loss, respectively. In this paper, their values are 0.2, 0.3, and 0.15 [23], respectively. The parameter λ_g represents the coefficient of the negative sample confidence loss value, which is taken as 0.05 in this article; S^2 represents the total number of grids; B represents the total number of predicted bounding boxes corresponding to each grid (it has a value of 3 in this article); $\hat{c}_i$ is the confidence value corresponding to the i-th grid; and c_i is the confidence value corresponding to the j-th prediction border selected in the i-th grid.

5. Experimentation and Results

In this section, we introduce the experimental setup, data processing, and training process of the improved model. We analyze the data during the training process, verify the impact of each module on model improvement through ablation experiments, and then compare and analyze the experiment in comparison with other models. Finally, we present and analyze some of the detection results of the model.

5.1. Experimental Setup

We present the specific experimental settings for the model training in Tables 1 and 2.

In Table 1, the environmental configuration during the experimental process is described. We experimentally trained the model on NVIDIA GeForce RTX 6000 Ada using the GPU driver of Windows 10. We used an environment of Python 3.8.16 and a torch 1.13.1 + cu116. We used a CPU model of Intel (R) Xeon (R) w9-3495X with a RAM capacity of 64 GB.

Table 1. Experimental environment.

Experimental Parameters	Value
Operating system	Windows 10
Deep-learning framework	Pytorch (torch 1.13.1 + cu116)
Programming language	Python 3.8.16
CPU	Intel(R) Xeon(R) w9-3495X
GPU	NVIDIA GeForce RTX 6000 Ada
RAM	64 G

In Table 2, the specific values of the hyperparameters during the training process are shown. All the experiments described in this article were set to train for 230 periods, with a batch size of 16. The image size of the input model during model training was 640 × 640, with a learning rate of 0.01, SGD momentum [48] of 0.937, and optimizer weight decay of 0.0005. All other training parameters were set to the default values of the YOLOv7 network.

Table 2. Hyperparametric configuration.

Hyperparameters	Value
Learning Rate	0.01
Image Size	640 × 640
Momentum	0.937
Optimizer	SGD
Batch Size	16
Epoch	230
Weight Decay	0.0005

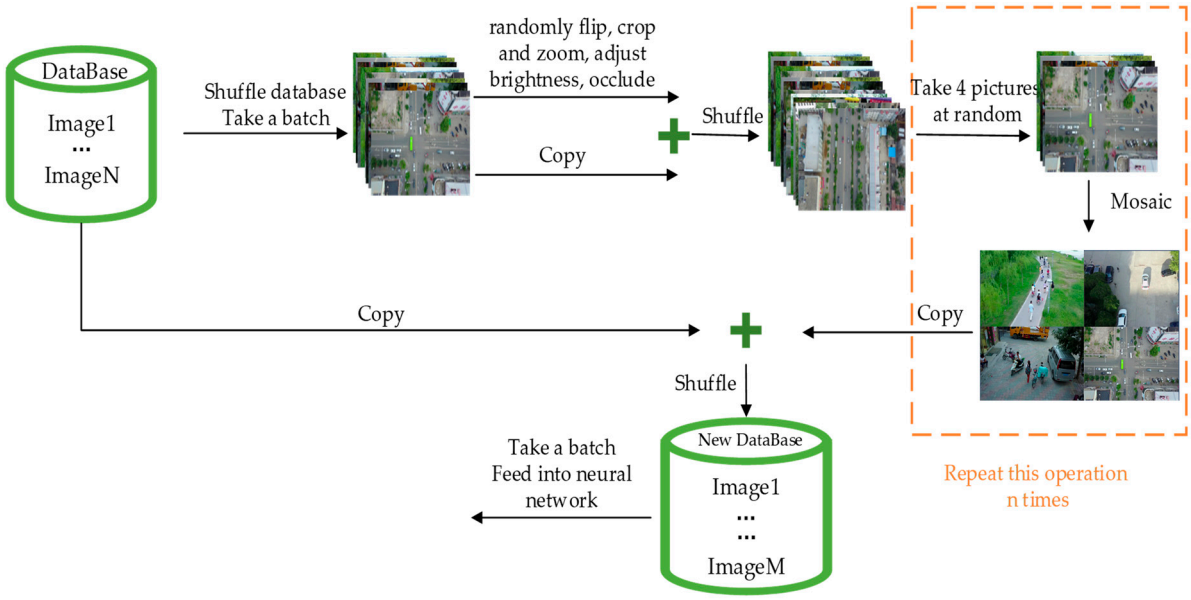
5.2. Image-Data Preprocessing

For the dataset, we used the Mosaic method [49] for image-data augmentation, as shown in Figure 14. The principal concept behind Mosaic's is to randomly crop four images and then concatenate them onto one image as training data. There are four benefits to doing this. Firstly, it is possible to increase data diversity by randomly selecting four images for combination, resulting in a larger number of images than the original image. Secondly, enhancing the robustness of the model by mixing four images with different semantic information can enable the model to detect targets beyond the conventional context. Thirdly, strengthening the effect of the batch normalization layer. When the batch normalization (BN) [50] operation is set in the model, the total number of batch samples increases as much as possible during training. This is because the BN principle is to calculate the mean and variance of each feature layer. If the total number of batch samples is larger, the mean and variance calculated by BN will be closer to the mean and variance of the entire dataset, and the effect will be better. Finally, the Mosaic data-augmentation algorithm is beneficial for improving the performance of small-target detection. Mosaic processes image data by concatenating four original images, which increases the probability of each image containing small objects.

5.3. Training Procedures

The overall process of model training is shown in Figure 15, whereby the pretraining weights are loaded using transfer learning, which accelerates the convergence of the model and shortens the training time. We adopted a fine-tuning strategy to train the model. This involved freezing and activating the relevant layers of the network throughout the entire training process to accelerate the convergence speed of training.

During the training process, we made multiple attempts to set hyperparameters, and the hyperparameters in Table 2 were the good results obtained in our experiment. We have also attempted training from scratch and based on pretraining weights. The results of multiple attempts have shown that the training effect is best in the pretraining mode. We have used pretraining methods in all the experiments. At the same time, in response to device limitations, we attempted different batch sizes, and the batch sizes in Table 2 were also the best results from the experiment.



(**a**)

(**b**)

(**c**)

Figure 14. Mosaic operation and result display. The subgraph (**a**) represents the process of Mosaic image-data augmentation, while (**b**,**c**) represent four randomly selected images from VisDrone2019 that have been processed using Mosaic. The green rectangular boxes in the two images represent the labels of bounding boxes in the original image, which are displayed in the synthesized image through corresponding transformation processing.

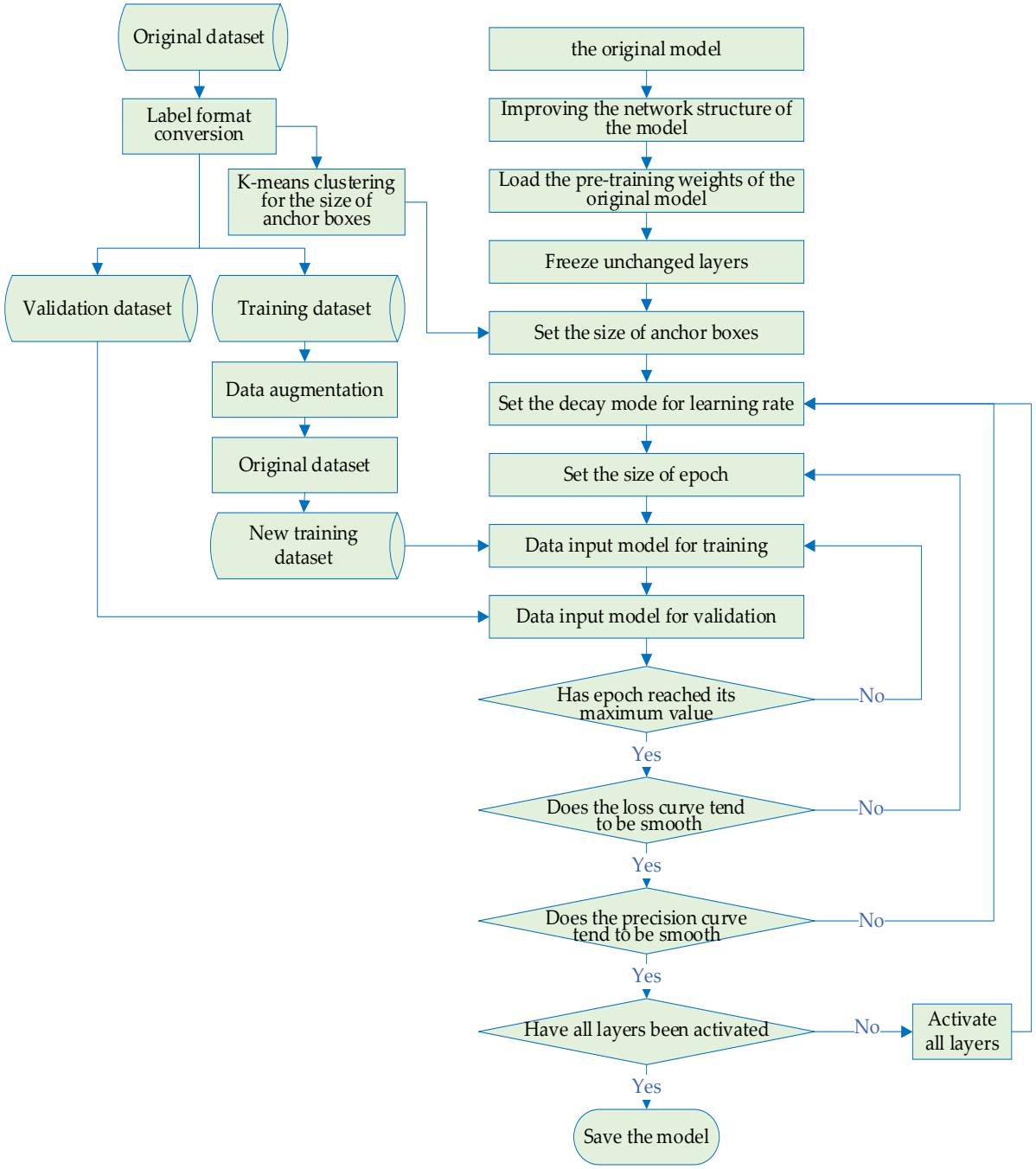


Figure 15. Training flowchart.

5.4. Result and Analysis

We trained the improved model, and Figures 16 and 17 show the relevant data statistical results.

In Figure 16, the three subgraphs—(a), (b), and (c)—represent the border loss, confidence loss, and category loss, respectively, of the improved model on the training dataset during the training process. It can be observed from the graph that, when the epoch value is greater than 200, the size of the three types of loss values tends to be stable. The three subgraphs, (e), (f) and (d), represent, respectively, the bounding-box loss, confidence loss, and category loss of the model on the validation dataset during the training process, which is different from the validation set. When the epoch is greater than 100, the three types of loss values start to stabilize. Our explanation for the difference in convergence speed between

the training set and the validation set is that the validation set has a smaller amount of data and covers fewer types of scenarios. In the less-trained epoch, the model already learns the internal relationships of the data. For training datasets, due to the large amount of data and more diverse and complex scenarios, learning internal data relationships takes longer, requiring more epochs to stabilize various loss values. The precision curves and recall curves of the models in the two subgraphs (g) and (h) of the training dataset are shown in the figure. Both curves generally increase with the increase in the training epoch, but there are also certain fluctuations during the increasing process. Analyzing the reasons, the model approximates the physical model through backpropagation and gradient updates, and each update calculation utilizes small batches of data instead of the entire large dataset. Therefore, the precision and recall of the model may fluctuate at each epoch. However, the model as a whole is constantly converging. Indeed, when an epoch is greater than 150, both curves tend to stabilize. The two curves also reflect the efficiency and effectiveness of using small-batch strategies to train the model.

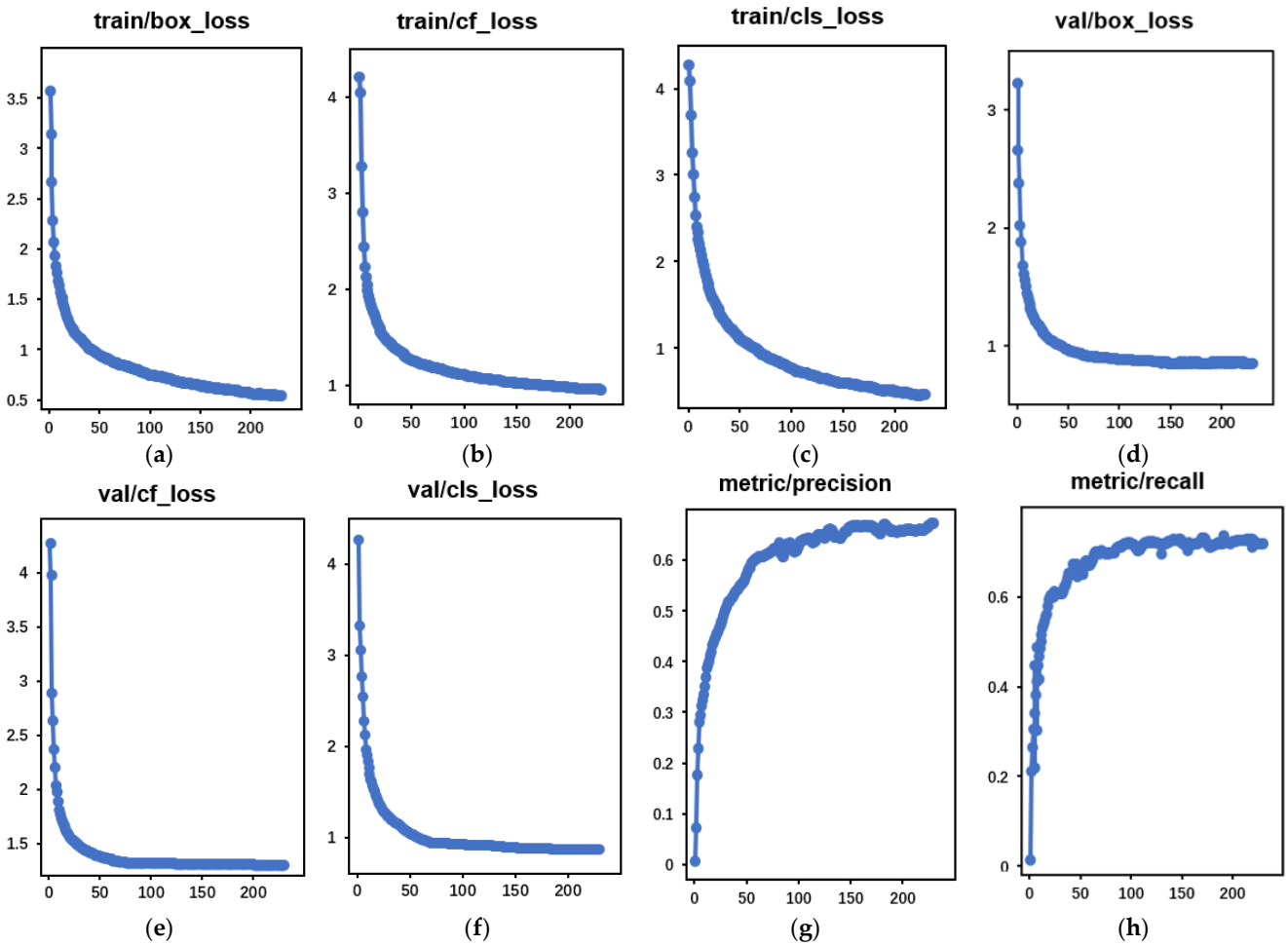


Figure 16. Model training results. The horizontal axis in the figure represents the value of the epoch, which ranges from 0 to 230. The vertical coordinates in the figure represent the specific values of the corresponding types of data. (**a**) The bounding box loss curve of the training set; (**b**) The confidence loss curve of the training set.; (**c**) The classification loss curve of the training set; (**d**) The bounding box loss curve of the validation set; (**e**) The confidence loss curve of the validation set.; (**f**) The classification loss curve of the validation set; (**g**) The precision curve; (**h**) The recall curve.

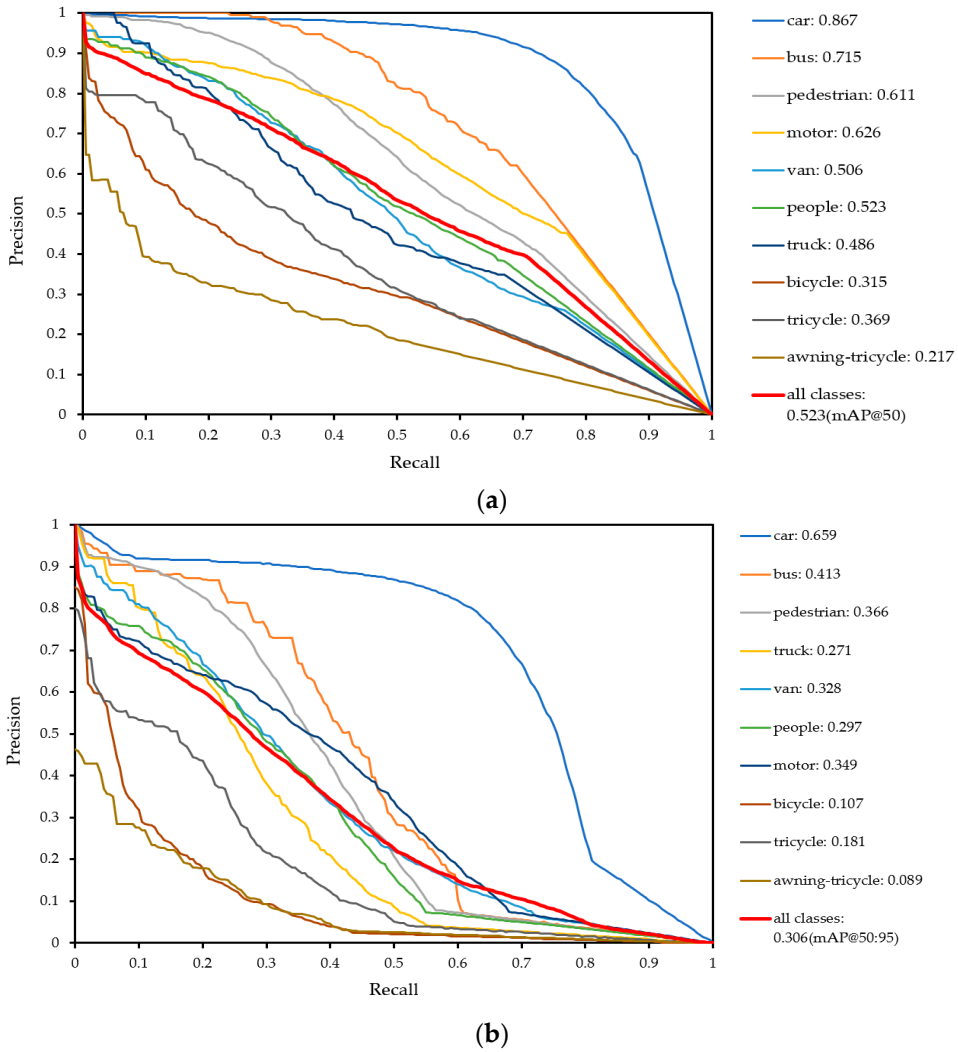


Figure 17. Precision–recall curves of various types of objects in the modified model. (**a**) The precision–recall curve of each type of object being detected when the IoU threshold is 0.5. (**b**) When the mAP threshold is between 0.5 to 0.95, with an interval of 0.05, the precision–recall curve of each type of object being detected yields 10 mAP values, as determined by taking the average of 10 values.

Figure 17 shows the precision–recall curves for each category, with curves of different colors representing different categories. The bold red curve represents the average value of AP for all the categories. In the legend on the right side of each subgraph, the number after the corresponding category label represents the AP value for each category. From Figure 17, it can be seen that the precision–recall curves vary for different types of objects, with cars having the highest AP and awning-tricycles having the lowest. Based on the characteristics of the dataset, it can be seen that the total number of cars is the highest, while the total number of awning-tricycles is the lowest. From Figure 17, we can conclude that the total quantity of training data to some extent determines the quality of the network training. The recognition effect of the network on various objects can be improved by increasing the accuracy of the network model.

5.5. Ablation Experiment

In response to the modules proposed by our model framework, we utilized ablation experiments to explore the impact of deformable attention (DA), deformable convolution (DC), and depthwise separable convolution (DSC) modules on the accuracy of the model and the size of the model parameters. The specific experimental results are shown in Table 3.

Table 3. Ablation experiments with the modules.

DA	DC	DSC	mAP_{50} (%)	$mAP_{50:95}$ (%)	Param
√			50.3	29.1	37.9 M
	√		49.1	28.6	35.6 M
		√	49.6	28.9	36.1 M
√	√		51.1	29.7	36.6 M
√	√	√	52.3	30.6	35.8 M

In Table 3, the symbol √ indicates whether the module corresponding to its column has been added to the model, DA represents the deformable attention module, DC represents the deformable convolution module, DSC represents the depthwise separable convolution module, mAP_{50} and $mAP_{50:95}$ represent the model test values under different IoU values, and Param represents the size of the model parameter quantity. When the model is improved using three modules separately, there is a certain degree of reduction in the parameter quantity and mAP value of the model, among which deformable convolution and depthwise separable convolution have the most significant reduction in the parameter quantity of the model. The deformable attention module significantly improves the detection accuracy of the model. When the three models are combined, the overall accuracy of the model is significantly improved, and the total number of model parameters is also significantly reduced. The ablation experiment has verified the effectiveness of our proposed improvement strategy for the model.

5.6. Comparison Experiment

Similarly, we compared the values of mAP_{50} and $mAP_{50:95}$ before and after model modification on the VisDrone2019 dataset to generate the curve shown in Figure 18. We analyzed the test results of the improved model on the test dataset according to different target types and obtained the confusion matrix shown in Figure 19.

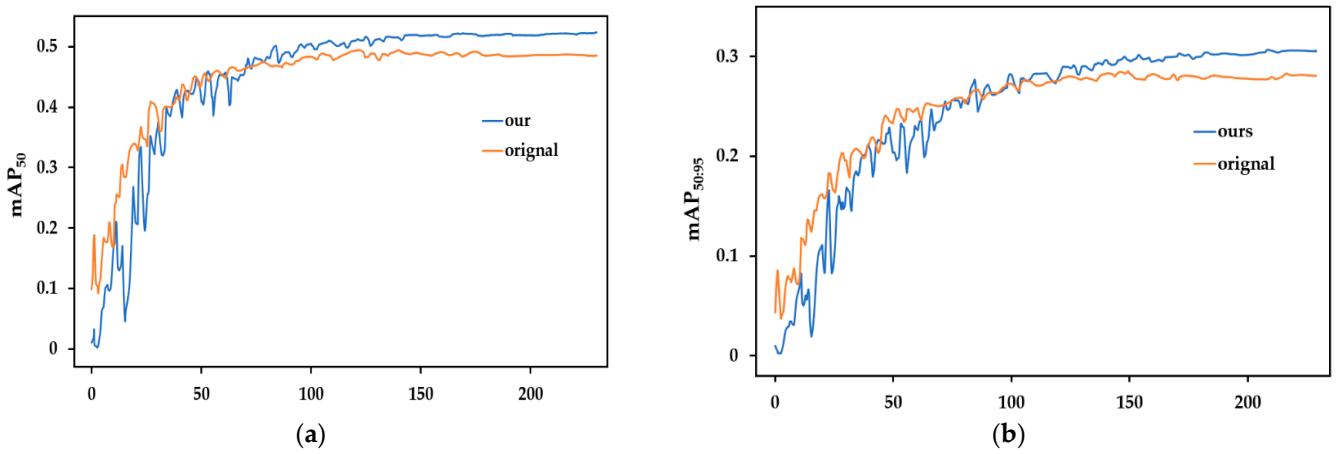
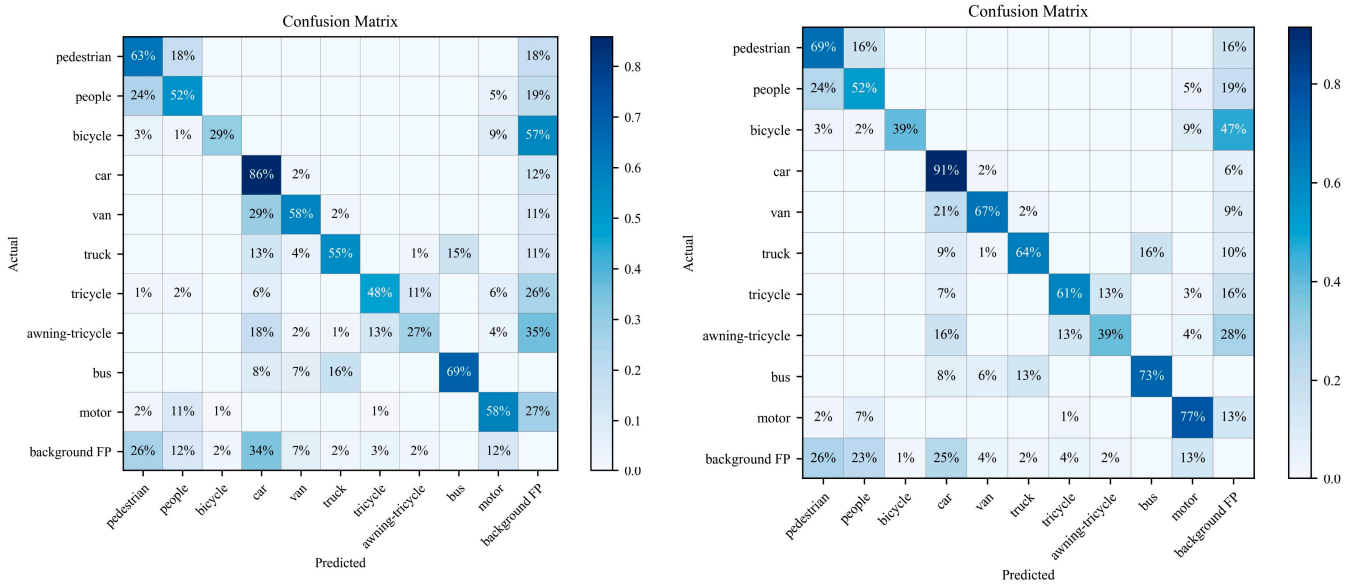


Figure 18. Comparison of the mAP before and after model modification. The horizontal axis in the figure represents the epoch of training, the vertical axis represents the magnitude of mAP values, the orange-red curve represents the period before the model improvement, and the blue curve represents the period after the model improvement. (**a**) Comparison of the mAP_{50} before and after model modification. (**b**) Comparison of the $mAP_{50:95}$ before and after model modification.



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

Figure 19. Comparison diagram of the confusion matrix before and after model modification. The horizontal axis in the figure represents the type of target predicted by the model, while the vertical axis represents the actual type of target. The numerical values in the grid represent the proportion of targets of the type in the row predicted as targets of the type in the column. The color of the grid is a visual display of the numerical value; the larger the numerical value in the grid, the darker its color. (**a**) The confusion matrix of the original model. (**b**) The confusion matrix of the improved model.

From the orange-red curve in Figure 18, it can be seen that the model tends to stabilize when the epoch is greater than 100 prior to modification. For the blue curve, which is the improved model, convergence begins when the epoch is greater than 200. Although the convergence speed of the model slows, there is a significant improvement in the detection mAP of the model.

In Figure 19, through the comparison of subgraphs (a) and (b), it can be seen that using the improved model improved the prediction accuracy of each category, with the largest improvement being the motor, which attained an accuracy improvement of 19%. The probability of each category being missed decreased, with cycles, tricycles, and motors all decreasing by more than 10%. From a visual perspective, when comparing the two subgraphs as a whole, we can see that subgraph (b) is sparser than subgraph (a). This is because there is a significant reduction in the number of misjudgments in other categories when using the improved model to make predictions. At the same time, we can also see from the graph that there is a high misjudgment rate between pedestrian and people, car and van, and tricycle and awning-tricycle. This may be due to the fact that when the object is small in the image, their appearance is similar, and the model's discrimination ability is low.

To further validate the effectiveness of the model improvement, we compared the mAP values obtained from different models on the VisDrone2019 dataset to form Table 4.

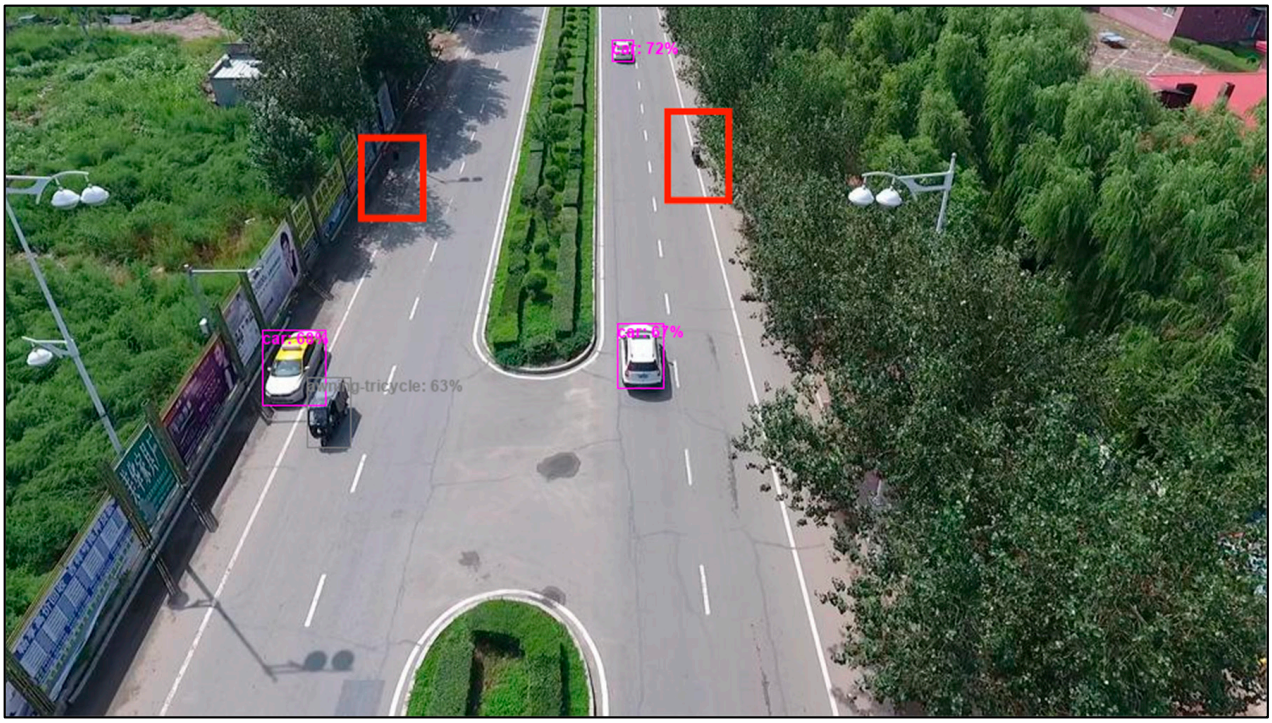
From Table 4, it can be seen that the improved model has a 7.63% improvement compared with YOLOv7, which has the highest mAP_{50}, and an 8.93% improvement compared with $mAP_{50:95}$. By comparison with other models, we further demonstrate that our improvements to the model are effective for the VisDrone2019 dataset. Additionally, because our application is similar to that in the VisDrone2019 dataset, these results indirectly prove that our improvements to the model contribute to the improvement of object-detection performance during the performance of drone patrols.

Table 4. Comparison experiments.

Algorithm	mAP_{50} (%)	$mAP_{50:95}$ (%)
YOLOv3 [18]	39.28	22.07
YOLOv4 [19]	30.91	18.42
YOLOv5l [20]	41.41	24.36
YOLOv7 [23]	48.50	28.10
TPH-YOLOv5 [51]	44.05	26.08
PP-YOLOE [52]	39.60	24.64
TA-GU YOLOv7 [35]	48.79	24.57
Ours	**52.35**	**30.61**

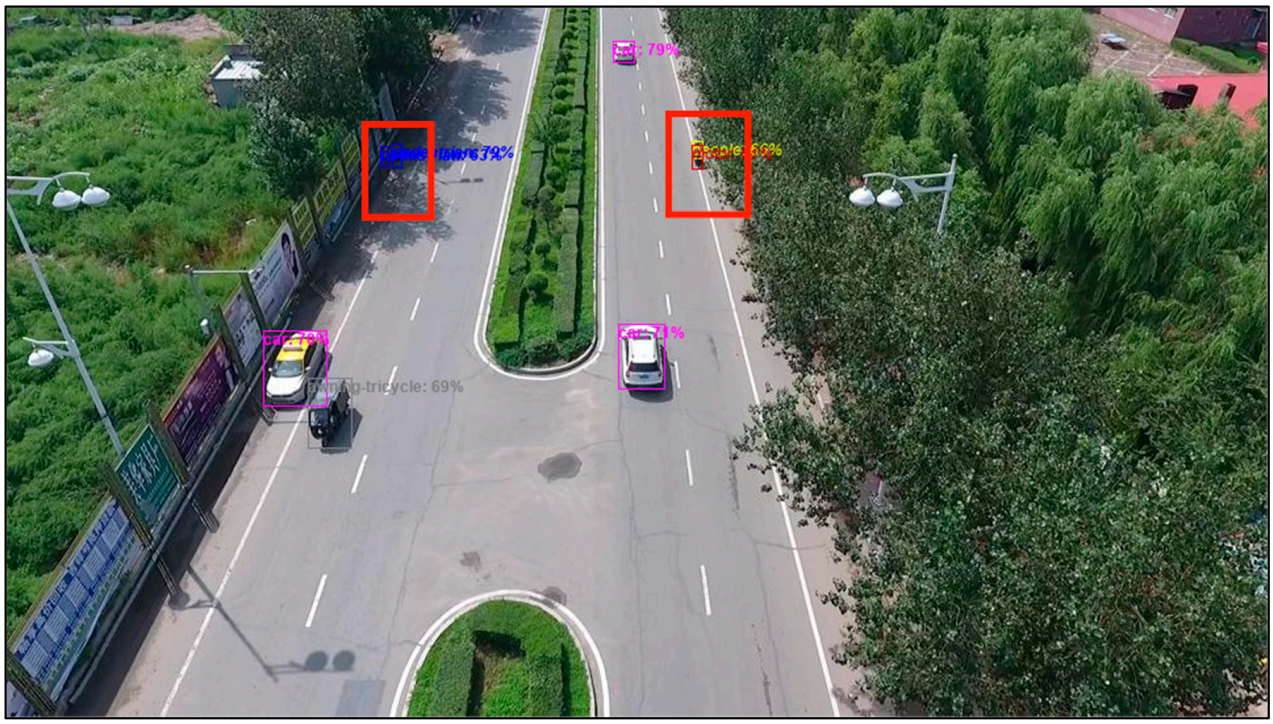
5.7. Visualization and Discussion

We tested the validation set on the VisDrone2019 dataset using both pre-improved and post-improved models, and some of the test results are shown in Figures 20–23. In Figures 20–23, the targets detected by the model are marked with a rectangular border, and the category labels of the targets are displayed in the rectangular box. At the same time, the predicted confidence values of the model for the targets are also displayed immediately following the category labels of the model. We used different colors for labeling different categories. For each image, we highlighted the differences in model-detection performance using red and bold rectangular boxes.



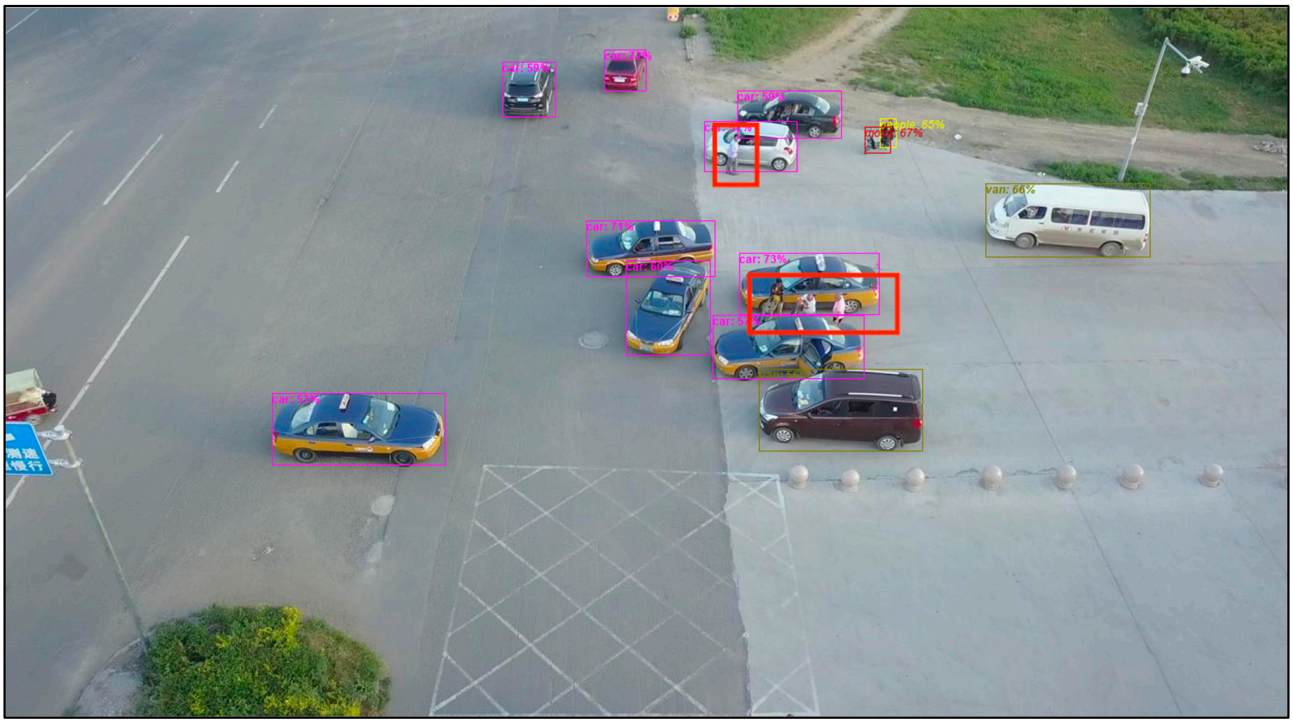
(**a**)

Figure 20. *Cont.*



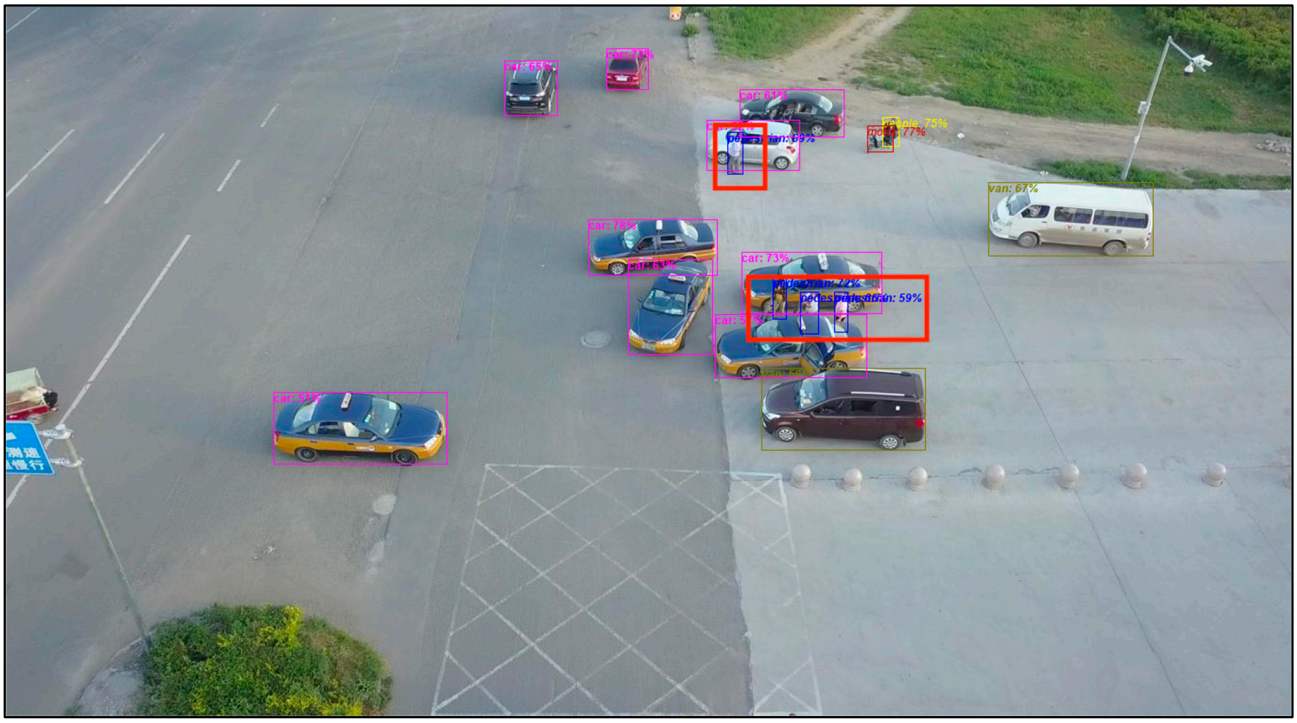
(**b**)

Figure 20. Comparison of model-detection results before and after the model improvement. (**a**) detection results of original model; (**b**) detection results of improved model.



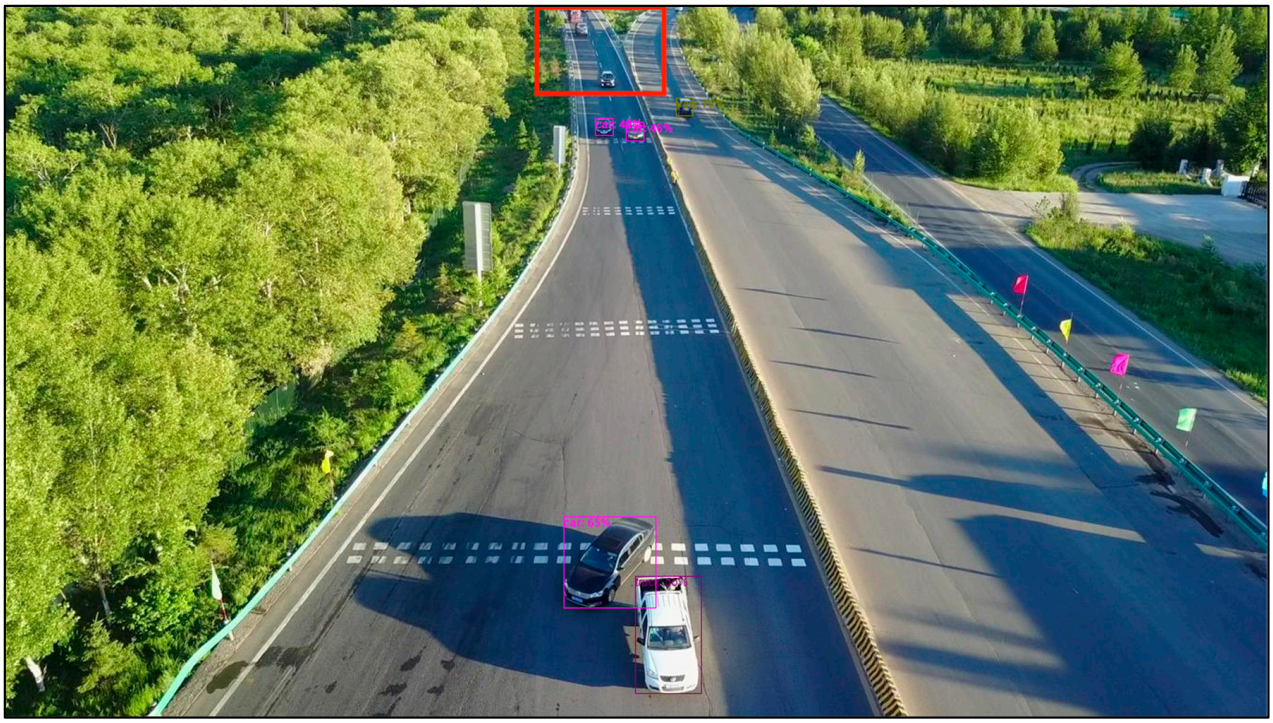
(**a**)

Figure 21. *Cont.*



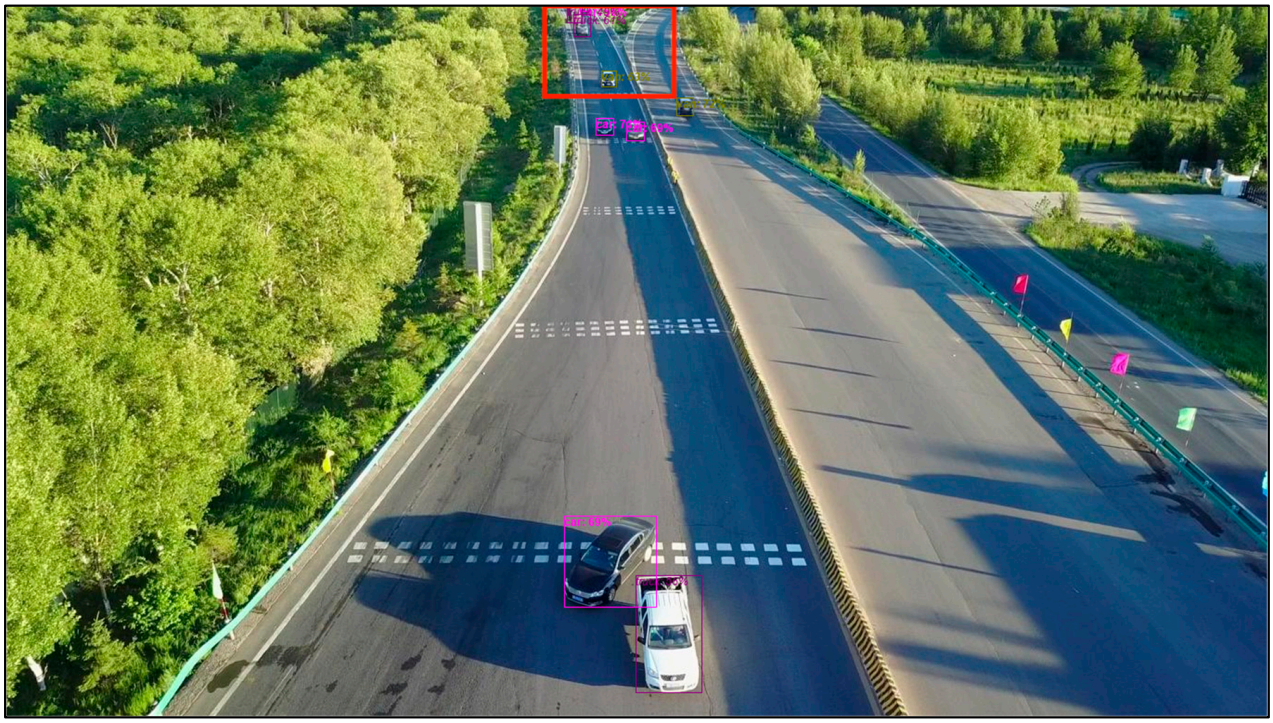
(**b**)

Figure 21. Comparison of model-detection results before and after the model improvement. (**a**) detection results of original model; (**b**) detection results of improved model.



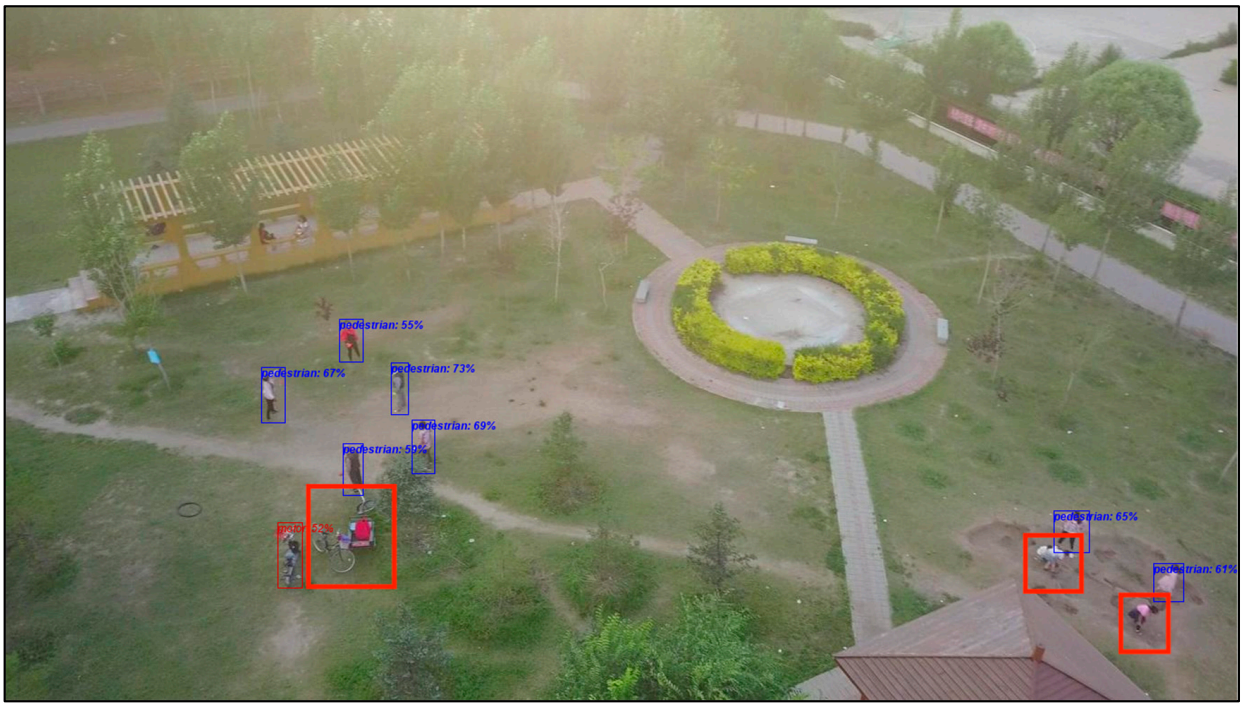
(**a**)

Figure 22. *Cont.*



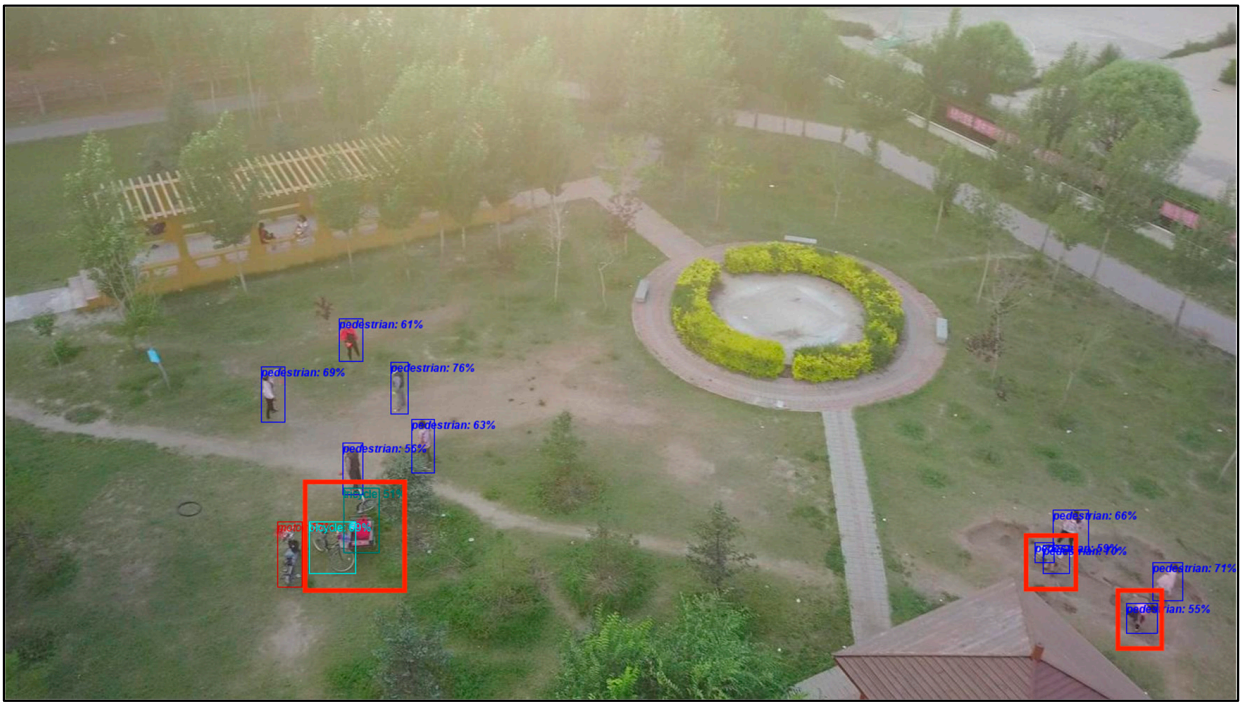
(**b**)

Figure 22. Comparison of model detection results before and after the model improvement. (**a**) detection results of original model; (**b**) detection results of improved model.



(**a**)

Figure 23. *Cont.*



(**b**)

Figure 23. Comparison of model detection results before and after the model improvement. (**a**) detection results of original model; (**b**) detection results of improved model.

In Figure 20, in the red rectangular boxes on both sides of the road in subgraphs (a) and (b), the improved model missed the detection of pedestrians and vehicles in the figure, while the improved model made accurate predictions for these objects. Analyzing the reasons as to why this occurred, we established that it is easy for pedestrians and vehicles on both sides of the road to miss detection due to vegetation obstruction and dim lighting. The same improved model overcomes the problems of occlusion and dim light and detects all these objects.

In Figure 21, in the red rectangular boxes next to the car in subgraphs (a) and (b), the people in the figure were missed before the model improvement. However, the improved model gave accurate predictions for these objects. Analyzing the reasons, it is easy for people next to cars to be missed during inspections because the color of their clothes is similar to that of the vehicle. The same improved model overcame such problems and successfully detected these objects.

In Figure 22, in the red rectangular boxes near the center of the upper edge in subgraphs (a) and (b), the vehicles in the figure were missed prior to the model improvement, but the improved model made accurate predictions for these objects. We analyzed the reasons for this, considering how this could have occurred when the objects in the distance and near areas in the picture belonged to the same category. We found, however, that their sizes differed greatly, making them easy to be missed in the detection. The same improved model overcame such problems and successfully detected these objects.

When analyzing the red rectangular boxes in Figure 23, near the center of the upper edge in subgraphs (a) and (b), the bicycles, tricycles, and some people in subgraph (a) were missed before the model improvement. The improved model in subgraph (b) accurately predicted these objects. Analyzing the reason behind this, we found that detection failed due to the large difference in shape between the objects in the picture, especially the tricycle, and the objects captured at a normal angle due to the shooting angle. For those who missed being detected, by careful observation, it was noted that they were all children and in a bent state; thus, the large deformation of the human body caused the missed detection. The same improved model overcame these problems and successfully detected these objects.

Although the model has achieved certain results, due to the limitations of the dataset size, some scenarios, for example, weather conditions such as rain, snow, and fog, are not present in the dataset. Based on supervised learning, the application of the model in these scenarios will be relatively poor. At the same time, based on the drone platform, when the collected images experience shaking, ghosting, and other blurring factors, the model may not achieve ideal results due to not being trained on this. For some disguised targets, the application of the model may also be limited when the target is close to the background.

6. Conclusions

It is crucial to improve the accuracy of small-scale object detection based on drone detection during patrols in control areas as this determines the quality and effectiveness of patrols. Without increasing the scale of the YOLOv7 model, this article proposed an improved YOLOv7 model based on combining the deformable attention, a deformable convolution module, and a depthwise separable convolution module. By adding these modules to the original model's network, the association between distant pixels was increased, and the semantic information extraction of small objects was strengthened. The model could more effectively extract features, improve the detection accuracy of the model, and, to some extent, reduce the number of model parameters. For model training, we adopted pretraining and fine-tuning techniques. The model, trained on the VisDrone dataset, achieved an mAP_{50} of 52.3%. Compared with the original YOLOv7 model, the parameter quantity was reduced by 2.98%, and the mAP_{50} and $mAP_{50:95}$ were relatively improved by 7.63% and 8.93%, respectively. Classification errors of the categories were also more concentrated, and there was a significant improvement in detecting small objects, such as pedestrians, in the image under occluded and fuzzy conditions.

However, the overall parameter size of the model still needs to be further reduced. The dataset used in the model was relatively small compared with the scene, with most of it being artificial landscapes such as highways, communities, parks, etc., and with limited inclusion of other types of natural landscapes such as lakes, grasslands, and hills. Moreover, the scenarios in the dataset were mostly collected during the spring and summer periods, and the weather was clear. The singularity of data affects the actual usage range. In future research, we will use more effective modules and methods to improve the model, further reducing the number of parameters needed to meet the requirements of embedded deployment. We will collect more background images for practical applications, including different plant backgrounds in different seasons, such as in autumn and winter, and different backgrounds in different weather conditions, such as fog, rain, and snow, in order to further train the model and increase its reliability.

Supplementary Materials: The following supporting VisDrone2019 dataset can be downloaded at: https://github.com/VisDrone/VisDrone-Dataset.

Author Contributions: Conceptualization, D.Z. and F.S.; methodology, F.S. and Q.L.; software, H.Z.; validation, Q.L. and X.L.; formal analysis, F.S. and L.Y.; investigation, Z.Z.; resources, F.S.; data curation, D.Z. and L.Y.; writing—original draft preparation, D.Z.; writing—review and editing, D.Z. and F.S.; project administration, Q.L.; funding acquisition, F.S. and X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Nature Science Foundation of China (grant number: 61671470).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable.

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

References

1. Yoon, C.-H.; Kang, J.M.; Park, J. A Study on the Direction of Technology Introduction of Drone Patrol. *Archives* **2019**, *81*, 2439–2442.

2. Liu, J.; Li, D.-W. A Drone Patrol System for Target Object Counting and Geolocalization. In Proceedings of the 2021 18th International Conference on Ubiquitous Robots (UR), Gangneung, Republic of Korea, 12–14 July 2021; pp. 357–362.
3. Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 20–25 September 2002; Volume 1, p. I-I.
4. Viola, P.A.; Jones, M.J. Rapid Object Detection using a Boosted Cascade of Simple Features. In Proceedings of the Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001.
5. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, San Diego, CA, USA, 20–25 June 2005.
6. Forsyth, D. Object Detection with Discriminatively Trained Part-Based Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: New York, NY, USA, 2014.
7. Girshick, R.; Iandola, F.; Darrell, T.; Malik, J. Deformable Part Models are Convolutional Neural Networks. In Proceedings of the Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Comm. ACM* **2012**, *60*, 84–90. [CrossRef]
9. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv* **2013**, arXiv:1312.6229.
10. Girshick, R.B.; 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, Vancouver, BC, Canada, 17–24 June 2013; pp. 580–587.
11. Girshick, R.B. Fast R-CNN. *arXiv* **2015**, arXiv:1504.08083.
12. Ren, S.; He, K.; Girshick, R.B.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transact. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef] [PubMed]
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
14. Berg, A.C.; Fu, C.Y.; Szegedy, C.; Anguelov, D.; Erhan, D.; Reed, S.; Liu, W. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
15. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2016**, arXiv:1612.03144.
16. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007.
17. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
18. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
20. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Hogan, A.; Hajek, J.; Diaconu, L.; Kwon, Y.; Defretin, Y.; et al. Ultralytics/yolov5: v5.0—YOLOv5-P6 1280 Models, AWS, Supervise.ly and YouTube Integrations. Available online: https://www.semanticscholar.org/paper/ultralytics-yolov5%253A-v5.0-YOLOv5-P6-1280-models%252C-and-Jocher-Stoken/fd550b29c0efee17be5eb1447fddc3c8ce66e838 (accessed on 29 October 2023).
21. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976.
22. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
23. 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), Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
24. Bouafia, Y.; Guezouli, L. An Overview of Deep Learning-Based Object Detection Methods. In Proceedings of the International Conference on Artificial Intelligence and Information Technology (ICA2IT19), Yogyakarta, Indonesia, 13–15 March 2019.
25. Kang, J.; Tariq, S.; Oh, H.; Woo, S.S. A Survey of Deep Learning-based Object Detection Methods and Datasets for Overhead Imagery. *IEEE Access* **2022**, *10*, 20118–20134. [CrossRef]
26. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. *arXiv* **2017**, arXiv:1703.06870.
27. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
28. Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P.H.S.; Hospedales, T.M. Learning to Compare: Relation Network for Few-Shot Learning. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2017; pp. 1199–1208.
29. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. *arXiv* **2020**, arXiv:2005.12872.
30. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–27 October 2017; pp. 764–773.

31. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2020**, arXiv:2010.04159.
32. Haase, D.; Amthor, M. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14588–14597.
33. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. In Proceedings of the British Machine Vision Conference, Bhubaneswar, India, 7–10 September 2020.
34. Meethal, A.; Granger, E.; Pedersoli, M. Cascaded Zoom-in Detector for High Resolution Aerial Images. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 17–24 June 2023; pp. 2046–2055.
35. Zhou, Z.; Yu, X.; Chen, X. Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer. *Drones* **2023**, *7*, 466. [CrossRef]
36. Yang, Z.; Feng, H.; Ruan, Y.; Weng, X. Tea Tree Pest Detection Algorithm Based on Improved Yolov7-Tiny. *Agriculture* **2023**, *13*, 1031. [CrossRef]
37. Wen, C.; Guo, H.; Li, J.; Hou, B.; Huang, Y.; Li, K.; Nong, H.; Long, X.; Lu, Y. Application of improved YOLOv7-based sugarcane stem node recognition algorithm in complex environments. *Front. Plant Sci.* **2023**, *14*, 1230517. [CrossRef]
38. Cao, F.; Ma, S. Enhanced Campus Security Target Detection Using a Refined YOLOv7 Approach. *Trait. Signal* **2023**, *40*, 2267–2273. [CrossRef]
39. Zeng, Y.; Zhang, T.; He, W.; Zhang, Z. YOLOv7-UAV: An Unmanned Aerial Vehicle Image Object Detection Algorithm Based on Improved YOLOv7. *Electronics* **2023**, *12*, 3141. [CrossRef]
40. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
41. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* **2019**, *109*, 43–76. [CrossRef]
42. Tajbakhsh, N.; Shin, J.Y.; Gurudu, S.R.; Hurst, R.T.; Kendall, C.B.; Gotway, M.B.; Liang, J. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transact. Med. Imaging* **2016**, *35*, 1299–1312. [CrossRef] [PubMed]
43. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, 11–13 April 2011.
44. Zhu, P.F.; Wen, L.; Du, D.; Bian, X.; Ling, H.; Hu, Q.; Nie, Q.; Cheng, H.; Liu, C.; Liu, X.; et al. VisDrone-DET2019: The Vision Meets Drone Object Detection in Image 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. 213–226.
45. Henderson, P.; Ferrari, V. End-to-End Training of Object Class Detectors for Mean Average Precision. *arXiv* **2016**, arXiv:1607.03476.
46. Mohammed, S.A.K.; Razak, M.Z.A.; Rahman, A.H.A. An Efficient Intersection Over Union Loss Function for 3D Object Detection. In Proceedings of the 2022 Iraqi International Conference on Communication and Information Technologies (IICCIT), Basrah, Iraq, 7–8 September 2022; pp. 38–43.
47. Müller, R.; Kornblith, S.; Hinton, G.E. When Does Label Smoothing Help? *arXiv* **2019**, arXiv:1906.02629.
48. Goyal, P.; Dollár, P.; Girshick, R.B.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv* **2017**, arXiv:1706.02677.
49. Dadboud, F.; Patel, V.; Mehta, V.; Bolic, M.; Mantegh, I. Single-Stage UAV Detection and Classification with YOLOV5: Mosaic Data Augmentation and PANet. In Proceedings of the 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Washington, DC, USA, 16–19 November 2021; pp. 1–8.
50. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
51. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021.
52. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.