applied sciences

MDPI

Article

An Improved YOLO Model for UAV Fuzzy Small Target Image Detection

Yanlong Chang [1], Dong Li [2,*], Yunlong Gao [1], Yun Su [2] and Xiaoqiang Jia [2]

[1] Department of Electronic Information Engineering, School of Information Engineering, Inner Mongolia University of Technology, Hohhot 010000, China
[2] School of Information Engineering, Inner Mongolia University of Technology, Hohhot 010000, China
* Correspondence: lidong@imut.edu.cn; Tel.: +86-1594-941-1984

Abstract: High-altitude UAV photography presents several challenges, including blurry images, low image resolution, and small targets, which can cause low detection performance of existing object detection algorithms. Therefore, this study proposes an improved small-object detection algorithm based on the YOLOv5s computer vision model. First, the original convolution in the network framework was replaced with the SPD-Convolution module to eliminate the impact of pooling operations on feature information and to enhance the model's capability to extract features from low-resolution and small targets. Second, a coordinate attention mechanism was added after the convolution operation to improve model detection accuracy with small targets under image blurring. Third, the nearest-neighbor interpolation in the original network upsampling was replaced with transposed convolution to increase the receptive field range of the neck and reduce detail loss. Finally, the CIoU loss function was replaced with the Alpha-IoU loss function to solve the problem of the slow convergence of gradients during training on small target images. Using the images of *Artemisia salina*, taken in Hunshandake sandy land in China, as a dataset, the experimental results demonstrated that the proposed algorithm provides significantly improved results (average precision = 80.17%, accuracy = 73.45% and recall rate = 76.97%, i.e., improvements by 14.96%, 6.24%, and 7.21%, respectively, compared with the original model) and also outperforms other detection algorithms. The detection of small objects and blurry images has been significantly improved.

Keywords: UAV photography; small object detection algorithm; YOLOv5s; SPD-Convolution module; coordinate attention mechanism

Citation: Chang, Y.; Li, D.; Gao, Y.; Su, Y.; Jia, X. An Improved YOLO Model for UAV Fuzzy Small Target Image Detection. *Appl. Sci.* **2023**, *13*, 5409. https://doi.org/10.3390/app13095409

Academic Editor: Andrea Prati

Received: 31 March 2023
Revised: 20 April 2023
Accepted: 24 April 2023
Published: 26 April 2023

1. Introduction

Object detection algorithms have rapidly developed with the advancement of image processing technology and artificial intelligence in recent years. With their small weight and rapid features, drones have been applied widely in fields such as agriculture, medicine, and city inspections. However, the various objects of interest in unmanned aerial vehicle (UAV) images, such as pedestrians and clusters of flowers, are generally small in scale owing to the high altitude from which these images are captured. Moreover, these small targets are straightforwardly affected by environmental interference, which hinders detection using conventional object-detection algorithms. Therefore, enhancing the capability to detect small targets in UAV [1–3] aerial images has become a challenging research direction in the field of object detection.

Currently, object detection algorithms are being improved and refined continuously [4]. Xie et al. [5] proposed an improved algorithm (called Drone-YOLO) based on YOLOv5 for small object detection in UAV images. They added a detection branch and designed a feature pyramid network with multi-level information aggregation. In addition, they introduced a feature fusion module to decouple the classification and regression tasks in the prediction head and used the Alpha-IoU optimized loss function to improve the detection

performance of the model. The proposed algorithm outperformed other mainstream models in detecting small objects and could effectively complete tasks of small target detection in UAV aerial images. Zhang et al. [6] proposed an improved YOLOv5-based algorithm for vehicle and pedestrian identification, comprising the Ghost Bottleneck module [7] to compress network parameters and reduce the overall computational workload of the model. In addition, this algorithm improved the model's inference speed, resulting in significantly increased detection accuracy and speed. Li et al. [8] proposed an improved algorithm for detecting small objects in UAV images in real time to enhance the safety of autonomous landings and the capability for target identification. The algorithm added a detection head and replaced the PANet structure with BiFPN to improve the detection performance for different scales. They also replaced CIoU loss with EIoU loss as the algorithm's loss function to improve the overall performance of the model while increasing the bounding box regression rate. The improved algorithm was applied to detect QR code landing markers in UAV autonomous landing scenarios. The improved algorithm exhibited a stronger feature extraction capability and higher detection accuracy than the previous algorithm. Tian et al. [9] proposed a small target recognition algorithm with lightweight improvements to the YOLOv4 network and tested it on the VisDrone dataset. The improved algorithm was 1.5% more accurate and 3.3 times faster, making it more effective and practical. In addition, the algorithm could also judge the KCF tracking situation by analyzing the response value and realized the template update of the adaptive learning rate. Through experiments, it was proved that the algorithm could stably track long-distance small targets. Cheng et al. [10] proposed a real-time UAV target detection algorithm based on edge computing, namely Fast-YOLOv4. On the edge-computing platform NVIDIA Jetson Nano, Fast-YOLOv4 was used to intelligently analyze video to achieve rapid detection of UAV targets. At the same time, technologies such as lightweight network MobileNetV3, Multiscale-PANet, and soft merging, have been introduced to improve YOLOv4, thus obtaining the Fast-YOLOv4 model. Compared with the original model, the detection accuracy and speed of the Fast-YOLOv4 model have been significantly improved. Li et al. [11] proposed a Densely Nested Attention Network (DNA-Net) for Single Frame Infrared Small Target (SIRST) detection, using a Densely Nested Interaction Module (DNIM) to achieve gradual progression between high-level and low-level feature interactions. Based on DNIM, a Cascaded Channel and Spatial Attention Module (CSAM) was proposed to adaptively enhance multi-level features, and it achieved better performance in terms of IoU (IoU). Ibrokhimov et al. [12] proposed a new two-stage deep learning method. In the first phase, the target area is extracted, and small squares are generated to narrow down the region of interest (RoI). In the second phase, the targets are detected and classified into BI-RADS categories. To improve the accuracy of classification within a class, a classification model was designed, and its results were combined with the classification score of the detection model. This method improves the average accuracy (mAP) by 0.09, which is better than the original model. In addition, the performance of this two-stage model was verified by comparison with the existing work. Šavc et al. [13] proposed a method for the detection of skull marks, SCN-EXT convolutional neural networks, based on SpatialConfiguration-Net networks, upgraded by adding a simpler local appearance and replication of spatial configuration components. By increasing the CNN capacity without increasing the number of free parameters, this method resulted in a significant increase of about 3% using the AUDAX database. Li et al. [14] proposed a residual convolutional neural network solution for pest recognition based on transfer learning. Data enhancement was realized by random planting, color transformation, CutMix, and other operations. The classification accuracy of the ResNeXt-50 (32 $\times$ 4 d) model was compared under different combinations of learning rate, transfer learning, and data enhancement, and the impact of data enhancement on different sample classification performance was also compared. The results showed that the classification effect of the model based on transfer learning was generally better than that of the model based on new learning.

This paper proposes an algorithm for small object detection in UAV images based on an improved YOLOv5s network, inspired by the literature discussed above. The proposed algorithm incorporates several improvements, including the introduction of a coordinate attention mechanism after the backbone convolution operation to enhance the feature extraction capability; replacement of the original convolution with SPD-Convolution to improve the detection of low-resolution and small objects; utilization of transposed convolution rather than nearest-neighbor interpolation in the neck to increase the network's receptive field and prevent issues, such as decreased image resolution and loss of details; and replacement of the CIoU Loss with Alpha-IoU to accelerate the model's gradient convergence. The experimental results demonstrate that the improved algorithm has better detection capability for low-resolution and small objects compared with the original algorithm.

2. Object Detection Algorithm

2.1. Overview of Algorithm Series

The core of object detection [15,16] is to automatically locate all the objects of interest in an image and determine their categories and relative positions. There are two main types of commonly used object detection algorithms: two-stage object detection algorithms and one-stage object detection algorithms. Two-stage object detection algorithms generate candidate boxes, classify them, and finally classify the detection targets. The common two-stage object detection algorithms include R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN [17–19], and R-FCN. One-stage object detection algorithms perform candidate box generation and classification simultaneously to obtain both category and position information. These algorithms are distinguished from two-stage object detection algorithms by their higher detection speed and lower computational workload. However, their accuracy is marginally lower. Common one-stage object detection algorithms include YOLO series detection networks [20], SSD [21], and Retina-Net.

2.2. YOLOv5s

The network architecture of YOLOv5s is illustrated in the Figure 1 below. It consists of four parts: input, backbone, neck, and prediction.

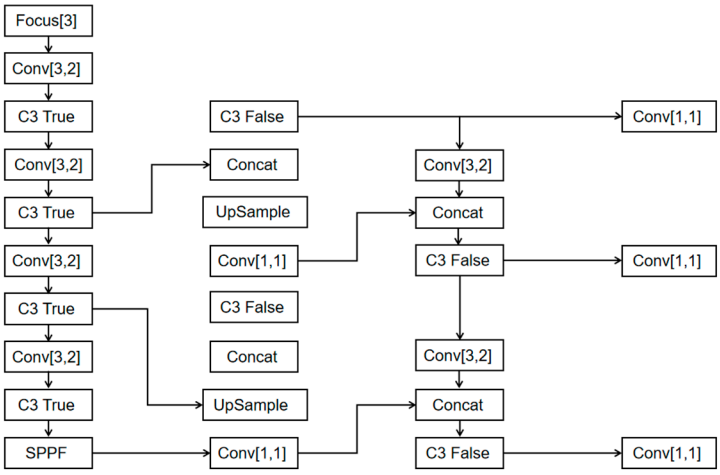


Figure 1. YOLOv5s network architecture.

Mosaic data augmentation is applied at the input end to increase the diversity of the images and improve the robustness of the network. The network utilizes adaptive anchor box calculation. Here, the network parameters are updated iteratively by automatically comparing the predicted and ground-truth bounding boxes and updating them accordingly. Furthermore, the adaptive image scaling method is utilized to expedite the inference of the network.

The slicing operation in the Focus module is a crucial part of the backbone, as shown in Figure 2. Its principle is to take an input image of size 608 × 608 × 3 and convert it into a feature map of size 304 × 304 × 32 after the slicing and convolution operations. After the Focus module, the network is connected to the CSP1_X structure. Here, X denotes the number of such modules. The CSP1_X structure consists of CBL modules, Resunit, Conv, Concat, Batch Normalization, and LeakyReLU. These components play a role in down-sampling and feature map extraction.

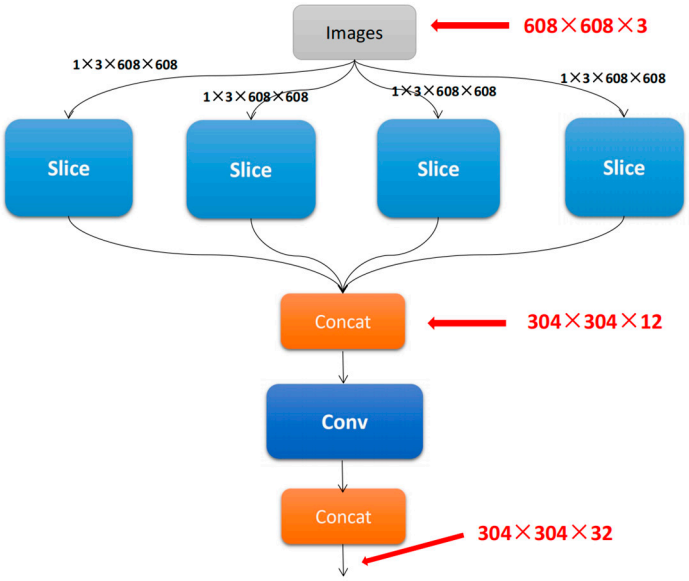


Figure 2. Slicing operation.

A structure of *FPN* + *PAN* [22,23] is used in the neck of the YOLOv5s network. Herein, the FPN layer conveys strong semantic features from the top to the bottom, and the *PAN* layer conveys strong positional features from the bottom to the top. Using this structure, the features extracted from different layers are aggregated in the detection layer, and the *CSP2_X* structure is introduced to enhance the network's feature extraction capability.

At the output end, the loss function for the bounding box is *CIOU_Loss*. Its calculation process is as follows:

$$CIoU\_Loss = 1 - CIoU = 1 - (IoU - \frac{Distance\_2^2}{Distance\ C^2} - \frac{v^2}{(1-IoU)+v}) \quad (1)$$

$$v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^p}{h^p})^2 \quad (2)$$

During the post-processing stage, the weighted *NMS* operation is used to filter a large number of bounding boxes and eliminate redundant detection boxes. This state is favorable for detecting occluded objects.

3. Improvement of YOLOv5s Algorithm

3.1. SPD-Conv Module

Convolutional neural networks play a significant role in computer vision tasks, such as image classification and object detection. However, their performance decreases rapidly when the image resolution or object size is small. A large number of strided convolutions and pooling layers are present in the feature extraction network of CNNs. These layers result in the loss of a large amount of fine-grained information and the learning of less efficient features. Therefore, in 2022, Raja Sunkara and Tie Luo from the University of Missouri proposed a novel CNN module called SPD-Conv [24]. It replaced each strided

convolution and each pooling layer with this module. SPD-Conv consists of an SPD layer and a non-strided convolution layer.

In the SPD layer, the feature map X is downsampled, while the information in the channel dimension is preserved, preventing loss of information.

For any intermediate feature map X with a size of S × S × C1, a series of sub-feature maps are cut out, as shown in Formula (3).

$$\begin{gathered} f_{0,0} = X[0:S:scale, 0:S:scale], f_{1,0} = X[1:S:scale, 0:S:scale], \ldots, \\ f_{scale-1,0} = X[scale-1:S:scale, 0:S:scale]; \\ f_{0,1} = X[0:S:scale, 1:S:scale], f_{1,1}, \ldots, \\ f_{scale-1,1} = X[scale-1:S:scale, 1:S:scale]; \\ \vdots \\ f_{0,scale-1} = X[0:S:scale, scale-1:S:scale], f_{1,scale-1}, \ldots, \\ f_{scale-1,scale-1} = X[scale-1:S:scale, scale-1:S:scale]. \end{gathered} \tag{3}$$

For any feature map X, the sub-feature map $f_{x,y}$ is formed by all the entries X(i + y) for i and y that are divisible by a proportion of i + x and i + y. Therefore, each sub-feature map is downsampled from X with the scale factor of the proportion.

The following Figure 3 shows an example of scale = 2. It results in four sub-feature maps, namely $f_{0,0}$, $f_{1,0}$, $f_{0,1}$, and $f_{1,1}$. Each of these maps has a shape of (S/2, S/2, C_1) and downsamples X by a factor of two.

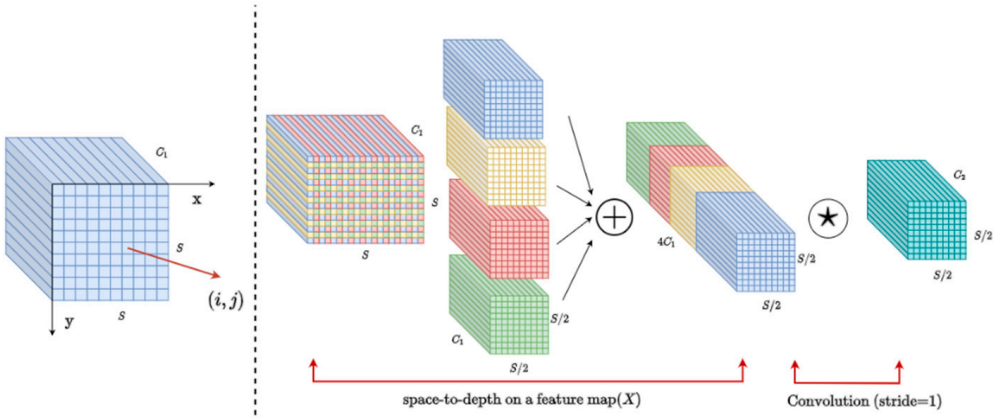


Figure 3. Downsampling operation.

Subsequently, these sub-feature maps are concatenated along the channel dimension to obtain a new feature map X′. The spatial dimensions are reduced by a scaling factor, and the channel dimensions are increased by a factor of two.

The additional convolution added after each SPD layer is a non-strided operation that can learn to decrease or increase the number of channels in the parameters. Specifically, after the SPD feature transformation layer, a non-strided convolutional layer with C2 filters (i.e., stride = 1) is added. Here, $C_2 < \text{scale}^2 C_1$, which is further transformed into X″ (S/scale, S/scale, C_2).

By incorporating this module in YOLOv5s, replacing the stride-2 convolutions in YOLOv5s with SPD-Conv, and replacing five stride-2 convolution layers in the backbone, the feature map is downsampled by a factor of 25. In addition, two stride-2 convolution layers are replaced in the neck, effectively improving the model's capability to identify small objects and low-resolution images while reducing the loss of fine-grained information and the learning of less efficient features during feature extraction. The network architecture of YOLOv5s with the addition of the *SPD-Conv* convolution module is presented in Figure 4.

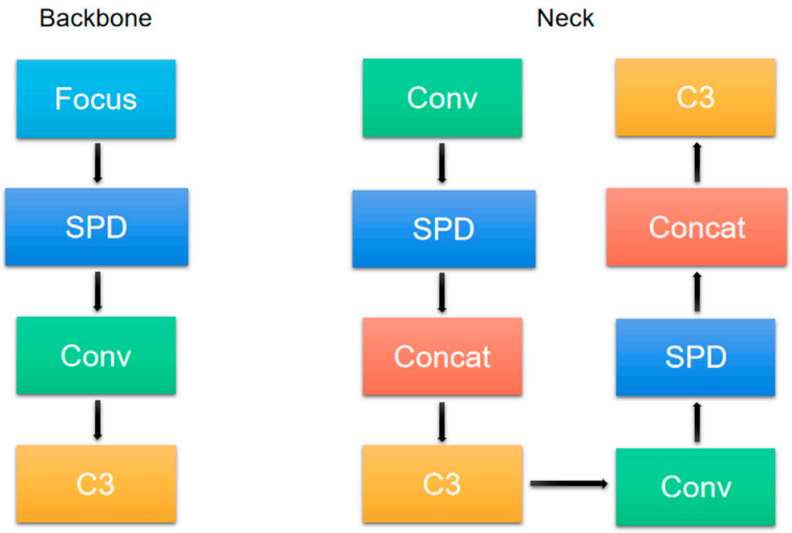


Figure 4. Network architecture of YOLOv5s model after the addition of SPD-Conv convolution module.

3.2. Attention Mechanism

The attention mechanism has a significant effect on the improvement of the model performance. In addition, positional information plays a crucial role in the generation of attention maps. However, positional information is generally omitted in the construction of attention mechanisms. In SE attention, the focus is on developing interdependence between channels while omitting spatial features. CBAM introduces large-scale convolution kernels to extract spatial features. However, it omits long-range dependence issues. The coordinate attention (CA) [25,26] published in CVPR in 2021 differs from the previous single feature vector channel attention. The CA module aggregates features through two spatial directions, thereby generating a direction-aware and position-sensitive attention map. This process enhances the feature aggregation for the target of interest. The CA module architecture is presented in Figure 5.

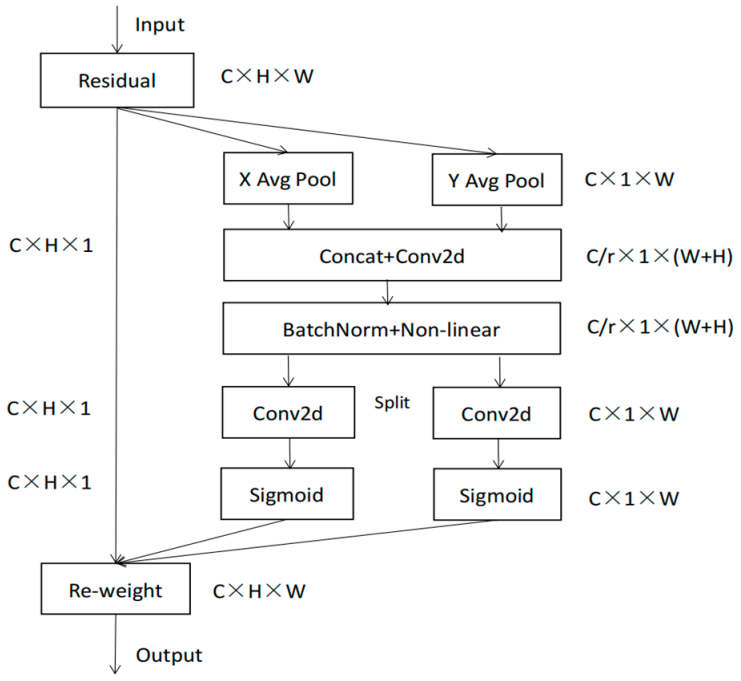


Figure 5. Architecture of CA module.

The algorithmic process of the CA module [27] is as follows:

Step 1: To prevent the compression of all the spatial information into channels, the network does not use average pooling. Global average pooling is decomposed as follows to capture detailed spatial information from distant spaces:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{4}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le i < H} x_c(j, w) \tag{5}$$

The input feature map with the size of CHW is pooled along the X- and Y-directions. This process yields two feature maps with sizes of C × H × 1 and C × 1 × W.

Step 2: The generated C × 1 × W feature map is transformed and then concatenated.

$$f = \delta(F_1([z^h, z^w])) \tag{6}$$

Concatenation is performed between the transformed z^h and z^w feature maps. This concatenation generates a new feature map. A × 1 × 1 convolution is applied to the concatenated feature map for dimension reduction and activation, yielding a new feature map $f \in R^{C/r \times (H+W) \times 1}$.

Step 3: Along the spatial dimension, the feature map f is split into $f^h \in R^{C/r \times H \times 1}$ and $f^w \in R^{C/r \times 1 \times W}$. Then, a × 1 × 1 convolution is used for dimensionality expansion, and the activation function sigmoid is applied to obtain the final spatial vectors $g^h \in R^{C \times H \times 1}$ and $g^w \in R^{C \times 1 \times W}$.

Here,

$$\begin{aligned} g^h &= \sigma(F_h(f^h)), \\ g^w &= \sigma(F_w(f^w)). \end{aligned} \tag{7}$$

The final output formula of CA can be presented as

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j) \tag{8}$$

After adding CA to the C3 [28] module in the backbone of YOLOv5s, cross-channel information, as well as direction-aware and position-sensitive information, is captured to help the model to accurately locate and identify objects in the training set. By emphasizing this information to enhance features, the network's ability to extract feature information in the backbone is strengthened, enabling the network to focus more on the features of small targets and to reduce the influence of other information. This process, in turn, improves the model's detection accuracy for small objects in blurred images. The backbone architecture after the CA module is added to the YOLOv5s model is presented in Figure 6.

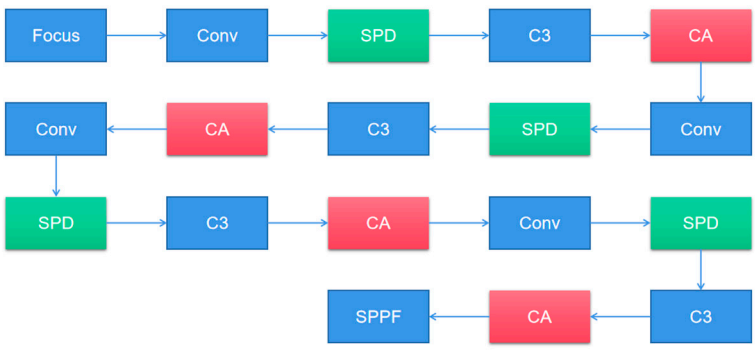


Figure 6. YOLOv5s model after CA module network architecture added.

3.3. Transposed Convolution

Transposed convolution is also a type of convolution operation. A large majority of transpose convolutions are used to achieve upsampling. Transposed convolution is an operation that maintains the correspondence between matrix elements, rather than being the inverse operation of convolution. Because it is upsampling, it is generally a one-to-many relationship. This fact implies that an element of the input matrix corresponds to multiple elements at the corresponding position in the output matrix. Transposed convolution can automatically learn parameters, enabling the network to learn the optimal upsampling method through training. In terms of form, transposed convolution is equivalent to backpropagation of the gradient computation of a convolutional layer. Here, unlike normal convolution, the weight matrix is first transposed and then left-multiplied with the input. The convolution process is illustrated in Figure 7.

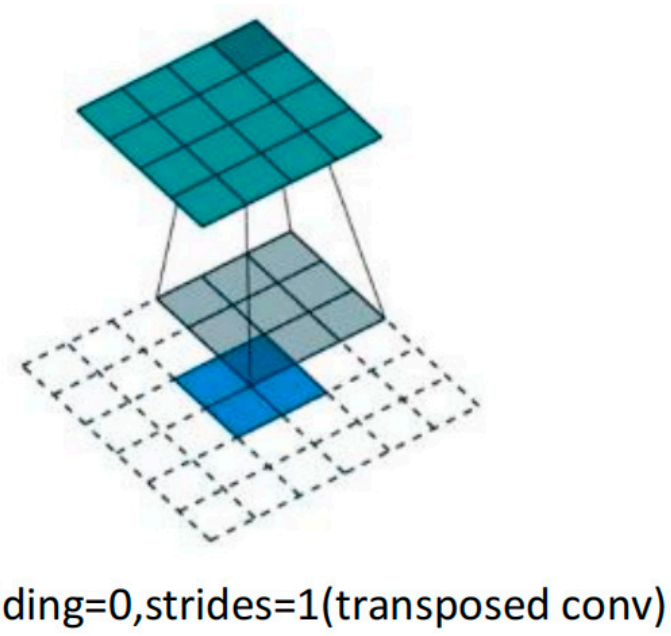


Figure 7. Convolution process and transposed convolution process.

Assuming that the convolution kernel is C and that the input is a column vector n, the convolution and transposed convolution processes can be represented as follows:

$$\begin{array}{c} CX = Y \\ X = C^T Y \end{array} \tag{9}$$

In transposed convolution, although the receptive field of the points on the convolutional kernel remains the same, the convolution layer added after transposed convolution would expand the receptive field range in the network.

In the YOLOv5s architecture, we replace the nearest-neighbor interpolation in the original network with transposed convolution to expand the receptive field [29] range in the neck structure. Doing so enables the network to capture more global information during the upsampling process, thereby reducing the loss of information and preventing problems such as low image resolution and detail loss at the end of the neck. The network architecture after the addition of the transposed convolution module to the YOLOv5s model is shown in Figure 8.

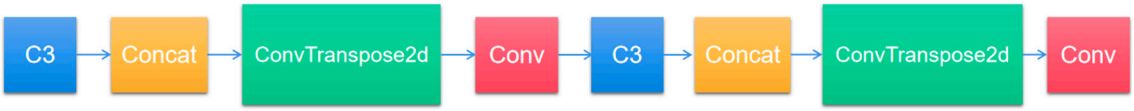


Figure 8. Network architecture of YOLOv5s after the addition of transposed convolution module.

3.4. Improved Loss Function

The loss function of the YOLOv5s network model is mainly composed of three parts. Its calculation method is as follows:

$$\psi = \psi_{obj} + \psi_{bbox} + \psi_{cls} \tag{10}$$

where ψ_{obj} is the loss of object confidence, which is calculated using the binary cross-entropy loss. ψ_{bbox} is the classification loss of the object, which is calculated using the cross-entropy loss. ψ_{cls} denotes the loss for the predicted bounding box positions. This study uses Alpha-IoU to optimize the calculation of box loss. The loss function for the predicted bounding box positions is calculated as shown in the formula.

$$\mathcal{L}\alpha - CIoU = 1 - IoU^{\alpha} + \frac{\rho^{2\alpha}\left(b, b^{gt}\right)}{c^{2\alpha}} + (\beta v)^{\alpha} \tag{11}$$

here, IoU represents the intersection over the union between the predicted bounding boxes (b) and ground-truth bounding boxes (b^{gt}). Alpha (α) is a hyperparameter that can be adjusted to control the accuracy of the predicted bounding boxes. ρ represents the Euclidean distance between the centers of the computed predicted bounding boxes and ground-truth bounding boxes. C represents the diagonal length of the minimum bounding box, which encloses the predicted bounding box and ground-truth bounding box. v is a parameter that measures the similarity of the aspect ratio between the predicted bounding box and ground-truth bounding box. Its calculation formula is shown in Equation (12).

$$v = \frac{4}{\pi}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right) \tag{12}$$

where (w^{gt},h^{gt}) and (w,h) are the width and height, respectively, of the ground-truth bounding box and predicted bounding box.

$$\beta = \frac{v}{(1 - IoU) + v} \tag{13}$$

The original YOLOv5 model used the *CIoU loss*. It is calculated using the following formula:

$$L_{CIoU} = 1 - I + \frac{\rho^{2}\left(b, b^{gt}\right)}{c^{2}} + \theta v \tag{14}$$

The advantage of the CIoU Loss is that it considers the intersection over union (IoU), center point distance, and aspect ratio between the predicted bounding box and ground-truth bounding box. However, the disadvantage is that the parameter v in the formula does not consider the differences in width and height relative to their confidence. This fact can reduce the model optimization speed and decrease the convergence rate of the gradient when training on a large number of images with small objects. Using the Alpha-IoU loss function and increasing the exponent of alpha, the gradient convergence speed is increased to address the limitations of the CIoU loss, improving the model's detection capability with small objects.

The network architecture of the improved YOLOv5s after the four aforementioned improvements is presented in Figure 9.

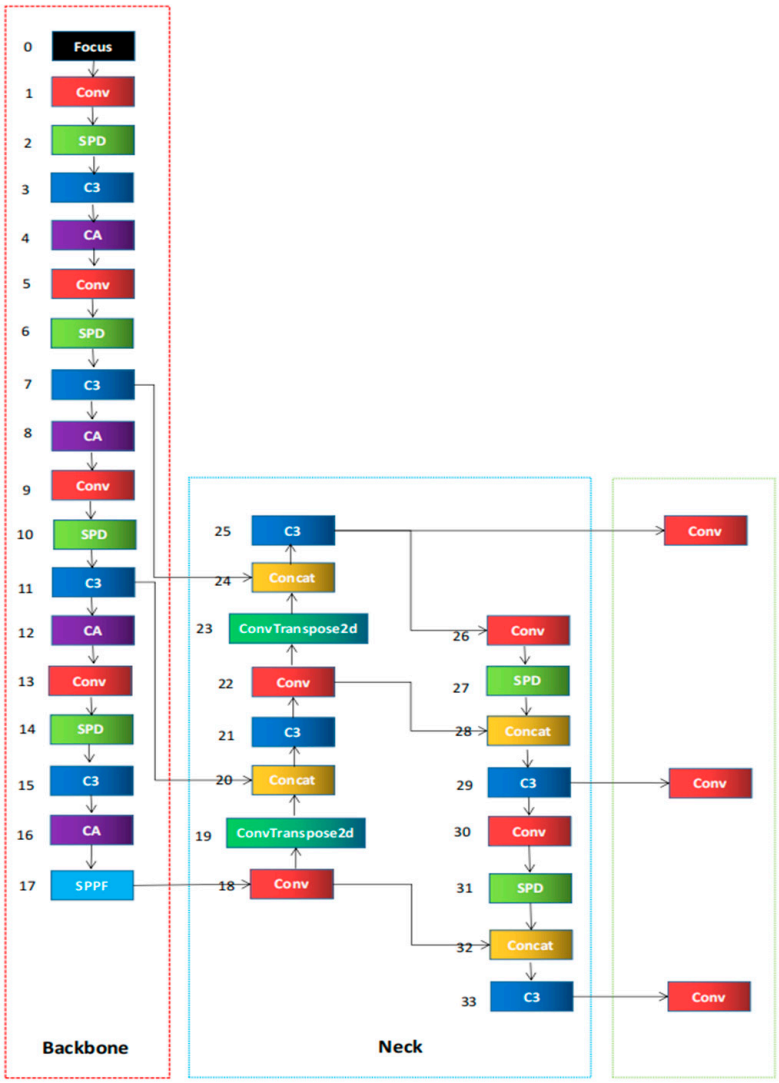


Figure 9. Network architecture of the improved YOLOv5s.

4. Experiment and Discussion

4.1. Experimental Environment

In the experiment, the specific equipment of the experiment is as follows. Experimental equipment parameters are shown in Table 1:

Table 1. Experimental equipment parameters.

Software and Hardware	Version or Model
Operating system	Windows11
CPU	i7-13700K
GPU	NVIDIA GeForce RTX 3080Ti
CUDA	12.0
Pytorch version	11.8
Python version	3.9
Software	PyCharm2021

4.2. Model Parameters

Training parameter settings are shown in Table 2:

Table 2. Training parameter settings.

Parameter Name	Parameter Settings
Weights	Yolov5s.pt
Img-size	640 × 640
Epochs	300
Batch-size	16
Max-det	1000

Training hyperparameter settings are shown in Table 3:

Table 3. Training hyperparameter settings.

Parameter Name	Parameter Settings	Parameter Explanation
Lr0	0.01	Initial learning rate
Lrf	0.1	Cyclic learning rate
Momentum	0.937	Learning rate momentum
Weight_decay	0.0005	Weight decay factor

The rest of the parameters are set as default parameters.

4.3. Dataset

In the experiment, a dataset of *Artemisia artemisia* images taken by a self-made drone and a small target dataset of human gestures are used. The *Artemisia salina* image dataset, a total of 2000 images of *Artemisia salina*, were obtained at a high altitude in the Hunshandake sandy land in the Inner Mongolia Autonomous Region of China using a DJI M300pro. The *Artemisia salina* image datasets mainly have the characteristics of low resolution and small detection targets. The small target dataset of human body postures was captured using a DJI M300pro at Inner Mongolia University of Technology. After collecting the dataset, we used labelImg software to complete the labeling of images. Finally, the dataset was divided into training and test sets according to the ratio of 9:1 for network training.

4.4. Evaluation Metrics

In this study, we evaluated the segmentation accuracy of the model using three performance metrics: mean average precision (mAP), precision, and recall. We used Parameters, FLOPs, and Latency to evaluate the complexity of the model. The formulas for these metrics are as follows:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{15}$$

$$AP = \int_0^1 PdR \tag{16}$$

$$\frac{TP}{TP + FP} \tag{17}$$

$$r = \frac{T_P}{T_P + F_n} \tag{18}$$

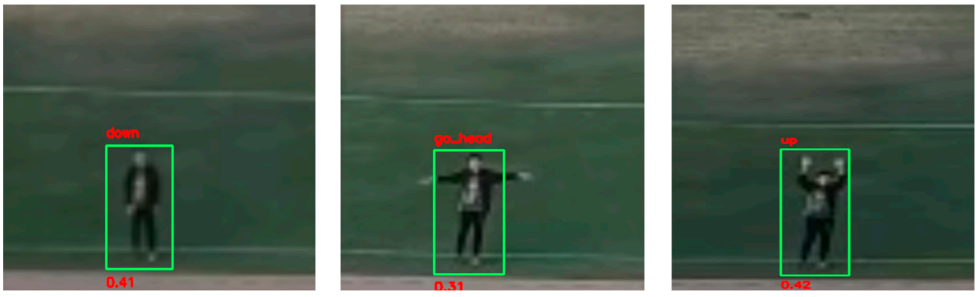
here, TP refers to the true positive samples predicted as positive by the model, F_n refers to the true positive samples predicted as negative by the model, FP refers to the false positive samples predicted as positive by the model, and TN refers to the true negative samples predicted as negative by the model. AP refers to the average value of the predicted results for different categories (i.e., the area under the P-R curve). mAP is the average AP value across all the categories. Precision is the proportion of correctly predicted results. Recall refers to the ratio of the number of identified categories to the number of categories in the test set. Parameters refers to the number of parameters contained in the model, which is one of the indicators of model complexity. FLOPs refer to the number of floating-point

operations in the model, which is a measure of the complexity of the model algorithm. Latency refers to the running time of the model when using the network for prediction, which only includes the time used in the forward propagation process of the network, and it is one of the indicators of model complexity.

4.5. Experiment

The training experiment is performed on the small target dataset of human body postures, and the original YOLOv5s training results are compared with the improved YOLOv5s training results:

The test results Before improvement are shown in Figure 10:



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

Figure 10. Test results before improvement. (**a**–**c**) are the three test actions before model improvement.

The improved model results are shown in Figure 11:

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

Figure 11. Improved detection results. (**a**–**c**) are the three test actions after the model improvement.

The performance comparison results of the models are shown in Figures 12–15:

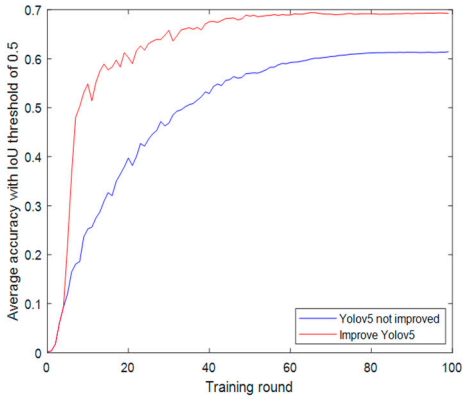


Figure 12. mAP comparison results.

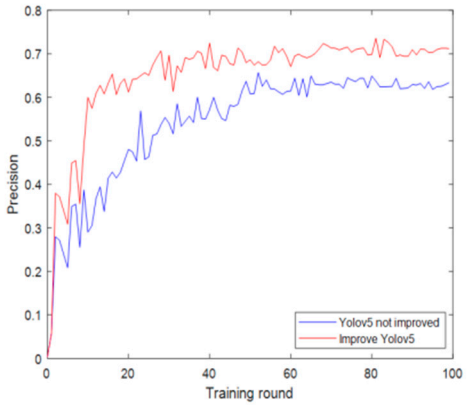


Figure 13. Precision comparison results.

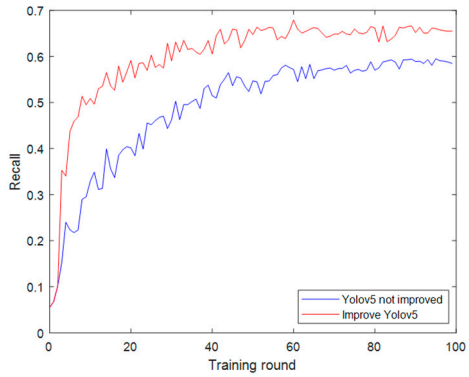


Figure 14. Recall comparison results.

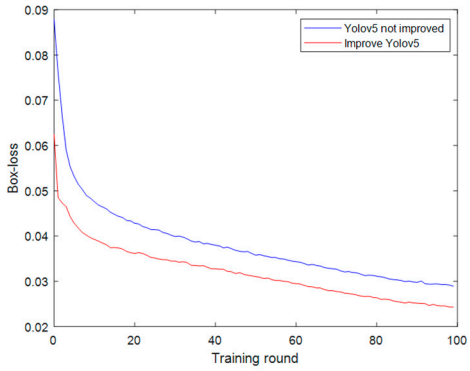


Figure 15. Loss comparison results.

It can be seen from the above comparison that the improved YOLOv5s has improved mAP, recall, Precision, and Loss index performance compared with the original YOLOv5s, and it has significantly improved detection ability for smaller targets in blurred images.

4.6. Ablation Experiment

We used our in-house UAV and Artemisia dataset for subsequent experiments to emphasize the detection of small objects and low resolution in experiments. A set of ablation experiments were designed to test the performance of different modules on the network to verify the effectiveness of the proposed improvements for small-object and low-resolution detection. The modules included the SPD-Conv module, attention mechanism, transpose convolution module, and alteration of the loss function.

Detection performance comparison:

- Detection results of Original YOLOv5s are shown in Figure 16:

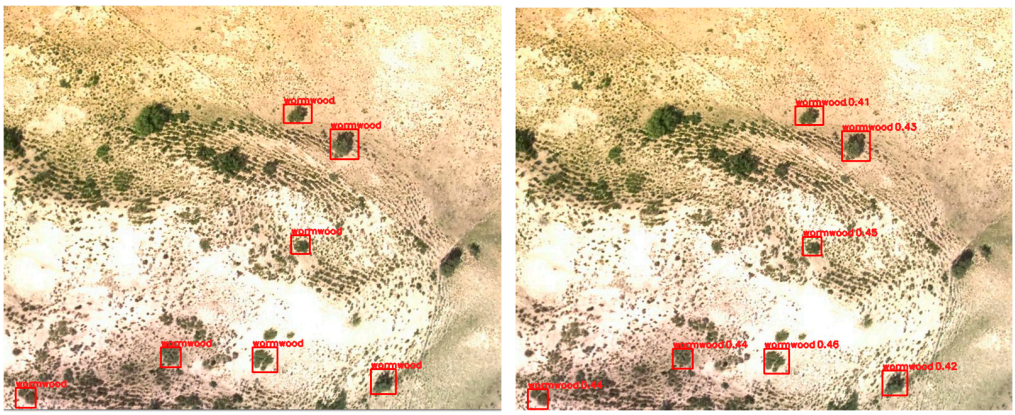


Figure 16. Detection results of Original YOLOv5s.

- Detection results with SPD-Conv module are shown in Figure 17:

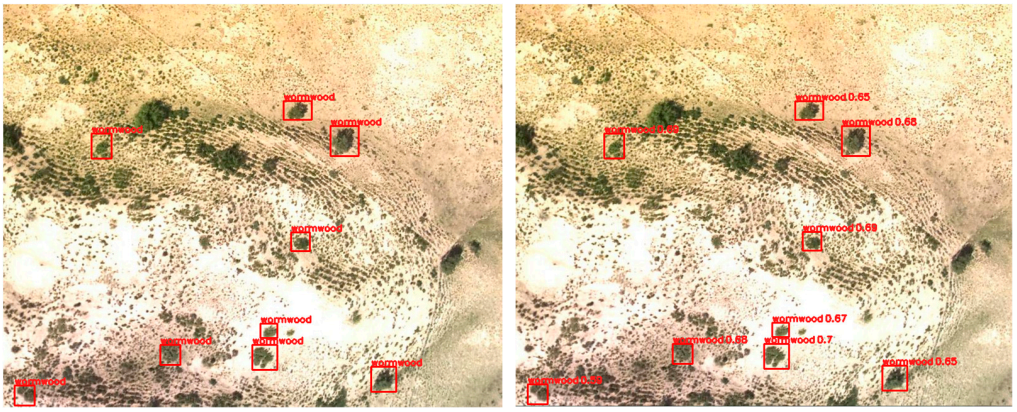


Figure 17. Detection results with SPD-Conv module.

- Detection results with CA attention mechanism are shown in Figure 18:

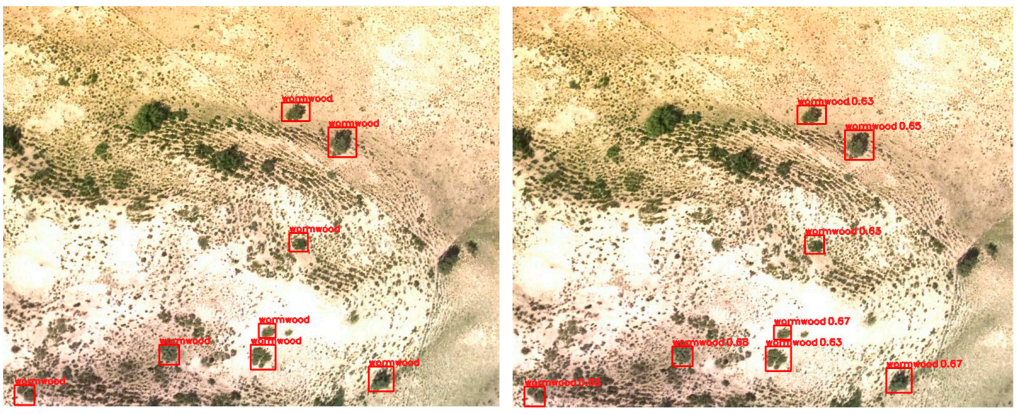


Figure 18. Detection results with CA attention mechanism.

- Detection results with transposed convolution are shown in Figure 19:

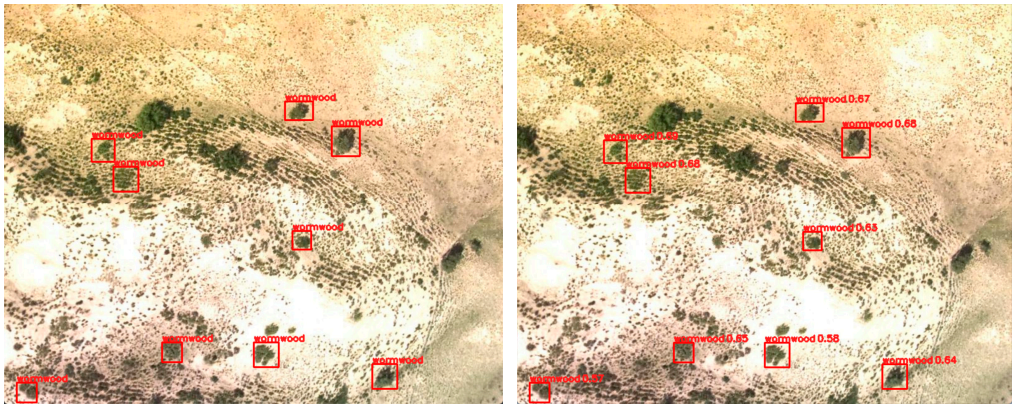


Figure 19. Detection results with transposed convolution.

- Detection results with all modules added are shown in Figure 20:

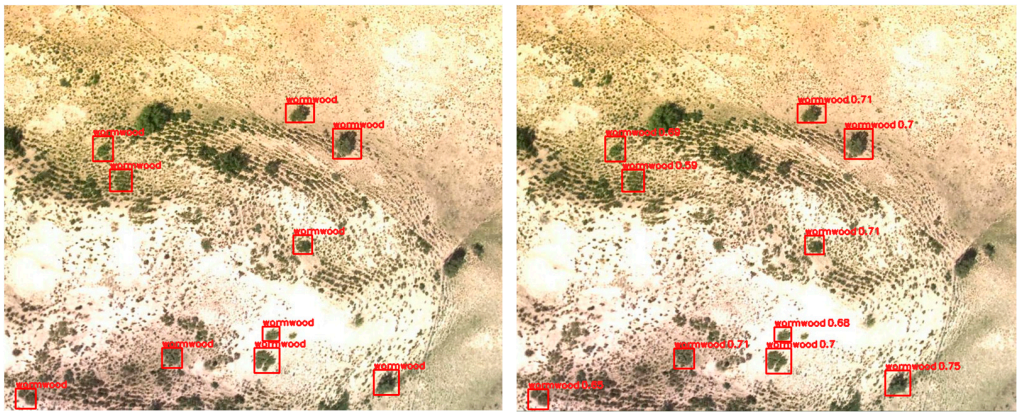


Figure 20. Detection results with all modules added.

The results shown in the graph indicate that the detection capabilities of the improved module are stronger than those of the original YOLOv5s for low resolution and small objects with an identical validation image. As observed in the graph, the detection confidence of the original YOLOv5s for small objects visible to the naked eye was 0.45. However, all the values of this parameter for the improved module were greater than 0.7. In addition, the improved model exhibited better detection performance for small objects at the edges of the image, whereas the original model had a higher rate of missed detection. To summarize, the improved model proposed in this study outperformed the original YOLOv5s model in terms of detection capabilities for low-resolution and small objects, along with a reduction in missed detections.

The specific performance of the model obtained by combining various improvement modules is presented in Tables 4 and 5.

As shown in the table, adding the SPD-Conv, CA mechanism, and transpose convolution modules individually resulted in nonsignificant improvements in the model's identification capabilities. The increase in mAP was large when only the SPD-Conv was used with the CA mechanism. However, the improvement in the model's identification accuracy was nonsignificant. Using only SPD-Conv with transpose convolution resulted in decreased model recall rate and precision, although a certain improvement in mAP was achieved. The model's recall rate and precision improved when the CA mechanism was used with transpose convolution. However, mAP decreased marginally. The model's recall rate, precision, and mAP improved to varying degrees after the three types of improvements and the loss improvement were added. Compared with the original YOLOv5s, the number of network training layers increased by 47 layers, the number of model parameters

increased by 2.89 M, the number of model floating-point operations increased by 24.2 G, and the inference time increased by 4.1 ms. Compared with the original network, the amount and reasoning time increased. Although the real-time performance of model detection decrease, the decrease has not been large. Moreover, the improved model significantly improved the accuracy of small-object detection in UAV fuzzy images. The improved model mAP, precision, and recall were increased by 14.96%, 6.24%, and 7.21%, respectively. Training results of different combinations of mAP, precision, recall and loss are shown in Figures 21–24:

Table 4. Specific performance of the model for different improvement modules.

Algorithm	SPD-Conv	CA	T-Conv	mAP/%	P/%	R/%
Yolov5s				65.21	67.24	69.76
SPD-Conv	√			75.34	69.42	74.54
CA		√		74.09	68.77	74.58
T-Conv			√	74.10	68.79	72.70
SPD-Conv + CA	√	√		78.53	71.52	75.23
SPD-Conv + T-Conv	√		√	77.12	71.36	73.62
CA + T-Conv		√	√	76.28	72.36	73.71
Ours	√	√	√	80.17	73.45	76.97

P: precision; R: recall; T-Conv refers to the transpose convolution module.

Table 5. Model complexity ratio.

Algorithm	Layers	Parameters/M	FLOPs/G	Latency/ms
Original Yolov5s	270	7.02	15.9	20.25
SPD-Conv	277	8.56	33.3	23.25
CA	280	7.05	16	21.23
T-Conv	283	8.57	23.6	22.58
SPD-Conv + CA	317	8.6	33.4	23.89
SPD-Conv + T-Conv	317	9.06	36.2	24.1
CA + T-Conv	317	9.36	38.3	22.98
Ours	317	9.91	40.1	24.35

Plots of the training results of different combinations of mAP, precision, recall, and loss reveal that the improved model in this study exhibited improvements in these parameters. It is also evident that this model performed better than the original model on low-resolution and small-object datasets.

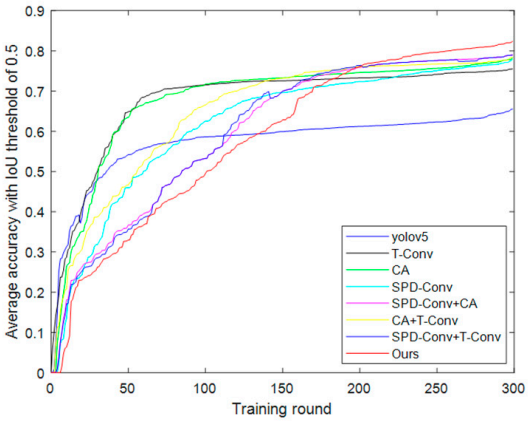


Figure 21. mAP for different modules.

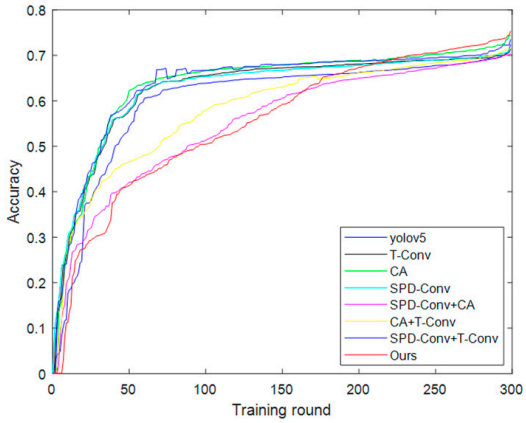


Figure 22. Precision for different modules.

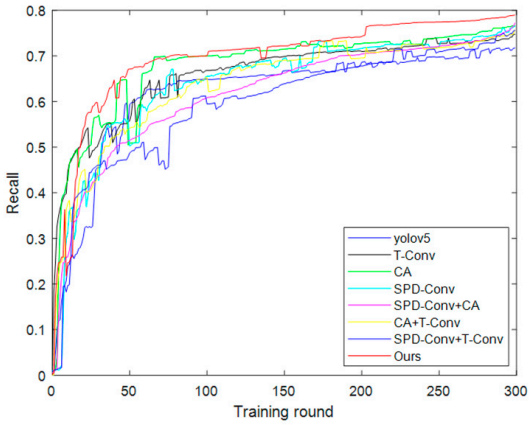


Figure 23. Recall for different modules.

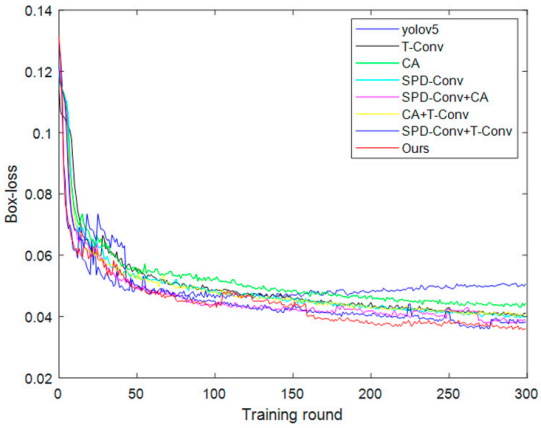


Figure 24. Loss for different modules.

4.7. Comparison of Algorithm Improvements

A model was trained on an in-house UAV dataset of weeds by conventional object-detection algorithms. The mAP, precision, and recall performance of the model are shown in Table 6.

It is worth noting that the improved model is 4.1 ms slower than the original YOLOv5s model inference time, and the number of parameters is 2.89 M larger, but the mAP, precision, and recall rate have increased by 14.96%, 6.24%, and 7.21%, respectively; TPH-YOLOv5 is mainly based on Transformer. The improved version of YOLOv5 with the prediction head can be used for target detection in scenes captured by drones. Compared with TPH-

YOLOv5, our improved target detection algorithm is 3.19 ms slower in inference time and 1.55 M larger in parameters, but the mAP, precision, and recall rate increased by 1.11%, 0.89%, and 1.85%, respectively; the improved YOLOv4 is a small-target recognition and tracking algorithm based on the UAV platform. For target detection, our improved target detection algorithm is 7.2 ms slower than the improved YOLOv4 inference time and has a larger parameter volume of 3.59M, but the mAP, precision, and recall rate increased by 11.81%, 4.97%, and 9.85%, respectively; Fast-YOLOv4 is a real-time UAV target-detection algorithm based on edge computing. It mainly adds the lightweight modules MobileNetV3, Multiscale-PANet, and soft-merge to YOLOv4 to improve the YOLOv4 model, simplify the network structure, and improve the detection speed. After our improvements, compared with Fast-YOLOv4's target detection algorithm, the inference time is 4.77 ms slower, and the number of parameters is 1.13 M larger, but the mAP, precision, and recall rate increased by 9.92%, 2.09%, and 7.19%, respectively; however, when our improved model is compared with other classic algorithms and the latest algorithm for UAV small targets, there is no advantage in the delay and model parameters, but the mAP, precision, and recall rate of the improved model are improved to varying degrees compared with other algorithms. The improved model in practical applications will bring higher detection accuracy and lower false detection rates, thereby improving the reliability and practicability of UAVs in various application scenarios.

Table 6. mAP, precision, and recall values obtained by training conventional object detection algorithms.

Algorithm	mAP/%	P/%	R/%	Latency/ms	Parameters/M
Yolov5s	65.21	67.24	69.76	20.25	7.02
Yolov4	67.32	68.36	71.45	22.72	8.39
SSD	81.23	74.21	74.49	25.16	8.63
Faster-RCNN	79.85	77.49	76.49	30.25	13.71
RetinaNet [30]	67.36	69.45	69.32	30.12	6.45
TPH-YOLOv5 [31]	79.06	72.56	75.12	21.16	8.36
Fast-YOLOv4 [10]	70.25	71.36	69.78	19.58	8.78
Improved YOLOv4 [9]	68.36	68.51	67.12	17.15	6.32
ours	80.17	73.45	76.97	24.35	9.91

5. Conclusions

This study proposed an improved object-detection algorithm based on YOLOv5s. The objective was to enhance the detection capability of unmanned aerial vehicles for low-resolution and small objects.

First, based on an experimental comparison and practical requirements, we selected YOLOv5s as the algorithm to be improved considering the differences in model size and performance. Then, we replaced the original convolution in the model with SPD-Conv convolution to enhance the network's feature extraction capability for low-resolution and small objects. Next, we integrated the CA mechanism into the YOLOv5s network structure to strengthen the feature extraction and improve the detection accuracy with small objects in blurry images. In addition, we replaced the nearest neighbor interpolation used for upsampling in the original network neck with transpose convolution to increase the receptive field of the neck and prevent issues such as low image resolution and loss of details at the end of the neck. Finally, we replaced the CIoU Loss with Alpha-IoU to accelerate the convergence of the model gradient.

Currently, the recognition and tracking of small targets based on the UAV platform have a broad range of applications, but the images captured by UAVs in the air usually have characteristics such as blur and small resolution, so it is of great value to design an algorithm suitable for fuzzy small-target detection.

This study comprehensively considers the characteristics of blurred images and small target scenes in UAV recognition, and it uses the single-stage YOLOv5s algorithm to improve it. For the small-target recognition and tracking application of the UAV platform, a detailed comparison was performed with the latest algorithm in the current field in the experiment. Our improved algorithm had higher detection accuracy and lower false detection rates, effectively improving the detection accuracy of the UAV platform for fuzzy small targets and the reliability of small-target detection in UAV fuzzy images. Although there is a slight gap in the real-time performance of our algorithm in the latest technology comparison of real-time UAV target detection algorithms based on edge computing, our detection accuracy is higher than that of the latest related research algorithms, which is very meaningful.

In addition, our improved method also provides a new idea and methodology for object detection problems in other related fields [11], with certain reference significance. In short, we believe that, through continuous optimization and improvement, UAV small-target detection technology will continue to develop toward higher accuracy and efficiency, bringing more possibilities and opportunities for UAV applications.

The algorithm proposed in this study improved the mAP, recall, and precision compared with the original YOLOv5s algorithm, resulting in better detection performance. However, the increased detection performance was achieved at the cost of increased model complexity and parameters, which may hinder its deployment on certain embedded devices. However, it is conjectured that the detection accuracy can be maintained while reducing the size and complexity of the network to facilitate deployment on low-computing devices.

Author Contributions: Conceptualization, Y.C. and D.L.; writing—original draft preparation, Y.C. and Y.G.; data curation, Y.C. and Y.G.; writing—review and editing, D.L., Y.S. and X.J.; project administration, D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by the Innovation and Entrepreneurship Training Program for College Students of Inner Mongolia University of Technology (Project Name: Design of Immersive UAV System Based on Gesture Control; Project Number: 2022023006) and the Natural Science Foundation of Inner Mongolia Autonomous Region (Project Number: 2022QN06004).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Liu, Y.; Yang, F.; Hu, P. Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks. *IEEE Access* **2020**, *8*, 145740–145750. [CrossRef]
2. Yu, W.; Yang, T.; Chen, C. Towards Resolving the Challenge of Long-tail Distribution in UAV Images for Object Detection. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021; IEEE Press: Waikoloa, HI, USA, 2021; pp. 3257–3266.
3. Zhang, X.; Izquierdo, E.; Chandramouli, K. Dense and small object detection in uav vision based on cascade network. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 118–126.
4. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer vision (ECCV), Munich, Germany, 8 September 2018; pp. 734–750.
5. Xie, C.; Wu, J.; Xu, H. Small object detection algorithm based on improved YOLO5 in UAV image. *Comput. Eng. Appl.* **2023**, 1–11. Available online: http://kns.cnki.net/kcms/detail/11.2127.TP.20230214.1523.050.html (accessed on 10 March 2023).
6. Zhang, Q.; Wu, Z.; Zhou, L.; Liu, X. Research on vehicle and pedestrian target detection method based on improved YOLOv5. *Chin. Test* **2023**, 1–8. Available online: http://kns.cnki.net/kcms/detail/51.1714.TB.20230228.0916.002.html (accessed on 10 March 2023).
7. Li, Y.; Wang, S.; Chen, W.; Tian, Z.; Hou, L. Improved YOLOv5 target detection algorithm based on Ghost module. *Mod. Electron. Tech.* **2023**, *46*, 29–34. [CrossRef]

8. Li, X.; Zhen, Z.; Liu, B.; Liang, Y.; Huang, Y. Object Detection Based on Improved YOLOv5s for Quadrotor UAV Auto-Landing. *Comput. Meas. Control.* **2023**, 1–10.
9. Tian, X.; Jia, Y.; Luo, X.; Yin, J. Small Target Recognition and Tracking Based on UAV Platform. *Sensors* **2022**, *22*, 6579. [CrossRef] [PubMed]
10. Cheng, Q.; Wang, H.; Zhu, B.; Shi, Y.; Xie, B. A Real-Time UAV Target Detection Algorithm Based on Edge Computing. *Drones* **2023**, *7*, 95. [CrossRef]
11. Li, B.; Xiao, C.; Wang, L.; Wang, Y.; Lin, Z.; Li, M.; An, W.; Guo, Y. Dense Nested Attention Network for Infrared Small Target Detection. *IEEE Trans. Image Process.* **2023**, *32*, 1745–1758. [CrossRef] [PubMed]
12. Ibrokhimov, B.; Kang, J.Y. Two-Stage Deep Learning Method for Breast Cancer Detection Using High-Resolution Mammogram Images. *Appl. Sci.* **2022**, *12*, 4616. [CrossRef]
13. Martin, Š.; Gašper, S.; Božidar, P. Cephalometric Landmark Detection in Lateral Skull X-ray Images by Using Improved Spatial Configuration-Net. *Appl. Sci.* **2022**, *12*, 4644. [CrossRef]
14. Li, C.; Zhen, T.; Li, Z. Image Classification of Pests with Residual Neural Network Based on Transfer Learning. *Appl. Sci.* **2022**, *12*, 4356. [CrossRef]
15. Li, C. Small target detection algorithm based on YOLOv5. *Chang. Inf. Commun.* **2021**, *34*, 30–33.
16. Tian, F.; Jia, H.-P.; Liu, F. Small Target Detection in Oilfield Operation Field Based on Improved YOLOv5. *Comput. Syst. Appl.* **2022**, *31*, 159–168. [CrossRef]
17. Braun, M.; Krebs, S.; Flohr, F.; Gavrila, D.M. EuroCity Persons:A Novel Benchmark for Person Detection in Traffic Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1844–1861. [CrossRef] [PubMed]
18. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
19. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE Press: Salt Lake City, UT, USA, 2017; Volume 39, pp. 1137–1149.
20. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
21. Li, Q.; Deng, Z.; Luo, X.; Gu, X.; Wang, S. SSD Object Detection Algorithm with Attention and Cross-Scale Fusion. *J. Front. Comput. Sci.* **2022**, *16*, 2575–2586.
22. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; Volume 106, pp. 936–944.
24. Sunkara, R.; Luo, T. No more strided convolutions or pooling: A new CNN building block for low-resolution images and small objects. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the European Conference, ECML PKDD 2022, Grenoble, France, 19–23 September 2022*; Part III; Springer Nature Switzerland: Cham, Switzerland, 2022; pp. 443–459.
25. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Effcient Mobile Network Design. *Natl. Univ. Singapore* **2021**, arXiv:2103.02907v1.
26. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3–8.
27. Dai, J.; Zhao, X.; Li, L.; Liu, W.; Chu, X. Improved Yolov5-based for Infrared Dim-small Target Detection under Complex Background. *Infrared Technol.* **2022**, *44*, 504–512. Available online: http://kns.cnki.net/kcms/detail/53.1053.TN.20220415.1612.002.html (accessed on 15 March 2023).
28. Song, Z.; Zhang, Y.; Liu, Y.; Yang, K.; Sun, M. MSFYOLO: Feature fusion-based detection for small objects. *IEEE Lat. Am. Trans.* **2022**, *20*, 823–830. [CrossRef]
29. Qu, J.; Su, C.; Zhang, Z.; Razi, A. Dilated convolution and feature fusion SSD network for small object detection in remote sensing images. *IEEE Access* **2020**, *8*, 82832–82843. [CrossRef]
30. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2980–2988. [CrossRef] [PubMed]
31. 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, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.