biomimetics

MDPI

Article

YOLO-DRS: A Bioinspired Object Detection Algorithm for Remote Sensing Images Incorporating a Multi-Scale Efficient Lightweight Attention Mechanism

Huan Liao and Wenqiu Zhu *

School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China; m21077500009@stu.hut.edu.cn

* Correspondence: zwq@hut.edu.cn

Abstract: Bioinspired object detection in remotely sensed images plays an important role in a variety of fields. Due to the small size of the target, complex background information, and multi-scale remote sensing images, the generalized YOLOv5 detection framework is unable to obtain good detection results. In order to deal with this issue, we proposed YOLO-DRS, a bioinspired object detection algorithm for remote sensing images incorporating a multi-scale efficient lightweight attention mechanism. First, we proposed LEC, a lightweight multi-scale module for efficient attention mechanisms. The fusion of multi-scale feature information allows the LEC module to completely improve the model's ability to extract multi-scale targets and recognize more targets. Then, we propose a transposed convolutional upsampling alternative to the original nearest-neighbor interpolation algorithm. Transposed convolutional upsampling has the potential to greatly reduce the loss of feature information by learning the feature information dynamically, thereby reducing problems such as missed detections and false detections of small targets by the model. Our proposed YOLO-DRS algorithm exhibits significant improvements over the original YOLOv5s. Specifically, it achieves a 2.3% increase in precision (P), a 3.2% increase in recall (R), and a 2.5% increase in mAP@0.5. Notably, the introduction of the LEC module and transposed convolutional results in a respective improvement of 2.2% and 2.1% in mAP@0.5. In addition, YOLO-DRS only increased the GFLOPs by 0.2. In comparison to the state-of-the-art algorithms, namely YOLOv8s and YOLOv7-tiny, YOLO-DRS demonstrates significant improvements in the mAP@0.5 metrics, with enhancements ranging from 1.8% to 7.3%. It is fully proved that our YOLO-DRS can reduce the missed and false detection problems of remote sensing target detection.

Keywords: bioinspired object detection; YOLOv5; multi-scale; attention mechanisms; transposed convolution

check for updates

Citation: Liao, H.; Zhu, W. YOLO-DRS: A Bioinspired Object Detection Algorithm for Remote Sensing Images Incorporating a Multi-Scale Efficient Lightweight Attention Mechanism. *Biomimetics* **2023**, *8*, 458. https://doi.org/10.3390/biomimetics8060458

Academic Editors: Haoran Wei, Fei Tao, Zhenghua Huang and Yanhua Long

Received: 4 September 2023
Revised: 20 September 2023
Accepted: 25 September 2023
Published: 1 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the rapid development of bioinspired image processing and remote sensing technology, remote sensing object detection technology has gradually become a hot spot in current research. It is widely used in the fields of national defense, rescue [1], urban construction, geologic disasters [2], and development. In remote sensing imagery, the task of target detection is to detect and identify the precise location of specific categories of targets, such as common aircraft, automobiles, oiltank, playgrounds, etc., in remotely sensed imagery. For remote sensing images, the targets in these images are usually densely distributed, have too many small-sized targets distributed at multiple scales, and can be affected by factors such as the complexity of the detection background. Initially, features were typically extracted by artificial means such as classical algorithms, such as AdaBoost [3], SVM [4], HoGDetector [5], DMP [6], etc. However, these algorithms perform poorly in complex settings, and the high algorithm complexity makes detection inefficient and time-consuming.

The convolutional neural network(CNN) based on deep learning [7] performed well in the ImageNet image classification competition in 2012, which led to the rapid development of convolutional neural networks. In target detection, the convolutional neural network is the main direction of target detection. At present, there are two kinds of target detection methods based on deep learning. One class is two-stage target detection methods based on candidate frames, such as R-CNN [8], Fast R-CNN [9], Faster R-CNN [10], and Mask R-CNN [11] algorithms, which are more complex in design, consume more resources, and have slower detection speeds and do not meet the requirements of real-time detection. Another is a single-level regression-based target detection algorithms representing the SSD [12], Retina-Net [13], CenterNet [14], and YOLO [15–18] series. Compared with the two-stage target detection algorithm, the single-stage target detection algorithm not only has high detection accuracy but also greatly improves the detection speed, so it has been more widely used.

In recent years, the YOLO series has become a representative algorithm in the field of target detection with its fast, accurate, and mature engineering capabilities. The YOLO family of algorithms has been repeatedly improved and optimized, and while it has now evolved into YOLOv8, the YOLOv5 algorithm is the most widely used and mature algorithm for both academic and industrial use.

2. Related Work

In recent years, numerous scholars have made significant advancements in deep-learning-based object detection methods. Farhan Ullah et al. [19] proposed a cyber threat detection system that combines migration learning and multi-model image characterization in a hybrid approach. Du et al. [20] introduced BV-YOLOv5S, a modification of YOLOv5S, to achieve real-time defect detection in road pits. Li et al. [21] developed a lightweight convolutional neural network called WearNet. This network is designed to enable real-time detection of scratches on sliding metal parts. Shen et al. [22] focused on enhancing cross-scale detection in road object detection tasks using the YOLOv3 model. They employed the K-means-GIoU algorithm to generate prior boxes and implement a detection branch specifically for small targets. Wang Jian et al. [23] analyzed the challenges posed by high resolution and complex backgrounds in UAV aerial images. To address these issues, they proposed MFP-YOLO, a lightweight detection algorithm based on YOLOv5. This method combines a multi-path inverse residual module and attention module. Additionally, it utilizes a parallel deconvolutional space pyramid pool to extract scale-specific information, thereby improving the detection performance of the algorithm. Furthermore, many scholars have made significant breakthroughs in the field of remote sensing image object detection. Using the YOLOv3 model, Qu et al. [24]. proposed an auxiliary network to improve the recognition of objects in remote sensing images. The CBAM module is backward-compatible to improve network performance and prevent the loss of crucial information during training. Reference [25] proposed using DenseNet [26] to enhance YOLOv3 and to improve the accuracy of remote sensing image detection by enhancing the structure in the backbone. However, DenseNet's structure is too complex and has too many parameters, leading to a drop in detection speed. Reference [27] introduced lightweight enhancements to the structure of YOLOv3 as well as an introduction to Res2Net [28] to improve the accuracy and speed of remote sensing target detection. In reference [29], the PPM (pyramid pooling module) [30] was added based on YOLOv4, and the Mish function was used to override the original activation function, which improved the detection precision and recall rate of aircraft and dockyards in remotely sensed imagery. Li et al. [31] proposed the YOLOSR-IST model. Based on the YOLOv5 method, this model introduces coordinate attention during the feature fusion process and integrates high-resolution maps.

However, these methods above do not give reasonable solutions for the problems of false detection and missed detection that occur in remote sensing image target detection. To address these challenges, we designed a remote sensing target detection algorithm YOLO-DRS based on YOLOv5. Our work makes the following main contributions.

- Based on the original EMA attention, a new module LEC(LDW-EMA-C3) is proposed for the fusion of a multi-scale lightweight efficient attention with the C3 structure in YOLOv5, replacing the last two C3 modules of the backbone with LDW-EMA to extract high-dimensional feature information at different scales.
- In the upsampling process of YOLOv5, the upsampling transposed convolution is introduced to replace the original nearest-neighbor interpolation upsampling to reduce the loss of the feature information of small targets in the upsampling process.

The rest of the paper is organized as follows: Section 3 introduces the YOLOv5 method. Section 4 introduces the methods proposed. Section 5 introduces the experimental part. Section 6 concludes with a summary of the paper.

3. The Basic Structure Of YOLOv5s

YOLOv5 is available in four different sizes based on depth and width: YOLOv5s, YOLOv5m, YOLOV5l, and YOLOv5x. As the model depth deepens and the width increases, YOLOv5 improves its detection accuracy, but the speed of detection decreases along with it. In this paper, we have selected YOLOv5s version 6.1, which combines both detection speed and accuracy advantages. YOLOv5s is mainly composed of four parts: input module (Input), backbone network module (Backbone), feature fusion module (Neck), and prediction module (Head). The overall architecture of YOLOv5s is shown in Figure 1.

The input side works as follows: first, a group of up to four images is scaled, aligned, or cropped to form a single image after capturing the enhanced image mosaic data. Secondly, the YOLOv5 algorithm adjusts the black edge by equidistant scaling and filling the smallest black edge with the smallest black edge, thus unifying the size of the image and preparing the neural network model for training. Figure 2 shows the picture enhanced by Mosaic4 data at the inputs.

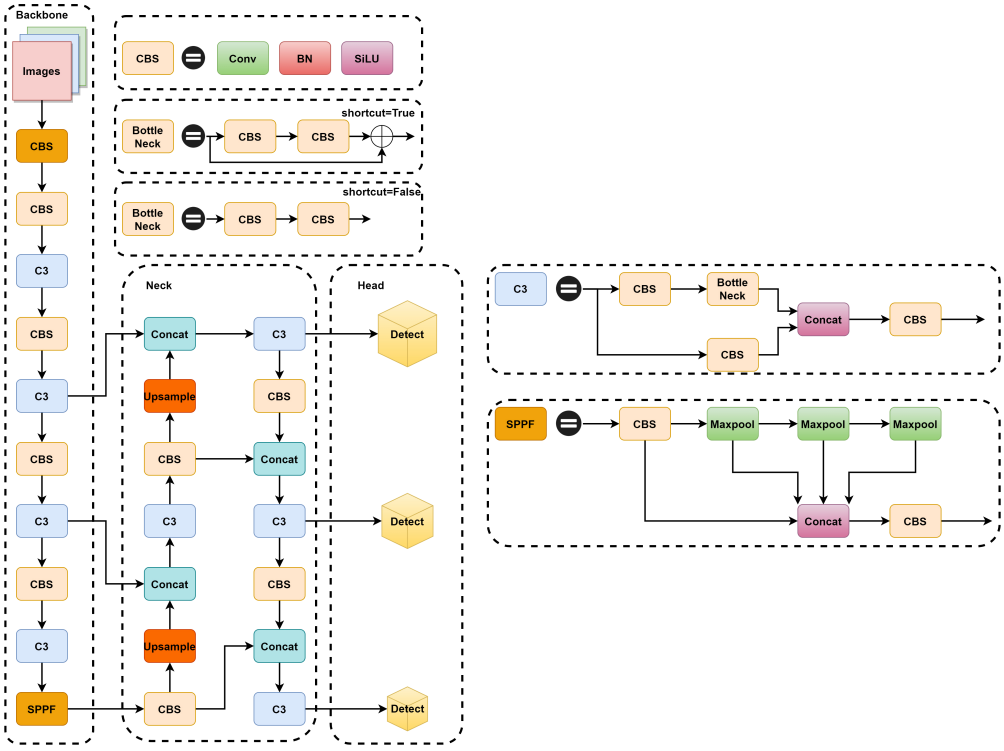

Figure 1. Structure of YOLOv5s.

The YOLOv5 backbone network consists mainly of the CSP, CBS, and SPPF structures. The CSP structure mainly draws on the idea of the cross-stage network CSPNet [32], where the input features are processed in two parts. The main part extracts features step by step through convolution, normalization, and activation functions, and the branches simply

adjust the channels through convolutional layers. By dividing the gradient information, a large amount of redundant gradient information is eliminated. The CBS structure consists of a convolution, Conv, a normalized BatchNorm [33], and an activation function, SiLU [34], which is used to extract the features of the model. The SPPF structure serially passes the input features through multiple 5×5 maximal pooling layers and then extracts the stacked features via the CBS network structure, which can increase the receptive field of the network and enhance the network's characterization capability.

Figure 2. Mosaic4 Enhanced Image.

The feature fusion module (Neck) is mainly composed of feature pyramid network (FPN) [35] and path aggregation network (PAN) [36] modules, which are responsible for fusing the feature maps of various scales and then decoding and generating feature maps containing more semantic information for input to the prediction module.

The YOLOv5s prediction module consists of three detection layers of different scales, 80×80, 40×40, and 20×20, which are used to predict the category and position prediction of small, medium, and large targets. The category information of the objects with the highest confidence scores is then output through post-processing operations, such as the non-maximum suppression algorithm.

YOLOv5s loss functions include cls_loss, box_loss, and obj_loss. The cls_loss and obj_loss are calculated using BCEWithLogitsLoss as shown in Equation (1).

$$C = -\frac{1}{n}\sum_{x}[ylna + (1-y)ln(1-a)] \tag{1}$$

where x denotes the sample, y denotes the label, a denotes the predicted output, and n denotes the total number of samples.

The box_loss is calculated via the IoU function [37]. The schematic and equations are shown in Figure 3 and Equation (2). YOLOv5s version 6.1 uses CIoU loss, as shown in Equation (3).

$$IOU = \frac{A \cap B}{A \cup B} \tag{2}$$

where $A \cap B$ is the area of overlap between the real frame and the predicted frame, and $A \cup B$ is the total area between the two.

$$CIoU = 1 - \frac{\rho^2(A,B)}{c^2} + \alpha v \tag{3}$$

where $\rho^2(A,B)$ represents the Euclidean distance between the centers of the predicted frame A and the real frame B, c represents the diagonal length of the smallest rectangle containing A and B, α represents the weight parameter, and v is used as a measure of the variability of the length, width, and height.

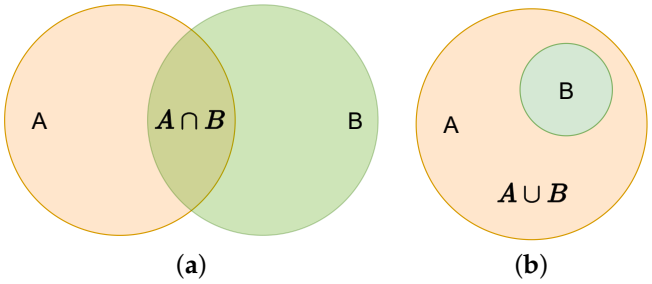

Figure 3. IOU loss. (**a**) $A \cap B$ is the intersection. (**b**) $A \cup B$ (Equal to A) is the union.

4. Proposed Method

The overall architecture of our proposed YOLO-GCRS is shown in Figure 4. First, the multi-scale feature information of the image can be fully extracted by the LEC module in the backbone network part. Then, the loss of small target feature information is reduced by the transposed convolutional upsampling method in the Neck section. Finally, the output target and category information is carried out through the prediction header.

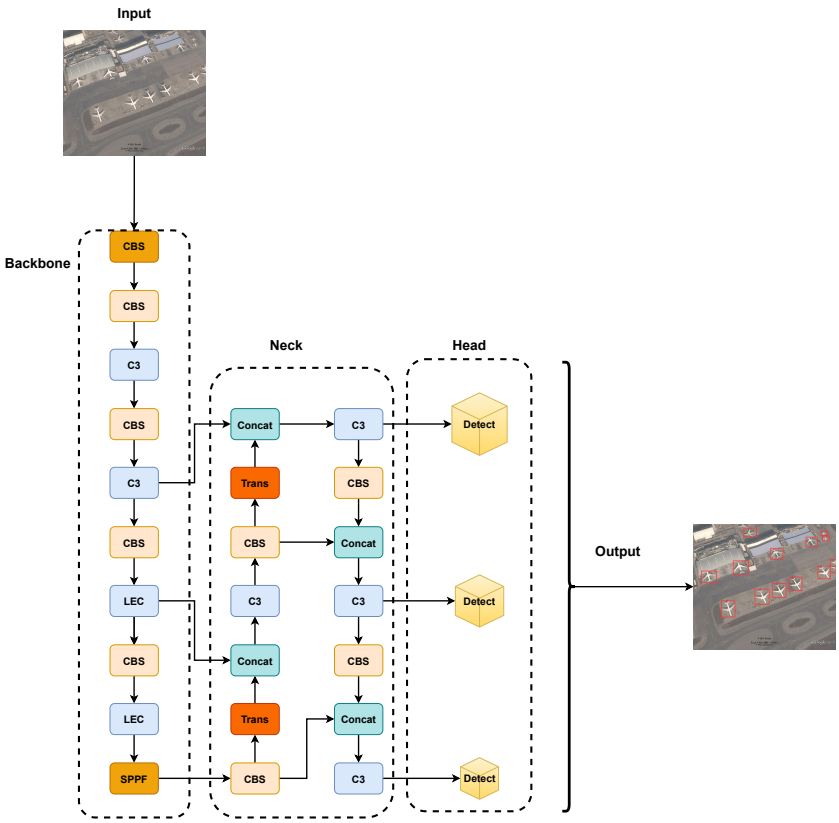

Figure 4. Structure of YOLO-DRS. (Trans is transposed convolution).

4.1. The LEC Module

With the development of deep convolutional neural networks, the attention mechanism has attracted great interest from the computer vision research community. The flexible structural features of the attentional mechanism approach not only enhance the learning of more discriminative feature representations but can also be easily inserted into the backbone architecture of neural networks.

It is widely recognized that there are three main mechanisms of attention that have been proposed, such as channeled attention, spatial attention, and both. As a representative of channel attention, SE [38] explicitly models cross-dimensional interactions to extract channel attention. The convolutional block attention module (CBAM) [39] builds cross-channel and cross-spatial information with semantic interdependencies between spatial and channel dimensions in the feature map. However, modeling cross-channel relationships using channel dimensionality reduction may introduce side effects when extracting deep visual representations. To solve these problems, Daliang Ouyang et al. proposed a new Efficient multiscale attention (EMA) [40] module by modifying the sequential processing of the CA [41] attention mechanism.

The general structure of EMA is shown in Figure 5. On the one hand, two coded features are connected in the image height direction and made to share the same 1×1 convolution without dimensionality reduction in the 1×1 branch by a similar process as CA. After decomposing the output of the 1×1 convolutional into two vectors, two nonlinear Sigmoid functions are used to fit a 2D binary distribution on the linear convolution. To realize different cross-channel interaction features between two parallel routes in a 1×1 branch, the two-channel attention maps within each group are aggregated together by simple multiplication. On the other hand, the 3×3 branch captures local cross-channel interactions via 3×3 convolutional to expand the feature space. In this way, EMA not only encodes inter-channel information to adjust the importance of different channels but also saves precise spatial structure information into the channels.

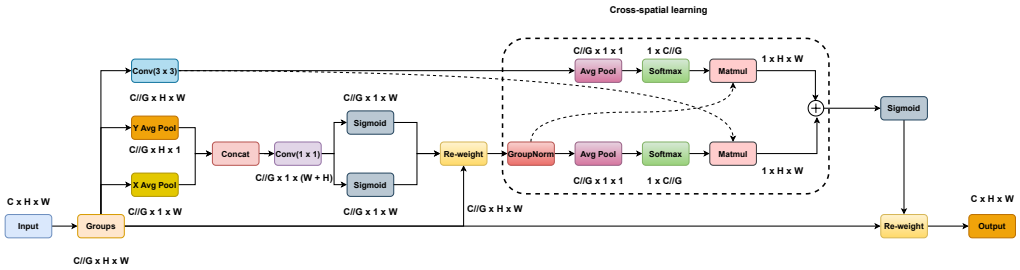

Figure 5. Structure of EMA.

Furthermore, a cross-spatial learning strategy is proposed in the EMA article, which is designed to encode global information and model long-range dependencies. For efficient computation, the natural nonlinear function Softmax for 2D Gaussian mapping is used at the output of the 2D global mean pooling (Avg Pool) to fit the linear transformation. The first spatial attention map was derived by multiplying the output of the above parallel processing with the matrix dot product operation. Similarly, 2D global average pooling is utilized to encode the global spatial information in the branch to derive a second spatial attention map that preserves the entire precise spatial location information. Finally, the output feature maps within each group were computed as an aggregation of the two generated spatial attention weight values. The sigmoid function captures pairwise relationships at the pixel level and highlights the global context of all pixels. The final output of the EMA is the same size as the input, which is efficient for stacking into modern architectures. The 2D global pooling and Softmax function formulas are (4) and (5), respectively.

$$Z_C = \frac{1}{H \times W} \sum_{j}^{H} \sum_{i}^{W} X_C(i, j) \tag{4}$$

where H, W, and C represent the height, width, and dimension of the input feature map, respectively.

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}} \quad (5)$$

where z_i is the output value of the ith node, C is the number of output nodes, and e is a constant term.

In order to extract the multi-scale feature information in the complex background without increasing the computation cost too much, first, in this paper, we propose a lightweight convolutional block, LDW. Then, based on LDW, we propose a lightweight multi-scale efficient attention mechanism module, LDW-EMA. Figure 6 and Figure 7 show the detailed structures of LDW and LDW-EMA, respectively.

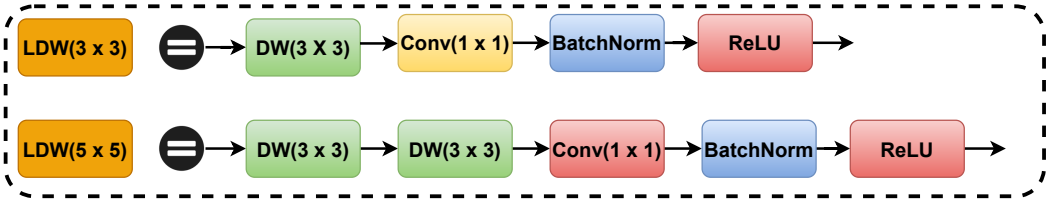

Figure 6. Structure of LDW convolutional block (DW is depthwise convolution).

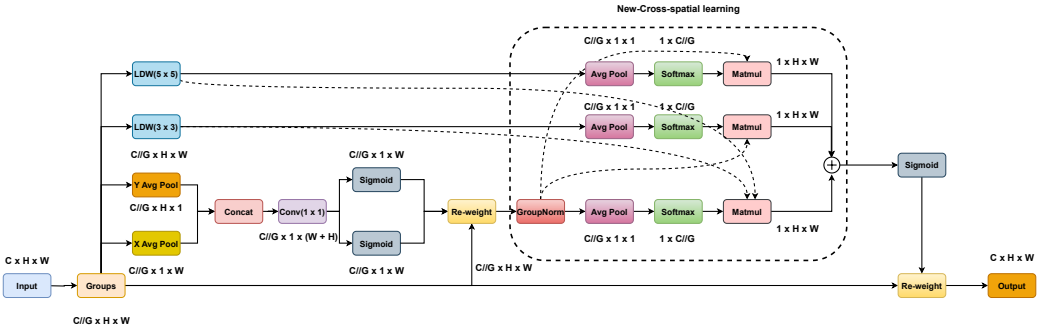

Figure 7. Structure of LDW-EMA.

The LDW convolutional block consists of Conv 1 × 1, DW3 × 3, BatchNorm, and ReLU activation functions. DSC (Deep separable convolution) consists of Conv 1 × 1 and DW 3 × 3. DSC dramatically reduces the convolutional parameters. BatchNorm speeds model convergence and improves the stability of the model. The ReLU activation function increases network non-linearity and prevents the gradient from disappearing. In addition, we use two DW 3 × 3 instead of one DW 5 × 5. This is because two DW 3 × 3 can achieve the same effect as one DW 5 × 5 with smaller parameters.

The LDW-EMA is composed of four branches: the principal branch, the coordinate branch, the 3 × 3 LDW, and the 5 × 5 LDW. Firstly, we used LDW 3 × 3 to replace the initial Conv 3 × 3. Then, we added a new 5 × 5 LDW branch and merged features from that branch with coordinate branch features for learning purposes. The new-cross-spatial learning consists of coordinate branches, LDW 3 × 3 and LDW 5 × 5, which can efficiently learn more multi-scale feature data.

Next, we fuse the proposed LDW-EMA with C3 of the YOLOv5 model Backbone to form the new module LEC. The overall structure of the LEC is shown in Figure 8. We replace the two C3 structures behind the Backbone layer with LEC, this is because the deeper features of the model are more difficult to extract and require more attention mechanisms to help, the shallow features can generally be extracted by the model more easily and accurately.

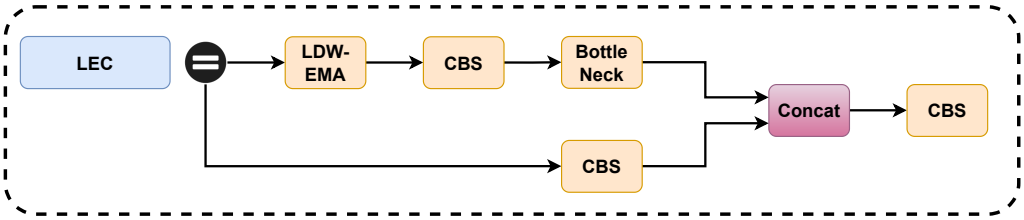

Figure 8. Structure of LEC Module.

4.2. Transposed Convolution

The original upsampling process of YOLOv5 used the up-adoption method of nearest-neighbor interpolation. In this method, upsampling is used where neighboring pixels are filled with blanks. In high-altitude remote sensing images, because background information is too complex and small targets occupy too many pixel points, the uppermost method of nearest-neighbor interpolation is equivalent to adding too much complex background information.

Transposed convolution can dynamically learn network-based weighting parameters, instead of fixing the use of a particular interpolation method when performing upsampling. Back in semantic segmentation, features would be extracted with a convolutional layer in the encoder, and then the original dimensions would be recovered in the decoder to categorize each pixel in the original image, a process that also requires transposed convolution. The classical methods are FCN [42] and U-Net [43].

The operation steps of transposed convolution can be divided into the following.

- Fill rows s-1 and column 0 between input feature mapping elements (where s denotes stride to transform convolution).
- Fill k-p-1 rows and column 0 around the input feature map (where k denotes the kernel_size size of the transposed convolutional and p is the padding of the transposed convolution).
- Flip the convolutional kernel parameters up and down, left and right.
- Perform normal convolution operations (padding = 0, stride = 1).

The following assumes that the input feature map is of size 2 × 2 (assuming that the input and output are single channels), and a feature map of size 4 × 4 is obtained after convolution by transposition(kernel_size = 3, stride = 1, padding = 0, ignore bias). Figure 9 shows the detailed execution of transposed convolution.

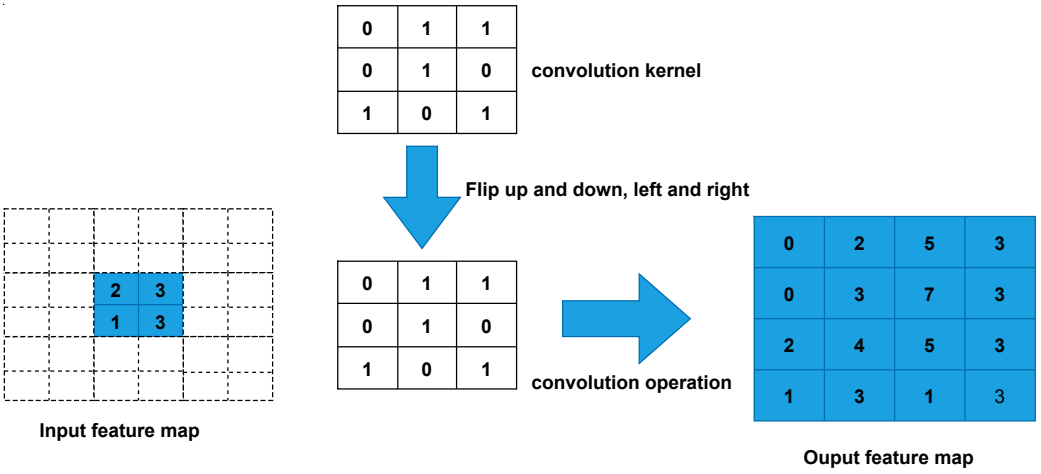

Figure 9. transposed convolution Computation Process.

- First, fill s-1 = 0 rows and column 0 (equal to 0 without padding) between elements.
- Second, fill k-p-1 = 2 rows and columns around the feature map 0.
- Third, the convolutional kernel parameters are flipped up and down, left and right.
- Finally, perform normal convolutional (padding = 0, stride = 1).

The size of the feature map after the transposed convolution operation can be calculated by Equations (6) and (7).

$$H_{out} = (H_{in} - 1) \times stride[0] - 2 \times padding[0] + kernel\_size[0] \quad (6)$$

$$W_{out} = (W_{in} - 1) \times stride[1] - 2 \times padding[1] + kernel\_size[1] \quad (7)$$

where stride[0] denotes stride in the height direction, padding[0] denotes padding in the height direction, kernel_size[0] denotes kernel_size in the height direction, and index [1] indicates width direction.

In this paper, we introduce the upsampling method of transposed convolution replacing the original nearest-neighbor interpolation. The transposed convolution can reduce the information loss when sampling small targets in the feature map as a way to solve problems such as missed detection and false detection of small targets in remote sensing images.

5. Experiments

In this paper, the experimental environment is shown in Table 1.

Table 1. Experimental Environment Configuration.

Project	Environment
Operating System	Ubuntu
CPU	E5-2680 v4
GPU	GeForce RTX 3060
Memory	14 GB
Pytorch version	1.10.0
CUDA	11.1

5.1. Datasets

RSOD is the remote sensing dataset employed in the experiment. RSOD is a publically available target detection dataset released by Wuhan University. There are four categories in the dataset: aircraft, playground, overpass, and oiltank. RSOD is labeled according to the PASCAL VOC dataset format.

Table 2 shows in detail the type and number of datasets.

Table 2. Distribution of Datasets.

Dataset Labeling	Number of Images
aircraft	446
playground	189
overpass	176
oiltank	165

In addition, the sample RSOD dataset and dataset characteristics are shown in Figures 10 and 11. As can be seen from Figures 10 and 11, the sample dataset has too many small target sizes and complex background information and is characterized by multi-scale distribution.

Figure 10. Sample Visualization of RSOD Dataset.

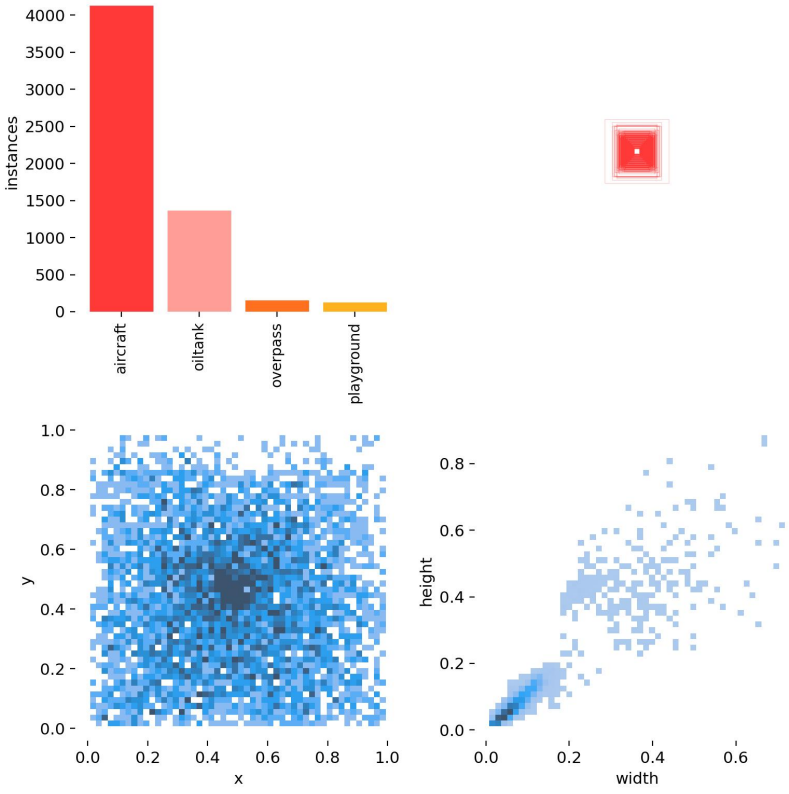

Figure 11. Label Information Distribution.

5.2. Evaluation Metrics

5.2.1. Precision

Precision is the rate of correct predictions among all results predicted for positive samples.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

where true positive (TP) means that the prediction is a positive example and the label value is also a positive example, and false positive (FP) means that the prediction is a positive example and the label value is a negative example.

5.2.2. Recall

Recall denotes the probability that of all the outcomes predicted to be positive samples, it is really a positive sample.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

where false negative (FN) indicates that the prediction is a negative example and the labeled value is a positive example.

5.2.3. Mean Average Precision

The mAP represents the average precision (AP) averaged over all categories.

$$mAP = \frac{1}{N} \sum AP_i \tag{10}$$

where N represents the total number of categories and AP_i represents the average precision in category i.

mAP@0.5 denotes the average accuracy value of the IoU parameter when selected as a 0.5 threshold.

5.2.4. FLOPs

FLOPs (floating-point of operations) is the number of floating-point operations, understood as the amount of computation, which can be used to measure algorithm complexity.

5.2.5. FPS

FPS is defined in the field of graphics as the number of frames transmitted per second of the picture. The FPS unit is $frame/s$.

$$FPS = \frac{Frames}{Time} \tag{11}$$

In this experiment, the FPS on the GPU was selected as the criterion.

5.3. Network Training and Parameter Setting

5.3.1. Parameter Setting

In this paper, the detailed training parameter settings are shown in Table 3.

Table 3. Experimental Parameter Setting.

Parameters	Value
weights	yolov5s.pt
division ratio	7:2:1 (train:val:test)
optimizer	SGD
batch size	16
epochs	100

Where yolov5s.pt comes from the pre-training weights learned from ImageNet migration, and the division ratio is the proportion of dataset division.

5.3.2. Network Training

The loss function curve shows the results of network training in the most straightforward way. In this paper, the loss function consists of three main components: cls_loss, box_loss, and obj_loss.

$$L_{loss} = L_{cls} + L_{obj} + L_{box} \tag{12}$$

where L_{cls},L_{obj} and L_{box} represent cls_loss, obj_loss, and box_loss, respectively.

Therefore, we can tell how well the network is trained by observing these three types of loss function images. The loss function curves for each category are shown in Figure 12. From the visualization results, it can be concluded that the YOLO-GCRS model loss decreases with the increase in the number of iterations, and the loss value tends to be stable and close to 0 after the number of iterations reaches 80, indicating that the model training has reached the optimal effect.

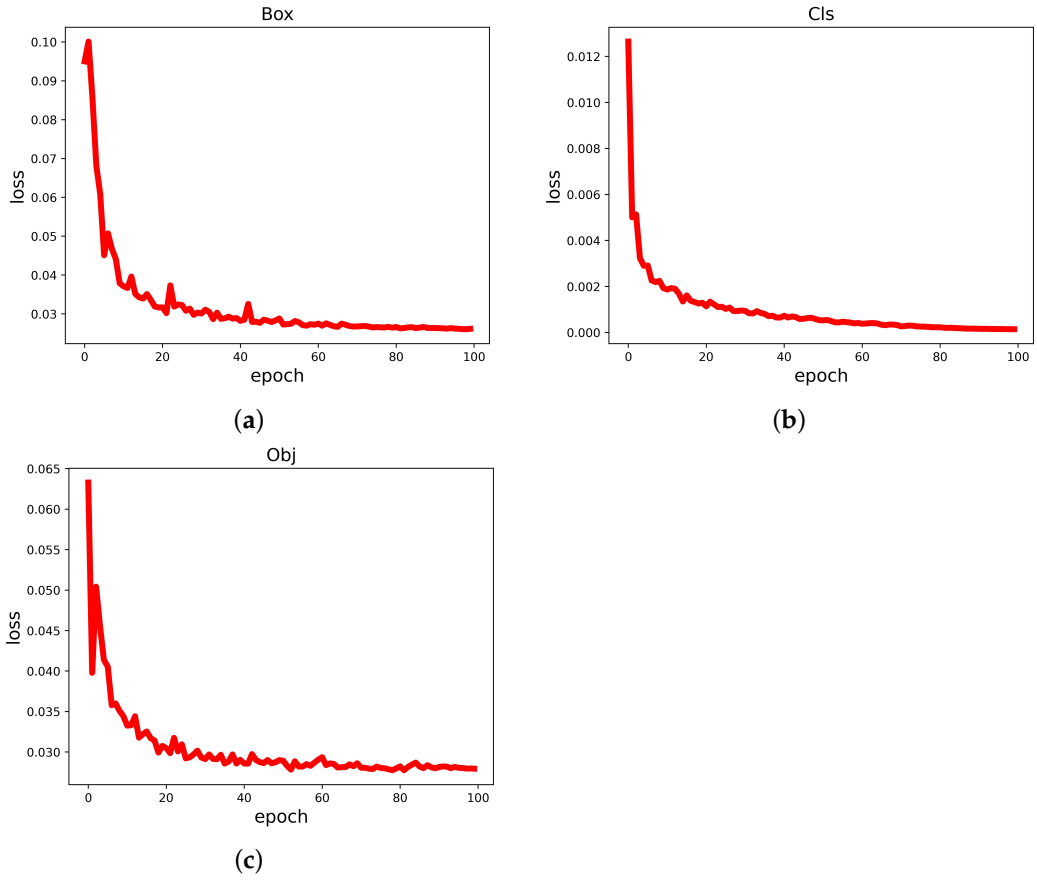

Figure 12. Loss Curve. (**a**) Box Loss. (**b**) Cls Loss. (**c**) Obj Loss.

5.4. Analysis of Results

In this paper, we have conducted extensive ablation experiments to demonstrate the effectiveness and sophistication of the designed module. So, as you know, the ablation experimental data were obtained on the validation set.

Firstly, we discuss the embedding location of the LDW-EMA module in the LEC structure. We name the cases where LDW-EMA is added to the residual structure branch of the LEC structure, the CBS branch, and both as LEC-top, LEC-bottom, and LEC-both, respectively. Table 4 shows detailed experimental data on the different positions of LDW-EMA in the LEC structure.

Table 4. Comparison of LEC Experiments at Different Locations.

Method	Precision	Recall	mAP@0.5	FLOPs
YOLOv5s	0.930	0.939	0.950	15.8
+LEC-top	0.959	0.956	0.972	15.9
+LEC-bottom	0.962	0.925	0.970	15.9
+LEC-both	0.967	0.936	0.963	16.0

Overall, it is clear that the LEC-top is an optimal outcome across all metrics. In particular, LEC-top achieved the best results on mAP@0.5. Therefore, we finally chose LEC-top as the structure of the LEC module.

Secondly, the activation function of LDW is discussed after determining the LEC locational structure. This is because the activation function has an important influence on the convergence and training effect of the model. We discuss the activation functions of RELU, Mish, and SILU. Table 5 shows the results of detailed experimental data on different activation functions in the LDW structure.

Table 5. Experimental Comparison of Different Loss Functions.

Method	Precision	Recall	mAP@0.5	FLOPs
YOLOv5s	0.930	0.939	0.950	15.8
+LDW (ReLU)	0.959	0.956	0.972	15.9
+LDW (SiLU)	0.953	0.975	0.967	15.9
+LDW (Mish)	0.961	0.952	0.968	15.9

It can be seen that the model is able to obtain the highest mAP@0.5 when the LDW module uses the ReLU activation function. Therefore, we choose ReLU as the activation function of LDW.

Thirdly, the position of the sample on the transformation convolution is discussed. The nearest-neighbor upsampling interpolation algorithm is replaced by a bottom-up transposed convolution, including replacing the former, replacing the latter, and replacing all. We named them Trans-first, Trans-second, and Trans-both. Table 6 shows the experimental data of transposed convolution at different locations.

Table 6. Experimental Comparison of Transposed Convolution at Different Positions.

Method	Precision	Recall	mAP@0.5	FLOPs
YOLOv5s	0.930	0.939	0.950	15.8
+Trans-first	0.963	0.939	0.961	15.8
+Trans-second	0.966	0.936	0.975	15.8
+Trans-both	0.959	0.971	0.971	15.8

For this experiment, we selected Trans-both as the transposed convolution structure. This is the result of synthesizing Trans-both in precision, recall, and mAP@0.5. Although the Trans-2 mAP@0.5 is not the best.

Next, we discuss various cases of EMA in relation to each other. Among them, LDW (3×3) denotes the replacement of the convolutional of 3×3 branches in EMA with our proposed LDW module. Table 7 shows the detailed data of the experiments for different cases of EMA. It should be noted that these data all have the same structure as the LEC except that the mechanisms of integrated attention are different.

Clearly, our proposed LDW-EMA performs well on precision, recall, and mAP0.5. And the LDW-EMA achieves the highest value of mAP.

Table 7. Experimental Comparison of Different EMA Attention Mechanisms.

Method	Precision	Recall	mAP@0.5	FLOPs
YOLOv5s	0.930	0.939	0.950	15.8
+EMA	0.937	0.957	0.956	15.8
+LDW (3×3)	0.921	0.960	0.964	15.8
+LDW-EMA	0.959	0.956	0.972	16.0

We then compare the LEC module proposed in this paper with mainstream attention mechanisms, such as CA, SE, and ECA [44]. We use the YOLOv5s base model in conjunction with each attention mechanism separately. Notably, to ensure the same structure as the LEC, we also integrated the various attentional mechanisms with the two C3 structures at the back of the YOLOv5s backbone. Table 8 shows detailed experimental data for the various different attentional mechanisms.

It is easy to find that LEC achieves the optimal result on the mAP0.5 metric, and collectively, P and R also perform very well, which is sufficient to show that the LEC module we designed embodies better than other mainstream attention mechanisms.

Table 8. Experimental Comparison of Mainstream Attention Mechanisms.

Method	Precision	Recall	mAP@0.5	FLOPs
YOLOv5s	0.930	0.939	0.950	15.8
+C3CA	0.970	0.935	0.957	15.8
+C3ECA	0.950	0.940	0.955	15.8
+C3SE	0.990	0.923	0.968	15.8
+LEC	0.959	0.956	0.972	16.0

In addition, in order to visualize the practicality of the innovations in each module of this paper, we conducted ablation experiments on the YOLO-DRS algorithm on the validation set. Table 9 demonstrates the detailed data of the ablation experiments.

Table 9. Ablation Experiment.

Method	Precision	Recall	mAP@0.5	FLOPs	FPS/(frame/s)
YOLOv5s	0.930	0.939	0.950	15.8	76.8
+LEC	0.959	0.956	0.972	16.0	51.9
+Trans-both	0.959	0.971	0.971	15.8	69.7
YOLO-DRS	0.953	0.971	0.975	16.0	53.9

Analyzing the data in Table 9, the LEC and transposed convolution proposed show a large improvement over the original YOLOv5s in P, R, and mAP@0.5.

Moreover, our proposed YOLO-DRS algorithm improves **2.3%**, **3.2%**, and **2.5%** on P, R, and mAP@0.5, respectively, compared with the original YOLOv5s, and the GFLOPS increases only by 0.2. Also, the FPS is within the real-time detection frame rate range. These data fully prove that the proposed YOLO-DRS algorithm is very effective.

Lastly, in order to check the sophistication of the YOLO-DRS algorithm, we use the same network metrics to compare it to the current state-of-the-art target detection algorithms of the same class. Table 10 shows the detailed data of YOLO-DRS experiments with different advanced algorithms.

It can be seen from Table 10 that our proposed YOLO-DRS achieves the best results on the P, R, and mAP@0.5 metrics, which is sufficient to prove the speed and sophistication of our proposed algorithms.

Table 10. Experimental Comparison of Mainstream Algorithms.

Method	Precision	Recall	mAP@0.5
YOLOv5s	0.930	0.939	0.950
YOLOv7-tiny	0.953	0.957	0.957
YOLOv8s	0.871	0.864	0.902
YOLO-DRS	0.953	0.971	0.975

5.5. Visualization Experiments

To more intuitively reflect the algorithm's solution to the problem of detecting RSOD datasets, we performed visualization and comparison experiments in a variety of scenarios. Note that the image used for the visualization experiment is the RSOD test dataset.

Firstly, Figure 13 illustrates small-target missed detection and false detection.

Obviously, YOLO-DRS can solve the problem of small-target aircraft missed and false detection and improve the detection accuracy of the model.

Secondly, Figure 14 demonstrates the average accuracy of detection of the target. It is not hard to see that YOLO-DRS is able to achieve improved detection accuracy.

Figure 13. Small-target missed detection and false detection in complex backgrounds. The detection results of YOLOv5s are shown on the **left** and the results of YOLO-DRS are shown on the **right**.

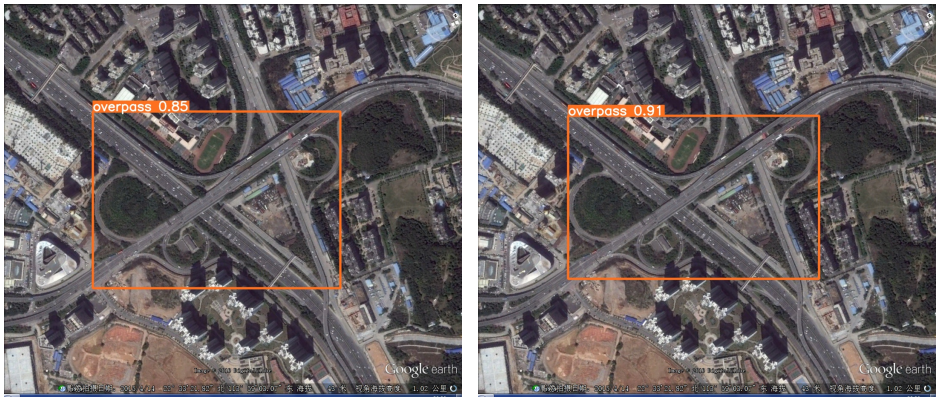

Figure 14. Detection accuracy of models in complex backgrounds. The detection results of YOLOv5s are shown on the **left** and the results of YOLO-DRS are shown on the **right**.

Thirdly, Figure 15 illustrates the detection results of the target on multiple scales. The background information in Figure 15 is complex and the aircraft types are characterized by a multi-scale distribution. YOLO-DRS can greatly reduce the problem of missed detection.

Figure 15. Multi-scale small-target missed detection in complex backgrounds. The detection results of YOLOv5s are shown on the **left** and the results of YOLO-DRS are shown on the **right**.

Finally, Figure 16 illustrates the detection of large scales with complex background information. YOLO-DRS still performs well in detecting large-scale targets and is able to reduce the problem of false detection of large-scale targets under complex background information.

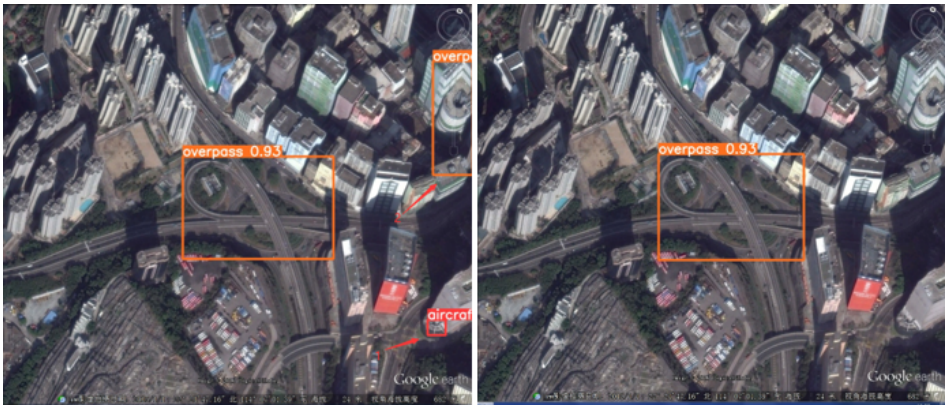

Figure 16. Large-target false detection in complex backgrounds. The detection results of YOLOv5s are shown on the **left** and the results of YOLO-DRS are shown on the **right**.

In conclusion, the YOLO-DRS proposed in this paper can solve the problems of low average detection accuracy, false detection, and missed detection caused by the characteristics of many small objects in remote sensing images, with multi-scale distribution and complex background information.

6. Conclusions

Based on YOLOv5s, we propose YOLO-DRS, a lightweight remote sensing image object detection algorithm that fuses multiple scales efficiently. Firstly, we propose an efficient and lightweight multi-scale attention mechanism, LEC, that is able to capture multi-scale target features under complex background information with little computational overhead. Then, we introduce the transposed convolutional replacement nearest-neighbor upsampling algorithm, which can dynamically learn the feature information and can reduce the loss of target feature information during the upsampling process.

On the RSOD dataset, we obtained 97.5% mAP@0.5, an improvement of 2.5% over the original YOLOv5s, and only an increase of 0.2 in FLOPs. In addition, YOLO-DRS improves the mAP@0.5 metrics number by 1.8% and 7.3% compared to the state-of-the-art algorithms YOLOv8s and YOLOv7-tiny, respectively. In summary, the YOLO-DRS algorithm is able to solve the problem of low average accuracy of detection, false detection, and missed

detection in remote sensing images due to the complex background information, many-small-target multi-scale distribution, and other characteristics. Moving forward, we will explore the study of pruning and lightweighting the model without degrading the detection accuracy so that it can be better deployed for grounded applications.

Author Contributions: Writing—original draft, H.L.;Writing—review and editing, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Hunan Province (funding number: 2021JJ50058), the Open Platform Innovation Foundation of the Education Department of Hunan (funding number: 20K046), and the National Key Research and Development Program (NKRDP) projects (funding number: 2019QY1604).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Liu, F.; Zhu, J.; Wang, W.; Kuang, M. Surface-to-air missile sites detection agent with remote sensing images. *Sci. China Inf. Sci.* **2021**, *64*, 1–3. [CrossRef]
2. Zhang, Y.; Ning, G.; Chen, S.; Yang, Y. Impact of rapid urban sprawl on the local meteorological observational environment based on remote sensing images and GIS technology. *Remote Sens.* **2021**, *13*, 2624. [CrossRef]
3. Friedman, J.; Hastie, T.; Tibshirani, R. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Stat.* **2000**, *28*, 337–407. [CrossRef]
4. Platt, J.C. *Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines*; Microsoft: Redmond, WA, USA, 1998.
5. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
6. Felzenszwalb, P.F.; Girshick, R.B.; Mcallester, D.A. Cascade object detection with deformable part models. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
7. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
11. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
12. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
13. 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, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
14. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
16. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 7263–7271.
17. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
18. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
19. Ullah, F.; Ullah, S.; Naeem, M.R.; Mostarda, L.; Rho, S.; Cheng, X. Cyber-threat detection system using a hybrid approach of transfer learning and multi-model image representation. *Sensors* **2022**, *22*, 5883. [CrossRef] [PubMed]
20. Du, F.J.; Jiao, S.J. Improvement of lightweight convolutional neural network model based on YOLO algorithm and its research in pavement defect detection. *Sensors* **2022**, *22*, 3537. [CrossRef] [PubMed]
21. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef] [PubMed]

22. Shen, L.; Tao, H.; Ni, Y.; Wang, Y.; Stojanovic, V. Improved YOLOv3 model with feature map cropping for multi-scale road object detection. *Meas. Sci. Technol.* **2023**, *34*, 045406. [CrossRef]
23. Wang, J.; Zhang, F.; Zhang, Y.; Liu, Y.; Cheng, T. Lightweight Object Detection Algorithm for UAV Aerial Imagery. *Sensors* **2023**, *23*, 5786. [CrossRef]
24. Qu, Z.; Zhu, F.; Qi, C. Remote sensing image target detection: Improvement of the YOLOv3 model with auxiliary networks. *Remote. Sens.* **2021**, *13*, 3908. [CrossRef]
25. Xu, D.; Wu, Y. Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. *Sensors* **2020**, *20*, 4276. [PubMed]
26. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 4700–4708.
27. Xu, D.; Wu, Y. FE-YOLO: A feature enhancement network for remote sensing target detection. *Remote Sens.* **2021**, *13*, 1311. [CrossRef]
28. Gao, S.H.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
29. Cao, C.; Wu, J.; Zeng, X.; Feng, Z.; Wang, T.; Yan, X.; Wu, Z.; Wu, Q.; Huang, Z. Research on airplane and ship detection of aerial remote sensing images based on convolutional neural network. *Sensors* **2020**, *20*, 4696. [CrossRef] [PubMed]
30. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 2881–2890.
31. Li, R.; Shen, Y. YOLOSR-IST: A deep learning method for small target detection in infrared remote sensing images based on super-resolution and YOLO. *Signal Process.* **2023**, *208*, 108962. [CrossRef]
32. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 390–391.
33. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 June 2015; pp. 448–456.
34. Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* **2018**, *107*, 3–11. [CrossRef] [PubMed]
35. 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, Honolulu, HI, USA, 21–26 June 2017; pp. 2117–2125.
36. 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, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
37. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
38. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
39. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
40. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
41. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
42. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
43. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; Proceedings, Part III 18; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
44. 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/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.

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