Mathematical Biosciences and Engineering

MBE, 20(8): 13947–13973.
DOI: 10.3934/mbe.2023621
Received: 07 March 2023
Revised: 17 May 2023
Accepted: 05 June 2023
Published: 20 June 2023

http://www.aimspress.com/journal/MBE

Research article

An ultra-lightweight detector with high accuracy and speed for aerial images

Lei Yang[1], Guowu Yuan[1,2,*], Hao Wu[1,2] and Wenhua Qian[1,2]

[1] School of Information Science and Engineering, Yunnan University, Kunming 650504, Yunnan, China
[2] Yunnan Key Laboratory of Intelligent Systems and Computing, Kunming 650504, Yunnan, China

* **Correspondence:** Email: gwyuan@ynu.edu.cn; Tel: +8687165033748.

Abstract: Aerial remote sensing images have complex backgrounds and numerous small targets compared to natural images, so detecting targets in aerial images is more difficult. Resource exploration and urban construction planning need to detect targets quickly and accurately in aerial images. High accuracy is undoubtedly the advantage for detection models in target detection. However, high accuracy often means more complex models with larger computational and parametric quantities. Lightweight models are fast to detect, but detection accuracy is much lower than conventional models. It is challenging to balance the accuracy and speed of the model in remote sensing image detection. In this paper, we proposed a new YOLO model. We incorporated the structures of YOLOX-Nano and slim-neck, then used the SPPF module and SIoU function. In addition, we designed a new upsampling paradigm that combined linear interpolation and attention mechanism, which can effectively improve the model's accuracy. Compared with the original YOLOX-Nano, our model had better accuracy and speed balance while maintaining the model's lightweight. The experimental results showed that our model achieved high accuracy and speed on NWPU VHR-10, RSOD, TGRS-HRRSD and DOTA datasets.

Keywords: deep learning; lightweight model; object detection; aerial image; YOLOX

1. Introduction

Along with the rapid development of aerial photography technology, aerial remote sensing data are becoming increasingly diversified. Data acquisition speed is accelerating, the update cycle is

shortening and the timeliness is becoming stronger. Therefore, automatic target detection technology in aerial images came into being. The technology is widely used in the fields of urban traffic planning [1], water conservancy construction [2], earth resource exploration [3] and military information processing [4]. Humans can already use UAVs to take numerous aerial images, or even to enable real-time monitoring of target areas. The UAVs need to carry a lightweight model with the highest possible precision to achieve real-time detection of ground targets.

Currently, the mainstream target detection models include two major categories: the two-stage algorithms represented by the R-CNN series [5–7], and the one-stage algorithms represented by the YOLO series [8–13]. The two-stage algorithms first extract the candidate boxes for the input images and then classify and regress the candidate boxes with high accuracy but low speed. The one-stage algorithms directly calculate the class probabilities and position coordinates of the targets in the input images. They are faster than the two-stage algorithms but not as accurate. In recent years, one-stage models have made breakthroughs. With the proposal of excellent models, such as YOLOv4 [11], YOLOv5 [12] and YOLOX [13], it has been found that one-stage models can already match the detection accuracy of two-stage models while maintaining their high-speed characteristics. These models work well on ordinary images but do not perform well on aerial images. Y. Li et al. [14] believed that the detection accuracy of ordinary models is not high because of the special perspective of aerial images. Aerial remote sensing images are mostly taken at high altitudes at a top angle. Compared to conventional images captured horizontally, the targets' size is small, the extractable features are few and the background is complex.

As a result, many researchers have proposed various improved models based on deep learning for aerial image detection. For example, A. V. Etten [15] proposed the YOLT model by improving YOLOv2, which enhanced the detection performance for small targets by connecting features of multiple layers to obtain a more fine-grained feature representation through a ResNet-like residual structure. M. Ahmed et al. [16] proposed the Fused RetinaNet model, which uses a new contextual fusion module instead of a feature pyramid network to improve the representation of the underlying semantic and top-level spatial information. H. Liu et al. [17] introduced a hybrid attention module and variable convolution in the C-CenterNet model to enhance the feature extraction and fusion of the model. S. Du et al. [18] proposed an improved YOLO model using a negative sample focusing mechanism and an inflated convolutional attention module to improve the detection accuracy of the model for small targets.

Although the above models have achieved high detection accuracy, they are not lightweight and their detection speed is slower than those using lightweight structures. The commonly used lightweight structures, including MobileNet [19–21] and ShuffleNet [22,23], can achieve high detection speed but low accuracy. However, some remote sensing applications, such as the real-time detection of UAVs, need a high accuracy and speed model. Balancing the speed and accuracy of the detection model is the key problem in these applications, and also the focus of this paper. This work uses YOLOX-Nano [13] as the basic model. On the premise of ensuring the lightweight of the model, some deep learning methods are used to improve the detection accuracy and speed of the model as much as possible. Four aerial image datasets (NWPU VHR-10, RSOD, TGRS-HRRSD and DOTA) are tested to verify the generalization ability of our improved methods.

The main contributions of this paper are as follows:

1) This paper proposes a new lightweight model for remote sensing image detection, which has high precision and speed.

2) This paper incorporates the YOLOX-Nano model and slim-neck structure to reconstruct the network, which is very effective for balancing the speed and accuracy of the detection.

3) This paper proposes a new upsampling paradigm that combines linear interpolation and attention mechanisms. This paradigm can effectively improve detection accuracy.

The rest of this paper is organized as follows: Section 2 introduces the current popular lightweight and remote sensing detection models, and then introduces the YOLOX-Nano model that inspired this article. Section 3 focuses on our methods to improve YOLOX-Nano. Section 4 presents the experimental results of our methods and our analysis. Section 5 concludes our work.

2. Related work

2.1. Lightweight models in target detection

To achieve real-time detection in some specific situations, researchers have proposed some lighter models, such as the MobileNet family built with depthwise separable convolution (DWC) [19–21] and the ShuffleNet family made with grouped point-by-point convolution [22,23]. Some researchers have used these lightweight structures for detection, such as the YOLOX-Nano model proposed by Z. Ge et al., which uses the DWC in MobileNet to replace the regular convolution [13]. RangiLyu used ShuffleNet as the backbone network to build the NanoDet model, greatly reducing the number of parameters and the computation of the model [24].

Some conventional target detection models, such as YOLOv3, YOLOv4, etc., also have their lightweight versions, e.g., YOLOv3-Tiny and YOLOv4-Tiny. These lightweight models are obtained by simplifying conventional models' infrastructure and reducing the original models' computational effort. YOLOv3-Tiny does not use residual structures, employs only a few conventional convolutional structures in the backbone network, significantly reducing the depth and uses only two feature layers in the neck network for classification and regression prediction. The backbone network of YOLOv4-Tiny uses the CSPNet [25] structure, and the neck network uses only two feature layers as outputs. Compared to YOLOv3 and YOLOv4, these models are very lightweight and are widely used in industry, but there is no doubt that this simplification significantly reduces the accuracy of the models.

2.2. Target detection in remote sensing images

Aerial remote sensing images are large and complex in the background, so they contain many small targets that are difficult to detect. After thoroughly studying the characteristics of remote sensing images, previous researchers have proposed a series of solutions.

Some researchers have tried to introduce attention mechanisms and feature fusion algorithms to extract the most valuable features possible, improving the accuracy of small-target detection. For example, X. Luo et al. [26] added the improved efficient channel attention module (IECA) and the adaptive feature fusion algorithm (ASFF) to the YOLOv4 model, greatly improving the detection accuracy. D. Yan et al. [27] introduced the SE attention mechanism and feature pyramid structure (FPN) into the Faster R-CNN model to achieve high-accuracy detection of tailing pools in remotely sensed images. Some researchers have addressed the challenge of small-target detection from the remote sensing images' pre-processing and post-processing stages. For example, F. C. Akyon et al.

[28] proposed the Slicing-Aided Hyper Inference (SAHI) framework, in which a large image is cut with overlap before it is detected. Then, each slice is fed to the detector one by one. Finally, the detection results are combined into a large complete image in the post-processing stage. L. Yang et al. [29] introduced the SAHI framework into their improved YOLOX model to achieve high-precision automatic detection of small objects in large remote sensing images.

All of the above researchers have made efforts and achieved good results in improving the accuracy of neural network models for remote sensing image detection. However, the methods they have adopted have increased the number of parameters and the computational effort of the original model. Therefore, they are not conducive to achieving real-time detection of aerial targets. Some researchers have applied lightweight models to remote sensing images to improve the model's speed. For example, J. Liu et al. [30] used a modified YOLOv4-Tiny to achieve real-time detection of insulator identification and defects in aerial images, while X. Li et al. [31] used MobileNet to replace the backbone network of the YOLO model, significantly reducing the parameters and computation and achieving fast detection of remotely sensed images.

Although the lightweight models have greatly improved the detection speed and achieved real-time detection, they also show a significant decrease in detection accuracy. How to balance the accuracy and speed of detection is the focus of our research. In this paper, we use YOLOX-Nano as the base model for improvement, and the following section provides a detailed description of our methodology.

2.3. About YOLOX-Nano

YOLOX is a new YOLO model proposed by Z. Ge et al. in 2021 [13], surpassing all previous YOLO versions in detection performance. YOLOX has made many improvements based on the previous YOLO version, improving detection accuracy and speed. YOLOX-Nano is a lightweight version of YOLOX that uses depthwise separable convolution to build the network. Its model structure is shown in Figure 1, and the overall model contains three parts: backbone network, neck network and detection head. The backbone network is used to implement feature extraction, the neck network is used to implement feature fusion and the detection head is used to calculate the class and location coordinates of a target. YOLOX-Nano's backbone is CSPDarknet. It employs a cross-stage partial (CSP) structure [25]. The structure divides the input into two parts: the one processed by a bottleneck structure and the other used by a short-circuit connection. Then, the model splices the two parts. The structure reduces memory consumption and enhances CNN's learning ability. The neck network uses a PAFPN structure to achieve information fusion of low-level and high-level features. The detection head uses a decoupled detection head, which trains the parameters used for classification and regression separately and calculates each branch, helping to improve the accuracy of classification and localization.

Comparing the previous YOLO model, the main improvement of YOLOX-Nano is using an anchor-free mode [32,33] and a decoupling head [34]. The anchor-free mode abandons the predefined anchor boxes and directly predicts the target's position by detecting the object's center point. Compared with the anchor-based mode used in the previous version of YOLO, the anchor-free mode eliminates the predefined anchor boxes. It dramatically reduces greatly the computation to achieve real-time high-precision detection further. The earlier versions of YOLO coupled both classification and regression in a single detection head. However, some studies [34] in recent years

have shown that classification and regression can interfere with one another, so decoupling classification and regression can improve the model's performance. The YOLOX-Nano adopts this idea by using a decoupled detection head to decouple classification and regression. Then, it adds a branch to calculate the object's confidence for determining whether it is in the foreground or background.

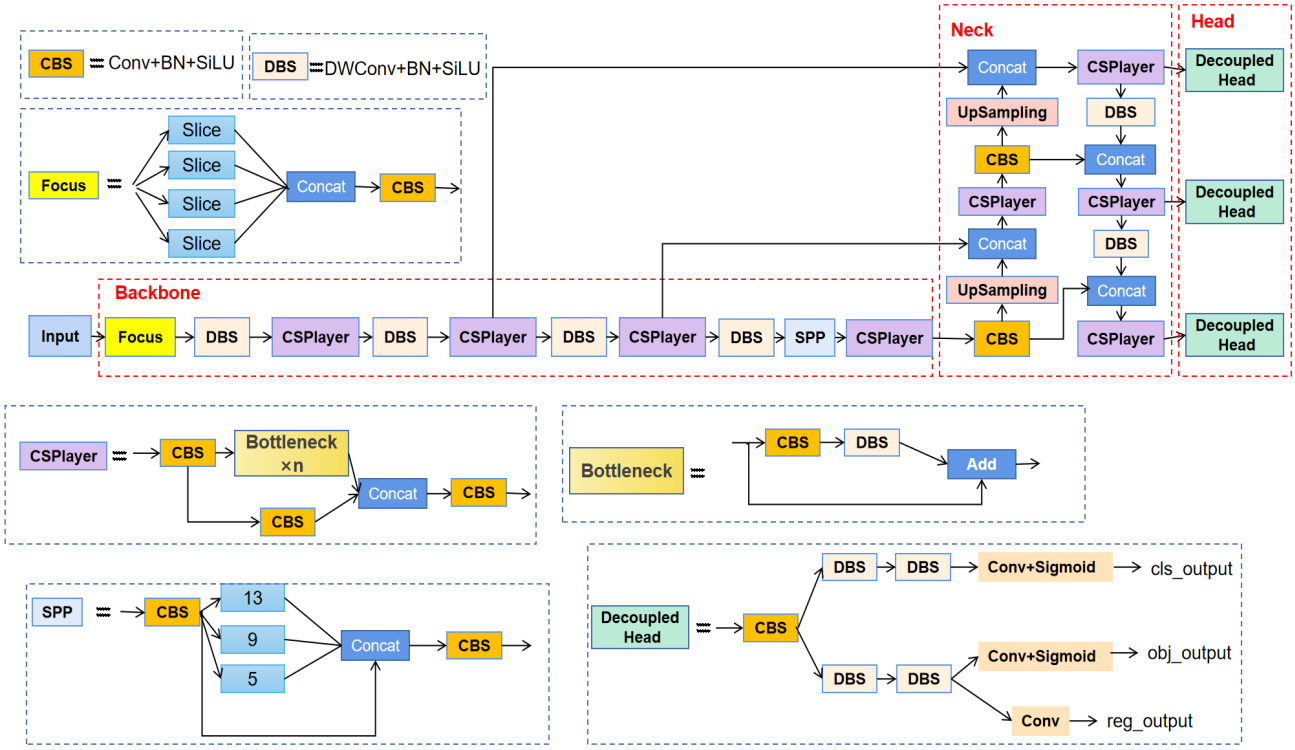


Figure 1. Structure of the YOLOX-Nano model.

YOLOX-Nano dramatically reduces the number of parameters in the model by using depthwise separable convolution. The model's size is only 0.9 M and YOLOX-Nano is the smallest and fastest version of YOLOX. To achieve real-time, high-precision aerial remote sensing image detection, we use YOLOX-Nano for training and detection on aerial remote sensing image datasets.

3. Methods

3.1. Our improved YOLOX-Nano model

This paper makes a series of improvements to YOLOX-Nano, and obtains a new lightweight model with the network structure shown in Figure 2. This model uses a more efficient spatial pyramid structure (SPPF) in the backbone network, which constructs a new slim-neck structure by replacing DWC with GSConv and CSP with VoV-GSCSP in the neck network. At the same time, we design a new upsampled structure (linear interpolation+ECA attention mechanism), and finally use the latest localization loss function SIoU to calculate the loss in the head. The SPPF and SIoU improve the model's detection speed and accuracy. Although the GSConv and new upsampled structure slightly increase the parameters and inference time, they enhance the model's fusion of remote sensing image features, which is essential for improving model accuracy.

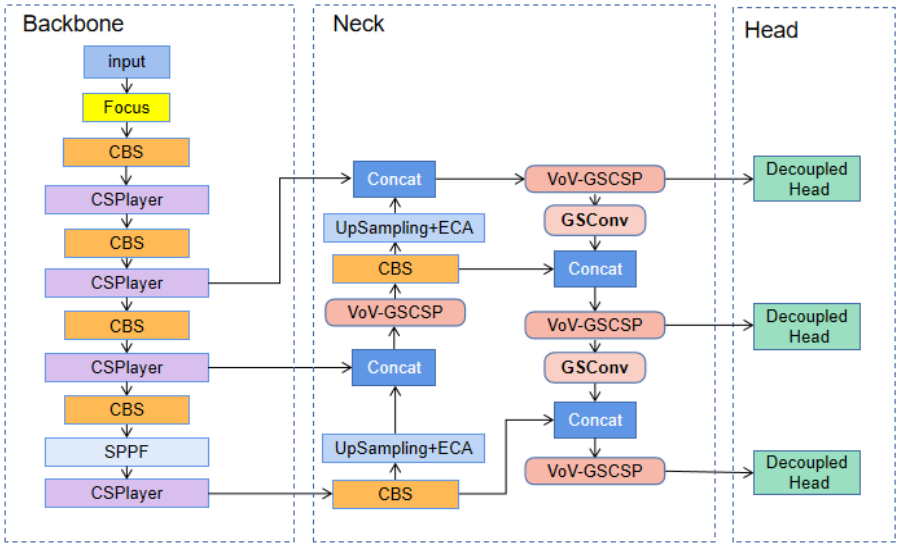


Figure 2. Our improved YOLOX-Nano model.

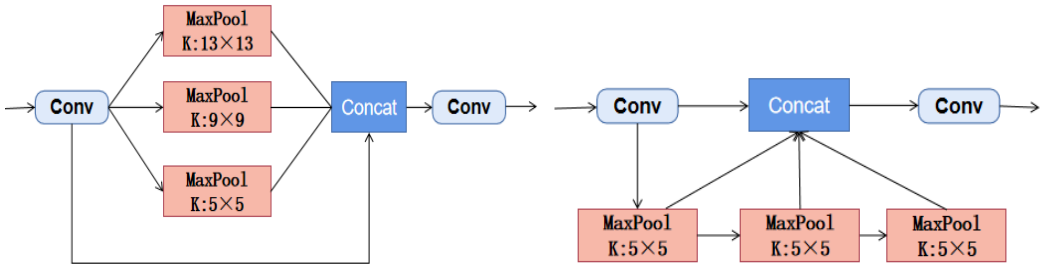
Our improvements are all about balancing the speed and accuracy of the model as much as possible. This paper reconstructs the network using some lightweight structures, and tries to avoid algorithms that increase the complexity of the model.

This paper improves all three parts of YOLOX (Backbone, Neck, Head). The following is our detailed description of all the improved methods. Section 3.2 describes our improvement in the backbone network, mainly by introducing a more efficient space pyramid pooling module; Section 3.3 describes our improvement in the neck network, including a new neck network and a new upsampling paradigm; Section 3.4 describes our improvement in the detection head, mainly by using a new loss function.

3.2. Improvement in the backbone

The back of YOLOX-Nano's backbone uses a Spatial Pyramid Pooling (SPP) module [35]. The module unifies the size of the input feature maps through a special pooling operation. YOLOX-Nano uses the SPP module to fuse features; its structure is shown in Figure 3(a). Three pooling layers with different sensory fields process the feature maps separately, then fuse their outputs to combine local and global features. The layers enhance the expression of the feature maps and effectively improve the model's accuracy.

Both classification and detection models can achieve good results using the SPP module. In subsequent studies, researchers have proposed many new pyramid pooling methods [12,36,37]. SPPF (SPP-Fast) is selected to replace the SPP module to improve this paper's detection accuracy and speed. Its structure is shown in Figure 3(b). The SPPF replaces the parallel pooling operation in the SPP with a serial operation. It all uses pooling kernels of 5 × 5 pooling layers; two serial 5 × 5 pooling layers are equivalent to a 9 × 9 pooling layer, and three serial 5 × 5 pooling layers are equivalent to a 13 × 13 pooling layer. Serial operation has the same effect as parallel operation but with higher efficiency. Therefore, this paper uses SPPF to improve the efficiency of feature fusion.



(a) SPP (b) SPPF

Figure 3. Structures of two pyramid pooling modules.

3.3. Improvements in the neck

In the neck network, this paper first uses a new convolutional structure, GSConv [38], to construct a slim-neck to replace the original neck network. Second, a new paradigm is designed in the upsampling part. The two improvements are our main innovations, and the following is a detailed introduction to these two improvements.

3.3.1. Slim-neck built by GSConv

GSConv is a new convolutional structure proposed by H. Li et al. [38] in June 2022. GSConv is essentially a fusion of regular convolution and DWC, and the operations of these two convolutions are shown in Figures 4 and 5, respectively.

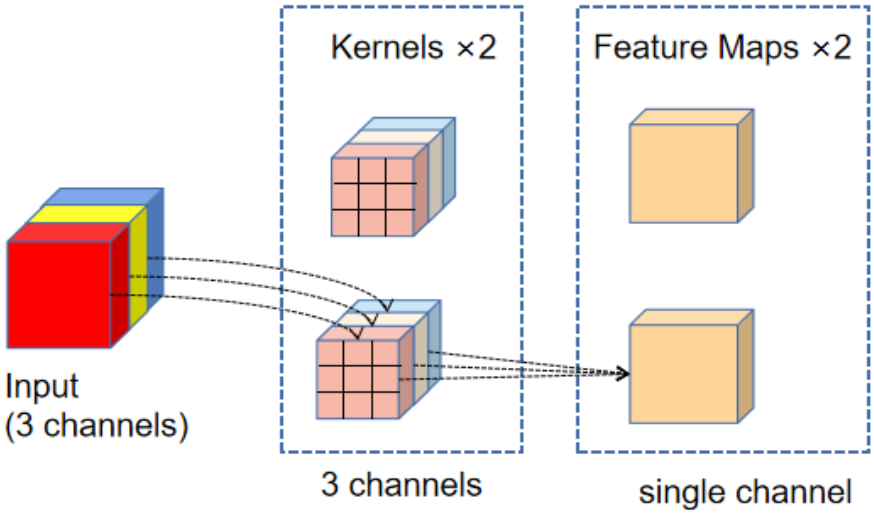


Figure 4. The operation process of 3 × 3 regular convolution.

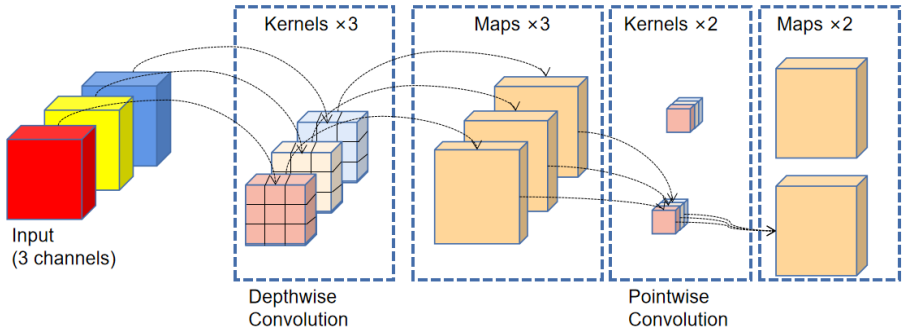


Figure 5. The operation process of depthwise separable convolution.

DWC is a lighter version of conventional convolution and includes two steps: depthwise convolution and pointwise convolution. First, each channel of the input image is assigned a single-channel filter (convolution kernel). The filter calculates all channels to obtain the same number of channels as the feature map, which is the operation process of depthwise convolution. Then, the 1 × 1 convolution makes regular convolution on the feature map, which is equivalent to summing up the weighted values of each channel to obtain a new feature map. This is the operation process of pointwise convolution. The DWC has 1/3 of the computational parameters of conventional convolution and greatly reduces the model's parameters and computation. Hence, it is widely used in various lightweight networks to speed up the model's training and inference. However, the model using depthwise separable convolution also has an obvious disadvantage: its accuracy is much lower than that of the same model structure using conventional convolution.

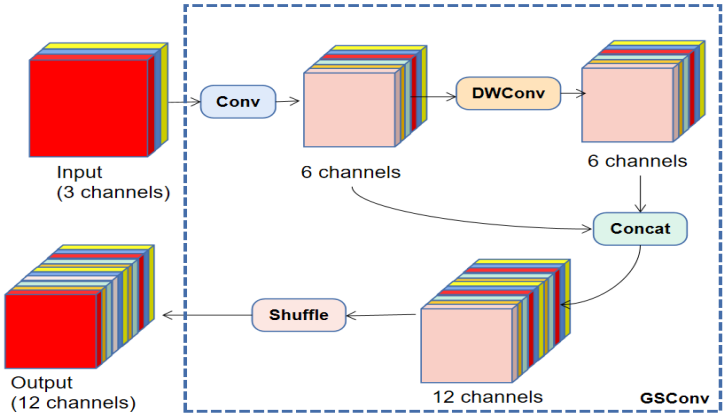
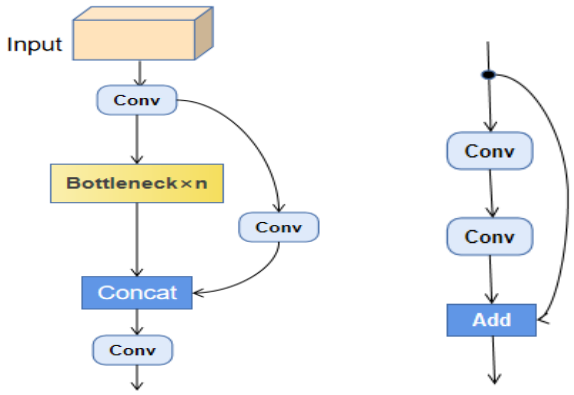


Figure 6. GSConv convolution process.

The compression operation and channel expansion of convolution cause the loss of feature information. DWC reduces the computation but cuts off the hidden connection between each channel by a 1 × 1 convolution operation. Conventional convolution can retain the relationship between channels, so it has higher accuracy. The GSConv is proposed to balance the model's speed and accuracy. The computation process of GSConv is shown in Figure 6. It combines both kinds of convolution and splices the results of conventional convolution and DWC to expand channels,

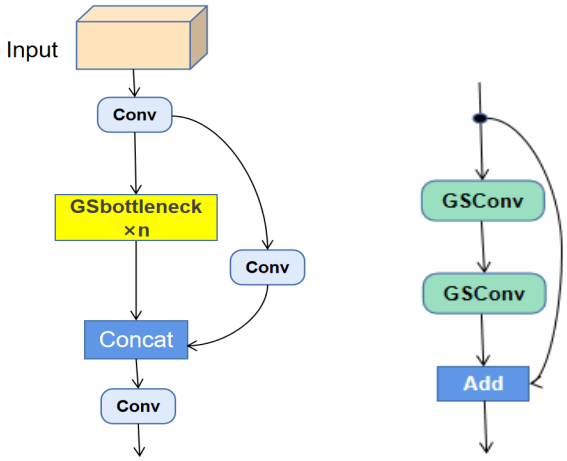
instead of the expansion by 1 × 1 convolution alone. Therefore, GSConv preserves the correlation between channels better than the DWC and reduces computation.

Slim-neck [38] is a neck network structure constructed by GSConv. Its overall structure design consults the neck network in YOLOv4 [11]. The neck network of YOLOv4 used the CSP (Cross Stage Partial) [25] modules. CSP is composed of multiple bottleneck modules, and the structure of CSP and bottleneck is shown in Figure 7.



(a) complete structure of CSP (b) bottleneck structure in CSP

Figure 7. Structural diagram of CSP.



(a) structure of VoV-GSCSP (b) structure of GSbottleneck

Figure 8. Structural diagram of VoV-GSCSP.

H. Li et al. imitated the structure of CSP and constructed VoV-GSCSP using GSConv [38]. VoV-GSCSP is composed of GSbottleneck, the structure of VoV-GSCSP and GSbottleneck is shown in Figure 8. As we can see, the difference between VoV-GSCSP and CSP is that they use different convolution structures, the former uses GSConv and the latter only uses ordinary convolution. VoV-GSCSP has fewer parameters and faster inference speed than the CSP structure constructed by

conventional convolution. It has higher accuracy than the CSP structure built using the DWC. GSConv combines the high accuracy of conventional convolution with the low computation of DWC, and it is a convolutional structure that balances accuracy and speed. However, if GSConv is used to build the whole model, it will deepen the network's layers and exacerbate the resistance to the data flow. Therefore, the final result is inferior to that of a model composed of conventional convolution, so GSConv is generally only used in the neck network. When the image data passes through the backbone network and reaches the neck network, the feature map has become slender (the channel dimension reaches the maximum and the width and height dimension reaches the minimum), and no further transformations are required. Therefore, by applying VoV-GSCSP constructed using GSConv to the neck network, the model can reduce parameters as much as possible while retaining the model's accuracy and inference speed.

Based on the above discussion, we replace the neck network of YOLOX-Nano with the slim-neck built by the GSConv. The slim-neck structure is shown in Figure 9. Compared with DWC, GSConv retains the accuracy of convolution operation better and is lightweight enough. Although the GSConv model is slightly larger than the DWC model, its accuracy is much higher than that of the DWC. Therefore, the GSConv is undoubtedly a better lightweight convolution structure than the DWC.

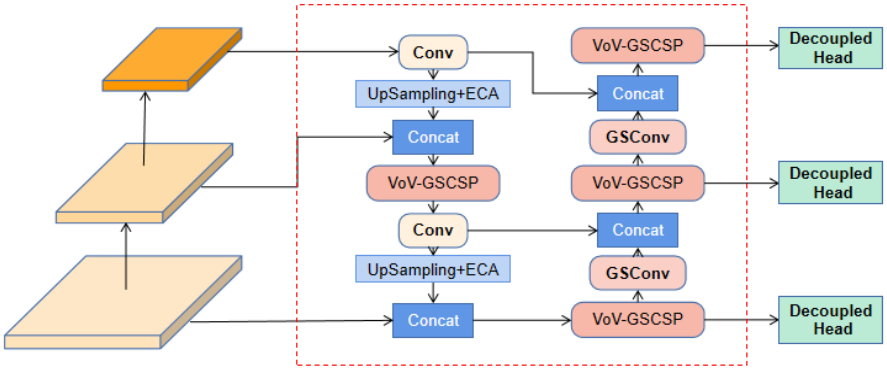


Figure 9. The slim-neck structure that we constructed.

This paper further improves the model's accuracy by replacing all DWC in the original neck network with GSConv. In addition, as shown in the upsampled section of Figure 9, this paper combines linear interpolation and attention mechanisms. This new approach is explained in detail in the following subsection.

3.3.2. A new upsampling paradigm

This paper proposes a new sampling paradigm: linear interpolation+attention mechanism to improve the model's ability to process remote sensing image features.

While most of the neural network model parameters are learned during training, the upsampling algorithm (linear interpolation) used in the neck network for feature fusion does not require adaptive learning of parameters. We envisioned that if the upsampling process also undergoes parameter learning, it may be possible to improve accuracy. In addition to linear interpolation, generic upsampling methods include transposed convolution [39], dilated convolution [40] and other

convolution algorithms that can expand the image size. Although these convolution methods enable parameter learning in upsampling, they also increase the model's parameters and computation because of adding convolution. We believe these convolution methods are not suitable for lightweight models. Therefore, we creatively combine linear interpolation and a lightweight attention module to propose a new paradigm of upsampling.

Our method expanded the feature map by linear interpolation and then adaptively adjusted the weight of each pixel in the feature map by parameter learning of attention. By comparing the experimental results of multiple lightweight attention mechanisms, this paper finally chooses the Efficient Channel Attention (ECA) [41] mechanism to combine with upsampling. The structure of the ECA module is shown in Figure 10.

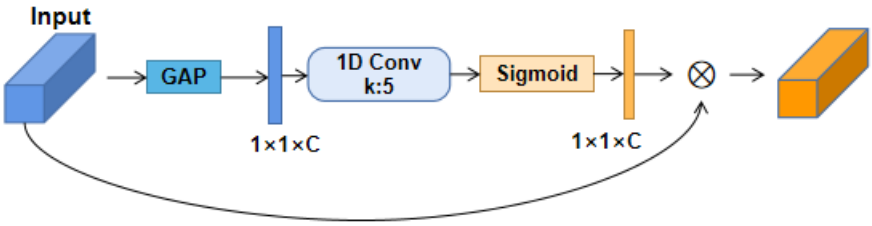


Figure 10. Structure of the ECA module.

First, a global-level pooling computes the feature map inputted to the ECA module to obtain a one-dimensional matrix. Then, a one-dimensional convolution is used for the operation. C denotes the number of channels, while k denotes the size of the convolution kernel of the one-dimensional convolution, which also represents the k-nearest neighbors of each channel; k is generally set to 5 for the best effect.

The parameters in the one-dimensional convolution are obtained by adaptive learning of the network. After processing by the one-dimensional convolution and the activation function (sigmoid), we get the attention map of the ECA module. The attention map is then dot-multiplied with the upsampled feature map to strengthen the critical information in the feature map, so that the network can also learn the upsampled image features.

The one-dimensional convolution used by ECA contains only small parameters, so it is a very lightweight and effective attention mechanism. This paper combines it with the upsampling operations to improve the model's accuracy without increasing its parameters or computation.

3.4. Improvement in the head

The localization loss function used by the YOLOX is an intersection over union (IoU) function [42]. The IoU is calculated by the intersection ratio between the prediction box and the ground truth box. The loss function $Loss_{IoU}$ used by the YOLOX is as follows:

$$Loss_{IoU} = 1 - IoU^2 \tag{1}$$

where $IoU = \frac{B \cap G}{B \cup G}$. B and G denote the prediction box and the ground truth box, respectively.

The standard IoU loss function does not have the squared term. The squared term in the

YOLOX came from the Alpha-IoU proposed by Jiabo et al. [43]. They proved that using a squared term for the IoU loss functions had better detection results and was suitable for small targets. After that, many studies added penalty terms to the IoU to build new loss functions. For example, GIoU [44] added a penalty term for the minimum rectangle size enclosing two boxes; DIoU [45] added a penalty term for the distance between the centroids of two boxes; CIoU [45] added a penalty term for the aspect ratio on top of this. Recently, Zhora has proposed a new localization loss function SIoU [46], which consists of four loss terms: angle, distance, shape and intersection ratio.

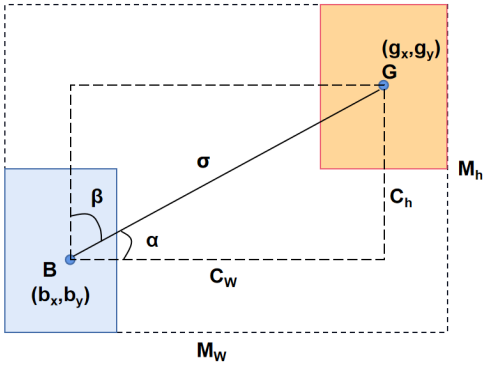


Figure 11. Diagrammatic representation of the SIoU calculation equation; (b_x, b_y) and (g_x, g_y) are the coordinates of the centroids of the predicted and real boxes, respectively, σ is the distance between the two centroids, C_h is the height difference between the centroids, C_w is the horizontal distance between the centroids and M_w and M_h are the minimum width and height of the outer rectangle, respectively.

Using the data in Figure 11, the angular loss Λ for SIoU can be calculated as shown in the following equations:

$$\Lambda = 1 - 2 \times sin^2\left(arcsin\left(\frac{C_h}{\sigma}\right) - \frac{\pi}{4}\right) = cos\left(2 \times \left(arcsin\left(\frac{C_h}{\sigma}\right) - \frac{\pi}{4}\right)\right) \tag{2}$$

$$\frac{C_h}{\sigma} = sin(\alpha) \tag{3}$$

$$\sigma = \sqrt{(g_x - b_x)^2 + (g_y - b_y)^2} \tag{4}$$

$$C_h = max(g_y, b_y) - min(g_y, b_y) \tag{5}$$

where, σ is the distance between the two centroids, C_h is the height difference between the centroids, Λ in Eq (2) represents the degree of angular deviation of the two centroids, that is, the angle loss value.

The distance loss Δ for SIoU can then be calculated as shown in the following equations:

$$\Delta = \sum_{t=x,y}\left(1 - e^{-\gamma\rho_t}\right) = 2 - e^{-\gamma\rho_x} - e^{-\gamma\rho_y} \tag{6}$$

$$\rho_x = \left(\frac{g_x - b_x}{M_w}\right)^2 \tag{7}$$

$$\rho_y = (\frac{g_y - b_y}{M_h})^2 \tag{8}$$

$$\gamma = 2 - \Lambda \tag{9}$$

where, M_w and M_h are the width and height of the minimum outer rectangle, respectively. As seen from Eq (6), angle loss Λ is used to calculate distance loss Δ. When α tends to be 0, the contribution of distance loss is reduced greatly. Conversely, when α is closer to $\Pi/4$, the contribution of distance loss is greater.

To calculate the shape loss Ω, we need to determine the width and height of the prediction and ground truth boxes. This paper sets B_w, B_h, G_w, G_h to be the width and height of the prediction and ground truth boxes, respectively. The calculation process is as follows:

$$\Omega = \sum_{(t=w,h)}(1 - e^{(-w_t)})^\theta = (1 - e^{(-w_w)})^\theta - (1 - e^{(-w_h)})^\theta \tag{10}$$

$$W_w = \frac{|B_W - G_W|}{max(B_W, G_W)} \tag{11}$$

$$W_h = \frac{|B_h - G_h|}{max(B_h, G_h)} \tag{12}$$

where θ is used to control the degree of attention to shape; to avoid focusing too much on shape and reducing the prediction accuracy, θ is often used by setting it to 4. The final SIoU loss is calculated as follows:

$$Loss_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{13}$$

Compared with the previous IoU series loss functions, SIoU considers the vector angle problem in the regression process, redefines the calculation of penalty terms and accelerates the model's convergence and inference. The positioning loss function can improve the model's accuracy and speed without increasing parameters.

4. Experimentation and discussion

4.1. Datasets

Four datasets were tested to demonstrate the generalization performance of the proposed model. They are NWPU VHR-10 [47], RSOD [48], TGRS-HRRSD [49] and DOTA [59]. The following is a brief description of these four datasets.

NWPU VHR-10 is a well-known geospatial object detection dataset that is also commonly used in remote sensing target detection. It includes 650 positive images with objects and 150 negative images without objects. The dataset has 3745 object instances in all 800 images. The images have a spatial resolution size of 0.5–2 m, and there are ten categories: airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge and vehicle. The dataset contains objects of various sizes and it was mainly used to quickly test the roles of various modules to find the best network architecture efficiently. We randomly selected 480 images as the training

set, 160 as the validation set and 160 as the test set.

RSOD includes 936 images with 6950 object instances and has four categories: aircraft, playground, overpass and oil tank. Its spatial resolution is 0.8–1 m. We randomly selected 561 images as the training set, 188 as the validation set and 187 as the test set.

TGRS-HRRSD is a large dataset of high-resolution remote sensing images for target detection, comprising 21,761 images with 55,740 object instances. It has 13 categories: ship, bridge, ground track field, storage tank, basketball court, tennis court, airplane, baseball diamond, harbor, vehicle, crossroad, T junction and parking lot. The images in this dataset have a spatial resolution of 0.15–1.2 m. The greatest advantage of this dataset is that the number of categories is balanced, with around 4000 instances per category. We used the officially divided dataset directly. The training set has 5401 images, the verification set has 5417 images and the test set has 10,943 images.

DOTA includes 2806 large-size images with 403,318 object instances. It has 16 categories: plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, swimming pool and container crane. The images in this dataset have a spatial resolution of 0.8–20 m. It contains many images of enormous size, which will cause a display memory explosion if it is directly used for model training and detection. Therefore, the traditional approach is to cut the images before using this dataset. So we cropped each image to a size of 640 × 640. The number of images after cropping is 21,310. We split the images into a training set, a validation set and a test set in a ratio of 6:2:2. The training set has 12,786 images, the validation set has 4262 images and the test set has 4262 images.

4.2. Evaluation indicators

This study used four metrics to evaluate the detection model's performance: mAP, latency, FPS and parameters; these metrics are described in detail below.

Precision and recall are calculated as follows:

$$Precision = \frac{TP}{TP+FP} \tag{14}$$

$$Recall = \frac{TP}{TP+FN} \tag{15}$$

where, TP is the number of samples that were positive and also correctly classified as positive, TN is the number of samples that were negative and also correctly classified as negative, FP is the number of samples that were negative but incorrectly classified as positive, and FN is the number of samples that were positive but classified as negative.

AP denotes the average precision of a single class of objects, combining both the precision and recall metrics. Using recall as the horizontal coordinate and precision as the vertical coordinate, a P-R curve can be drawn. The AP value of this object's class is equal to the area under that curve. The formula for calculating the AP value is as follows:

$$AP = \int_0^1 P(r)dr \tag{16}$$

where P indicates precision and r indicates recall. After calculating the AP value of each category, we can calculate mAP, which is equal to the average accuracy of all categories, as follows:

$$mAP = \frac{\sum_{k=1}^{n} AP^k}{n} \tag{17}$$

When using AP and mAP, this paper uses subscripts to indicate the threshold value of IoU (positive sample when the IoU of the predicted box is greater than the threshold value with the real box). For example, AP_{50} denotes the AP value when the IoU threshold is 0.50; $mAP_{50\text{-}95}$ denotes the mAP value when the thresholds are 0.50, 0.55, 0.60, ..., 0.90 and 0.95. In this paper, we use $mAP_{50\text{-}95}$ to measure the model's accuracy.

Latency denotes the average time for the model to process an image, FPS denotes the number of image frames detected per second and parameter denotes the total number of parameters of the model. In the subsequent experimental sessions, the letter P is used to denote the number of parameters in this paper. We use latency and FPS to measure the detection speed of the model and P to measure the model's size.

4.3. Experimental environment

We run all experiments in this paper with PyTorch 1.10.0, CUDA 10.2 and Python 3.8. Our machine had an NVIDIA GeForce RTX 2080Ti graphics processing unit (GPU) and an AMD Ryzen 5 3600X 6-Core CPU processor. All experiments were performed with FP16 and set batch size = 1.

4.4. Effects of spatial pyramid pooling

This work tests all existing spatial pyramid pool structures [12,35,36,37] in the NWPU VHR-10 dataset using the YOLOX-Nano model. Our results are shown in Table 1. The experimental results in Table 1 show that ASPP [36] and SPPCSPC [37] significantly improve the model accuracy. However, they increase the number of parameters of the model, which is not conducive to the lightweight model. SPPF and SimSPPF do not increase the number of parameters; they change the parallel structure in SPP to a serial structure, which speeds up model inference. SimSPPF replaces the SiLU activation function used in SPPF with the ReLU activation function, which is the fastest spatial pyramid pooling module but has slightly reduced accuracy. We chose to use SPPF as the pyramid pooling module in YOLOX-Nano through an overall consideration, as it scarcely increases the number of the model's parameters while also improving the model's performance.

After replacing SPP with SPPF in YOLOX-Nano, we tested the method on all four datasets to verify the method's generalization ability and obtained the results shown in Table 2.

Table 1. Test results of various SPP modules in YOLOX-Nano with the NWPU VHR-10 dataset.

Methods	mAP_{50} (%)	$mAP_{50\text{-}95}$ (%)	P (M)	Latency (ms)	FPS
YOLOX+SPP	86.16	52.55	0.90	19.77	50.58
YOLOX+SPPF	86.63	53.73	0.90	17.91	55.83
YOLOX+SimSPPF	84.41	52.61	0.90	**17.55**	**56.98**
YOLOX+ASPP	**86.78**	53.86	2.97	22.36	44.72
YOLOX+SPPCSPC	86.18	**54.36**	2.51	20.11	49.72

Table 2. Effectiveness of SPPF on four aviation datasets.

Methods	NWPU VHR-10		RSOD		TGRS-HRRSD		DOTA	
	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS
YOLOX (SPP)	52.55	50.58	56.62	51.57	57.92	87.79	55.48	93.02
SPPF	53.73	55.83	57.39	55.46	58.79	89.77	57.01	95.37

4.5. Effects of slim-neck formed by GSConv

This paper tests various combinations of GSConv and neck networks in YOLOX-Nano to balance the model's accuracy, speed and parameter number, and the experimental results are shown in Table 3. By comparing the model performance of the three combinations, we finally chose the program III, replacing all of the DWC in the neck structure of YOLOX-Nano with GSConv. Our slim-neck structure is shown in Figure 8.

After fusing the neck structure with the YOLOX model, the test results on the four datasets are shown in Table 4. GSConv is closer to the standard convolution than DWC in accuracy, and it can somewhat mitigate the slim-neck structure's accuracy loss. Compared to DWC, the slim-neck structure formed by GSConv slightly reduces the model's inference speed, but the accuracy is improved and the speed reduction is within our acceptable range. Our results show that GSConv maintains the accuracy of standard convolution better than DWC.

Table 3. Testing of the slim-neck structures for different configurations in NWPU VHR-10.

Methods	mAP_{50} (%)	mAP_{50-95} (%)	P (M)	Latency (ms)	FPS
YOLOX	86.16	52.55	0.90	19.77	50.58
Program I	79.16	46.68	0.95	22.10	45.25
Program II	82.56	49.59	0.77	**19.08**	**52.41**
Program III	**87.88**	**53.61**	1.06	20.23	49.43

Note: The program I replaces all of the convolutional structures in the neck network with GSConv; the program II replaces all of the standard convolutions in the neck with GSConv and keeps the DWC; the program III replaces all of the DWC in the neck with GSConv and maintains the standard convolutions.

Table 4. Effectiveness of slim-neck structure on the four aerial datasets.

Methods	NWPU VHR-10		RSOD		TGRS-HRRSD		DOTA	
	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS
YOLOX	52.55	50.58	56.62	51.57	57.92	87.79	55.48	93.02
+slim-neck	53.61	49.43	58.23	50.82	58.59	85.68	56.87	92.64

4.6. Effects of linear interpolation+attention mechanism

YOLOX uses nearest-neighbor interpolation to achieve upsampling. Each pixel value in the expanded feature map is calculated based on its nearest pixel in the original feature map. The nearest-neighbor interpolation is not needed to learn the parameters, which reduces the computational cost but is not conducive to the adaptive learning of neural networks. Therefore, we add attention mechanisms after linear interpolation for the network to adaptively learn the critical

information that should be enhanced after upsampling. To avoid increasing the number of parameters in the model, this paper compares several currently popular attention mechanisms for lightweight structures combined with linear interpolation to find the most suitable attention algorithm. The experimental results are shown in Table 5.

The attention mechanisms that we chose are all extremely lightweight. The experimental results show that these attention mechanisms bring almost no additional parameters after being added to the YOLOX model. The ECA module achieves the best result among all of these lightweight attention modules. Thus, we finally chose the ECA combined with linear interpolation to complete the upsampling process. The experimental results of the method on all four aerial datasets are shown in Table 6.

Table 5. Effect of different attention mechanisms combined with linear interpolation.

Methods	mAP_{50} (%)	mAP_{50-95} (%)	P(M)	Latency (ms)	FPS
Interpolation	86.16	52.55	0.90	**19.77**	**50.58**
+SimAM [50]	86.11	53.05	0.90	20.26	49.35
+SA [51]	87.02	52.87	0.90	21.18	47.21
+ULSAM [52]	85.28	52.14	0.90	24.17	41.37
+NAM [53]	87.26	53.71	0.90	20.87	47.91
+ECA [41]	**87.35**	**54.69**	0.90	20.34	49.16

Note: The dataset was NWPU VHR-10, and the interpolation in the table represents the nearest neighbor interpolation algorithm.

Table 6. Effectiveness of linear interpolation+ECA on the three aerial datasets.

Methods	NWPU VHR-10		RSOD		TGRS-HRRSD		DOTA	
	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS
YOLOX	52.55	50.58	56.62	51.57	57.92	87.79	55.48	93.02
+ECA	54.69	49.16	58.24	50.58	59.24	85.81	58.23	92.17

4.7. Study of the localization loss function

YOLOX-Nano uses the IoU loss as the localization loss function. This work tries other more advanced loss functions to improve the model's accuracy, which are obtained by adding a new penalty on the IoU. Table 7 shows our results of testing these loss functions in the YOLOX. To ensure fairer experimental results, this work squares the penalty terms of the locus loss functions, referring to the YOLOX.

The experimental results show the SIoU undoubtedly has the best speed and accuracy, and it is the best localization loss function for our real-time detection. It can speed up training and inference by redefining the penalty calculation. It adds the angle calculation to make the prediction box fit the ground truth box faster towards the correct direction. The results of testing SIoU using four aerial datasets are shown in Table 8.

Table 7. Test results for different localization loss functions in the YOLOX-Nano model.

Methods	mAP_{50} (%)	mAP_{50-95} (%)	P (M)	Latency (ms)	FPS
IoU	86.16	52.55	0.90	19.77	50.58
GIoU	86.36	52.67	0.90	19.86	50.35
DIoU	86.22	54.08	0.90	20.03	49.92
CIoU	85.05	53.30	0.90	20.67	48.37
SIoU	**87.71**	**54.14**	0.90	**18.01**	**55.52**

Table 8. Effectiveness of SIoU on the three aviation datasets.

Methods	NWPU VHR-10		RSOD		TGRS-HRRSD		DOTA	
	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS	mAP_{50-95} (%)	FPS
YOLOX (IoU)	52.55	50.58	56.62	51.57	57.92	87.79	55.48	93.02
SIoU	54.14	55.52	58.10	54.73	59.78	89.05	58.12	96.64

4.8. Ablation experiments and comparison experiments

We add the above improvements to the YOLOX-Nano model, trained and tested it using the NWPU VHR-10 dataset, and obtained the experimental results shown in Table 9. Our improved YOLOX-Nano model upgraded mAP_{50} on the aerial dataset NWPU VHR-10 by 2.31% and mAP_{50-95} by 3.13% compared to the original model. The FPS also increased by 2.32. However, our model only increased the number of parameters by 0.16 M. The results demonstrate that our model better balances speed and accuracy.

Table 9. Results of the ablation experiments on the NWPU VHR-10.

SPPF	Slim -Neck	ECA	SIoU	mAP_{50} (%)	mAP_{50-95} (%)	P(M)	Latency (ms)	FPS
				86.16	52.55	0.90	19.77	50.58
√				86.63 (+0.47)	53.73 (+1.18)	0.90	17.91 (-1.86)	55.83 (+5.25)
√	√			87.82 (+1.19)	54.41 (+0.68)	1.06 (+0.16)	19.32 (+1.41)	51.75 (-4.08)
√	√	√		88.21 (+0.39)	55.12 (+0.71)	1.06	19.79 (+0.47)	50.53 (-1.22)
√	√	√	√	88.47 (+0.26)	55.68 (+0.56)	1.06	18.92 (-0.87)	52.85 (+2.32)

To validate our model's performance, this paper compares it with the currently popular lightweight models on the NWPU VHR-10 dataset. The results are shown in Table 10. Our model has the highest detection accuracy of all the lightweight models shown in Table 10. The accuracy of YOLOv5n is the closest to ours, but its detection speed is far inferior to our model. The faster models are YOLO-Fastest, NanoDet and especially the FastestDet model. They have extremely lightweight structures and very fast detection speed but very low accuracy. It can be seen that our model has the best balance of speed and accuracy among all lightweight models.

In addition, to further verify the balance between the speed and accuracy of our model, this paper also compares it with the current popular conventional models, and the comparison results are

shown in Table 11. Conventional models use conventional convolution and do not use lightweight convolution structures. All models listed in Table 11 have been used for the detection of aerial remote sensing images, so the comparative experiments have reference value. As seen from Table 11, in the NWPU VHR-10 dataset, our model has an absolute advantage in speed compared with the conventional model, and its accuracy is comparable to these models.

Table 10. Test results of different lightweight models on the NWPU VHR-10 dataset.

Models	mAP_{50} (%)	mAP_{50-95} (%)	P (M)	Latency (ms)	FPS
YOLOv3-Tiny [10]	75.46	42.38	8.86	48.46	20.53
YOLOv4-Tiny [11]	80.27	47.65	6.06	42.98	23.27
YOLOv5n [12]	87.23	54.06	1.90	26.12	38.28
YOLO-Fastest [54]	65.18	31.79	0.35	10.28	97.27
FastestDet [55]	65.67	32.82	**0.24**	**9.32**	**107.29**
NanoDet [24]	80.12	46.23	0.95	14.69	68.07
YOLOX-Nano [13]	86.16	52.55	0.90	19.77	50.58
YOLOX-Nano++(ours)	**88.47**	**55.68**	1.06	18.92	52.85

Table 11. Comparison between our model and some conventional models on NWPU VHR-10.

Models	mAP_{50} (%)	mAP_{50-95} (%)	P (M)	Latency (ms)	FPS
Improved Faster R-CNN [27]	81.64	48.29	60.42	273.24	3.66
YOLOv4 [11]	86.57	53.29	64.02	26.54	37.68
YOLOv5m [12]	**92.61**	**61.02**	21.20	34.76	28.77
YOLOD [26]	88.13	56.69	70.12	30.19	33.12
RS-YOLOX [29]	90.21	58.34	14.38	27.83	35.93
YOLOX-Nano++(ours)	88.47	55.68	**1.06**	**18.92**	**52.85**

4.9. Display of object detection results

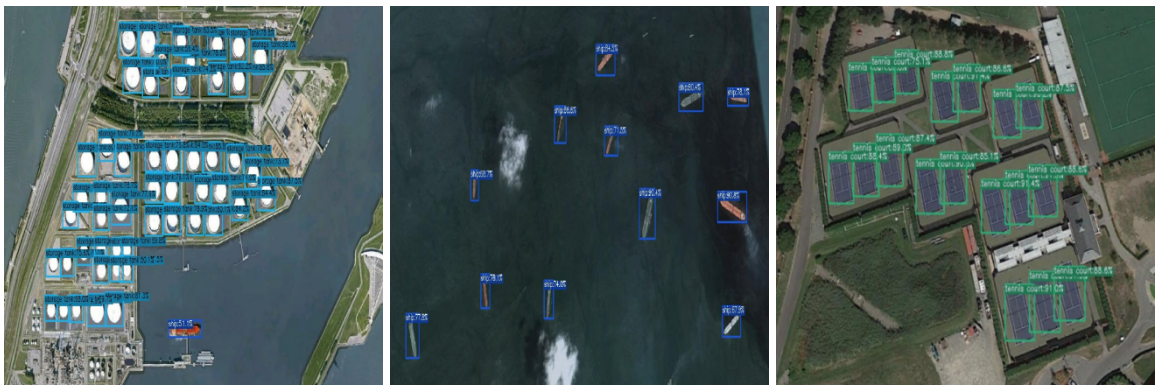


Figure 12. Detection results of our model on the NWPU VHR-10 dataset.

Figure 13. Detection results of our model on the RSOD dataset.

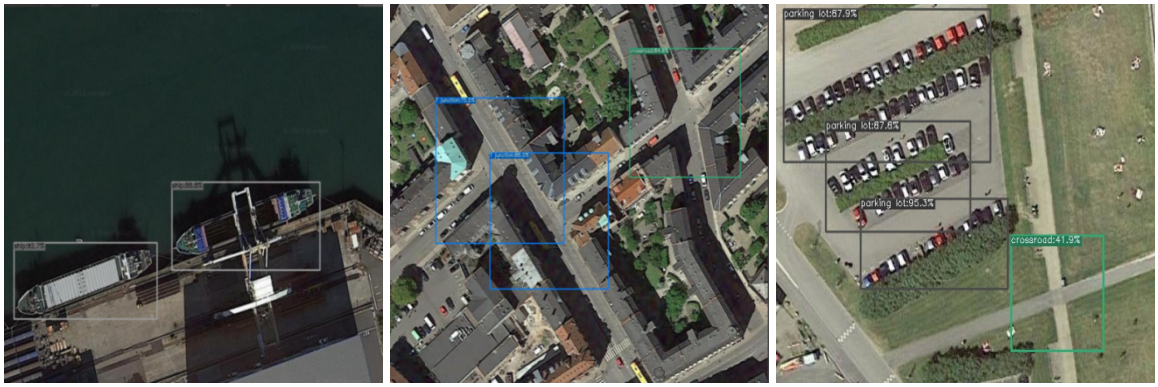
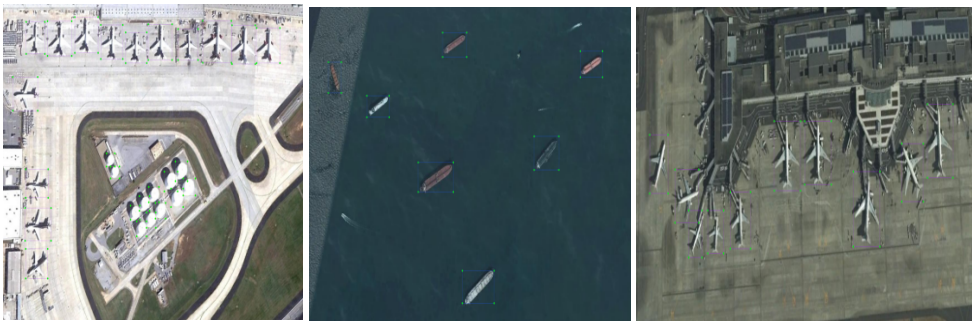


Figure 14. Detection results of our model on the TGRS-HRRSD dataset.

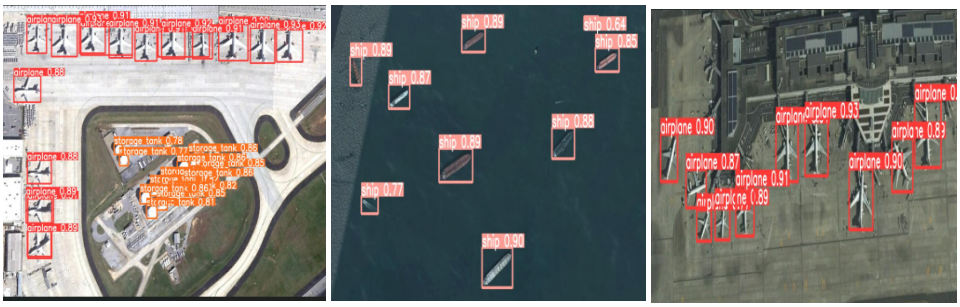
Figure 15. Detection results of our model on the DOTA dataset.

Our model can detect targets in remote sensing images quickly and accurately, and some detection results on the four aerial remote sensing image datasets are shown in Figures 12–15. In Figure 15, this paper uses large-size images in DOTA. The image is first cut into several smaller images (640 × 640) before detection. After detection, the results of small images are combined to get a complete detection image.



(a) Ground truth

(b) NanoDet



(c) YOLOv5m

(d) YOLOX-Nano++(ours)

Figure 16. Comparison of detection results of our model with other models.

Figure 16 shows the detection comparison between our model and other models on the NWPU VHR-10 dataset. Figure 16(a) shows the labeling of real target boxes (Ground truth). According to experimental data in Tables 10 and 11, NanoDet and YOLOv5m were selected as representatives of lightweight and conventional models, respectively. Figure 16(b)–(d) show the detection results of NanoDet, YOLOv5m and our model respectively. It can be observed that the NanoDet model missed the detected targets, while our model achieved the detection effect of the conventional model YOLOv5m. However, the detection accuracy of YOLOv5m is slightly higher than that of our model. The detection results of our model and YOLOv5m are very close to the true annotation results.

4.10. Discussion

This paper improves the YOLOX-Nano model using several lightweight improvement strategies to increase accuracy and speed, achieving a better balance of accuracy and speed.

Our two most significant innovations are the combination of YOLOX and slim-neck and the proposal of a new upsampling paradigm. In this paper, slim-neck constructed by GSConv is used, because DWC in the original model reduces the accuracy of the model. We need a lightweight neck while preserving the accuracy of conventional convolution to the maximum extent. GSConv just meets our demand. The new paradigm of upsampling is proposed because we consider that the upsampling process does not go through parameter learning. Therefore, this paper integrates the attention mechanism to enable the model to learn features adaptively during the upsampling process. Although these two methods slightly increased the number of parameters (increased by 0.16 M), they brought significant accuracy improvement. We have confirmed their superiority in target detection for aerial images.

In addition, our two minor innovations are the use of the SPPF module and the SIoU function. Their primary effect is to improve the model's detection speed. SPPF can accelerate the pyramid pooling process, and SIoU can accelerate the regression calculation process. These two methods not only improve the detection speed, but also slightly improve the detection accuracy of the model. When we added these improvements to the YOLOX model, both the accuracy and speed increased, so our model undoubtedly has a better accuracy and speed balance.

At present, some advanced rotating object detection models have been used in remote sensing images, such as R3Det [56], S2A-Net [57], Oriented R-CNN [58], etc. These models can adjust the prediction box's rotation angle according to the target's shape to better fit the target. Our model does not use the rotating object detection method, so rotation object detection is one of the future improvements in our model. In addition, our work can be extended to small object detection [60,61]. One of them is the new upsampling paradigm, a feature-strengthening method that can strengthen the features of small targets, to improve the detection accuracy of small targets. Therefore, extending our method to small target detection is also our future work.

5. Conclusions

The main objective of this paper is to achieve fast and highly accurate detection of aerial remote sensing targets using a new lightweight model. We have improved the YOLOX-Nano model, keeping it lightweight while improving its accuracy and speed. We replaced the SPP module in YOLOX with a more efficient SPPF module. Then, we reconstructed the neck network with a new convolutional

structure (GSConv), constructed a lightweight slim-neck structure and in the upsampling process, we proposed a new upsampling paradigm and introduced a lightweight attention mechanism ECA. Finally, we replaced the localization loss function of YOLOX with SIoU, while improving the accuracy and speed of the model.

All the improvement strategies in this paper are to enhance the detection speed and accuracy of the model as much as possible on the premise of keeping the model lightweight. We obtained a final improved model with superior accuracy and speed, with only a 0.16 M increase in the number of parameters. Our model balances speed and accuracy, which is an excellent model for the real-time detection of remotely sensed targets. This model can be mounted on a UAV to detect ground targets quickly and accurately. We will add a rotating detection box to fit the target position in future work better. In addition, we will extend our work to small target detection and explore the fusion of our approach and small target.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the Key R&D Projects in the Yunnan Province (Grant No. 202202AD080004), the Natural Science Foundation of China (Grant Nos. 62061049, 12263008), the Application and Foundation Project of the Yunnan Province (Grant No. 202001BB050032) and the Yunnan Provincial Department of Science and Technology-Yunnan University Joint Special Project for Double-Class Construction (Grant No. 202201BF070001-005).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. M. Lu, Y. Xu, H. Li, Vehicle Re-Identification based on UAV viewpoint: dataset and method, *Remote Sens.*, **14** (2022), 4630. https://doi.org/10.3390/rs14184603
2. S. Ijlil, A. Essahlaoui, M. Mohajane, N. Essahlaoui, E. M. Mili, A. V. Rompaey, Machine learning algorithms for modeling and mapping of groundwater pollution risk: A study to reach water security and sustainable development (Sdg) goals in a editerranean aquifer system, *Remote Sens.*, **14** (2022), 2379. https://doi.org/10.3390/rs14102379
3. Z. Jiang, Z. Song, Y. Bai, X. He, S. Yu, S. Zhang, et al., Remote sensing of global sea surface pH based on massive underway data and machine mearning, *Remote Sens.*, **14** (2022), 2366. https://doi.org/10.3390/rs14102366
4. Y. Zhao, L. Ge, H. Xie, G. Bai, Z. Zhang, Q. Wei, et al., ASTF: Visual abstractions of time-varying patterns in radio signals, *IEEE Trans. Visual Comput. Graphics*, **29** (2023), 214–224. https://doi.org/10.1109/TVCG.2022.3209469

5. R. Girshick, J. Donahue, T. Darrell J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81
6. R. Girshick, Fast R-CNN, in 2*015 IEEE International Conference on Computer Vision (ICCV)*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169
7. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
8. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
9. J. Redmon, A. Farhadi, YOLO9000: Better, Faster, Stronger, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6517–6525. https://doi.org/10.1109/CVPR.2017.690
10. J. Redmon, A. Farhadi, YOLOv3: an incremental improvement, *arXiv preprint*, (2018), arXiv:1804.02767. http://arxiv.org/abs/1804.02767
11. A. Bochkovskiy, C. Y. Wang, H. Liao, YOLOv4: optimal speed and accuracy of object detection, *arXiv preprint*, (2020), arXiv:2004.10934. http://arxiv.org/abs/2004.10934
12. G. Jocher, Yolov5, 2020. Available from: https://github.com/ultralytics/yolov5.
13. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding YOLO series in 2021, *arXiv preprint*, (2021), arXiv:2107.08430. https://arxiv.org/abs/2107.08430
14. Y. Li, X. Liu, H. Zhang, X. Li, X. Sun, Optical remote sensing image retrieval based on convolutional neural networks (in Chinese), *Opt. Precis. Eng.*, **26** (2018), 200–207. https://doi.org/10.3788/ope.20182601.0200
15. A. Van Etten, You only look twice: Rapid multi-scale object detection in satellite imagery, *arXiv preprint*, (2018), arXiv:1805.09512. https://doi.org/10.48550/arXiv.1805.09512
16. M. Ahmed, Y. Wang, A. Maher, X. Bai, Fused RetinaNet for small target detection in aerial images, *Int. J. Remote Sens.*, **43** (2022), 2813–2836. https://doi.org/10.1080/01431161.2022.2071115
17. H. Liu, G. Yuan, L. Yang, K. Liu, H. Zhou, An appearance defect detection method for cigarettes based on C-CenterNet, *Electronics*, **11** (2022), 2182. https://doi.org/10.3390/electronics11142182
18. S. Du, B. Zhang, P. Zhang, P. Xiang, H. Xue, FA-YOLO: An improved YOLO model for infrared occlusion object detection under confusing background, *Wireless Commun. Mobile Comput.*, **2021** (2021). https://doi.org/10.1155/2021/1896029
19. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: Efficient convolutional neural networks for mobile vision applications, *arXiv preprint*, (2017), arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861
20. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
21. A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M. Tan, et al., Searching for mobileNetV3, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 1314–1324. https://doi.org/10.1109/ICCV.2019.00140

22. X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An extremely efficient convolutional neural network for mobile devices, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 6848–6856.
23. N. Ma, X. Zhang, H. T. Zheng, J. Sun, ShuffleNet V2: Practical guidelines for efficient CNN architecture design, in *European Conference on Computer Vision (ECCV)*, (2018), 122–138. https://doi.org/10.1109/CVPR.2018.00716
24. RangiLyu, NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model, 2021. Available from: https://github.com/RangiLyu/nanodet.
25. C. Y. Wang, H. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, I. H. Yeh, CSPNet: A bew backbone that can enhance learning capability of CNN, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (2020), 1571–1580. https://doi.org/10.1109/CVPRW50498.2020.00203
26. X. Luo, Y. Wu, L. Zhao, YOLOD: A target detection method for UAV aerial imagery, *Remote Sens.*, **14** (2022), 3240. https://doi.org/10.3390/rs14143240
27. D. Yan, G. Li, X. Li, H. Zhang, H. Lei, K. Lu, et al., An improved faster R-CNN method to detect tailings ponds from high-resolution remote sensing images, *Remote Sens.* **13** (2021), 2052. https://doi.org/10.3390/rs13112052
28. F. C. Akyon, S. O. Altinuc, A. Temizel, Slicing aided hyper inference and fine-tuning for small object detection, in *2022 IEEE International Conference on Image Processing (ICIP)*, (2022), 966–970. https://doi.org/10.1109/ICIP46576.2022.9897990
29. L. Yang, G. Yuan, H. Zhou, H. Liu, J. Chen, H. Wu, RS-YOLOX: A high-precision detector for object detection in satellite remote sensing images, *Appli. Sci.*, **12** (2022), 8707. https://doi.org/10.3390/app12178707
30. J. Liu, C. Liu, Y. Wu, Z. Sun, H. Xu, Insulators' identification and missing defect detection in aerial images based on cascaded YOLO models, *Comput. Intell. Neurosci.*, **2022** (2022). https://doi.org/10.1155/2022/7113765
31. X. Li, Y. Qin, F. Wang, F. Guo, J. T. W. Yeow, Pitaya detection in orchards using the MobileNet-YOLO model, in *2020 39th Chinese Control Conference (CCC)*, (2020), 6274–6278. https://doi.org/10.23919/CCC50068.2020.9189186
32. Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 9626–9635. https://doi.org/10.1109/ICCV.2019.00972
33. H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, *Int. J. Comput. Vision*, **128** (2020), 642–656. https://doi.org/10.1007/s11263-019-01204-1
34. G. Song, Y. Liu, X. Wang, Revisiting the sibling head in object detector, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 11563–11572.
35. K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2015), 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824
36. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *IEEE Trans. Pattern Anal. Mach. Intell.*, **40** (2018), 834–848. https://doi.org/10.1109/TPAMI.2017.2699184

37. C. Y. Wang, A. Bochkovskiy, H. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 7464–7475.
38. H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, Q. Ren, Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles, *arXiv preprint*, (2022), arXiv: 2206.02424. https://doi.org/10.48550/arXiv.2206.02424
39. V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning, arXiv preprint, (2018), arXiv:1603.07285. https://doi.org/10.48550/arXiv.1603.07285
40. F. Yu, V. Koltun, T. Funkhouser, Dilated residual networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 636–644. https://doi.org/10.1109/CVPR.2017.75
41. Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, Q. Hu, ECA-Net: Efficient channel attention for deep convolutional neural networks, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 11531–11539. https://doi.org/10.1109/CVPR42600.2020.01155
42. B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 784–799.
43. J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, X. S. Hua, Alpha-IoU: A family of power intersection over union losses for bounding box regression, in *NeurIPS 2021 Conference*, 2021.
44. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 658–666. https://doi.org/10.1109/CVPR.2019.00075
45. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU loss: Faster and better learning for bounding box regression, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. https://doi.org/10.1609/aaai.v34i07.6999
46. Z. Gevorgyan, SIoU loss: More powerful learning for bounding box regression, *arXiv preprint*, (2022), arXiv:2205.12740. https://doi.org/10.48550/arXiv.2205.12740
47. G. Cheng, P. Zhou, J. Han, Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images, *IEEE Trans. Geosci. Remote Sens.*, **54** (2016), 7405–7415. https://doi.org/10.1109/TGRS.2016.2601622
48. Y. Long, Y. Gong, Z. Xiao, Q. Liu, Accurate object localization in remote sensing images based on convolutional neural networks, *IEEE Trans. Geosci. Remote Sens.*, **55** (2017), 2486–2498. https://doi.org/10.1109/TGRS.2016.2645610
49. X. Lu, Y. Zhang, Y. Yuan, Y. Feng, Gated and axis-concentrated localization network for remote sensing object detection, *IEEE Trans. Geosci. Remote Sens.*, **58** (2020), 179–192. https://doi.org/10.1109/TGRS.2019.2935177
50. L. Yang, R. Y. Zhang, L. Li, X. Xie, SimAM: A simple, parameter-free attention module for convolutional neural networks, in *Proceedings of the 38th International Conference on Machine Learning*, **139** (2021), 11863–11874.
51. Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, et al., Squeeze-and-attention networks for semantic segmentatio, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 13065–13074.

52. R. Saini, N. K. Jha, B. Das, S. Mittal, C. K. Mohan, ULSAM: Ultra-lightweight subspace attention module for compact convolutional neural networks, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, (2020), 1616–1625. https://doi.org/10.1109/WACV45572.2020.9093341
53. Y. Liu, Z. Shao, Y. Teng, N. Hoffmann, NAM: Normalization-based attention module, *arXiv preprint*, (2021), arXiv:2111.12419. https://doi.org/10.48550/arXiv.2111.12419
54. X. Ma, Yolo-Fastest: yolo-fastest-v1.1.0, 2021. Available from: https://github.com/dog-qiuqiu/Yolo-Fastest.
55. X. Ma, FastestDet: Ultra lightweight anchor-free real-time object detection algorithm, 2022. Available from: https://github.com/dog-qiuqiu/FastestDet.
56. X. Yang, J. Yan, Z. Feng, T. He, R3Det: Refined single-stage detector with feature refinement for rotating object, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 3163–3173. https://doi.org/10.1609/aaai.v35i4.16426
57. J. Han, J. Ding, J. Li, G. S. Xia, Align deep features for oriented object detection, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 1–11. https://doi.org/10.1109/TGRS.2021.3062048
58. X. Xie, G. Cheng, J. Wang, X. Yao, J. Han, Oriented R-CNN for object detection, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 3500–3509. https://doi.org/10.1109/ICCV48922.2021.00350
59. J. Ding, N. Xue, Y. Long, G. S. Xia, Q. Lu, Learning RoI transformer for oriented object detection in aerial images, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 2844–2853. https://doi.org/10.1109/CVPR.2019.00296
60. S. Zhong, H. Zhou, Z. Ma, F. Zhang, J. Duan, Multiscale contrast enhancement method for small infrared target detection, *Optik*, **271** (2022), 170134. https://doi.org/10.1016/j.ijleo.2022.170134
61. S. Zhong, H. Zhou, X. Cui, X. Cao, F. Zhang, J. Duan, Infrared small target detection based on local-image construction and maximum correntropy, *Measurement*, **211** (2023), 112662. https://doi.org/10.1016/j.measurement.2023.112662

AIMS Press