Mathematical Biosciences and Engineering

MBE, 20(9): 16148–16168.
DOI: 10.3934/mbe.2023721
Received: 07 June 2023
Revised: 17 July 2023
Accepted: 27 July 2023
Published: 09 August 2023

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

Research article

Aerial images object detection method based on cross-scale multi-feature fusion

Yang Pan, Jinhua Yang*, Lei Zhu, Lina Yao and Bo Zhang

School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, China

* **Correspondence:** Email: 210411041@stu.xpu.edu.cn.

Abstract: Aerial image target detection technology has essential application value in navigation security, traffic control and environmental monitoring. Compared with natural scene images, the background of aerial images is more complex, and there are more small targets, which puts higher requirements on the detection accuracy and real-time performance of the algorithm. To further improve the detection accuracy of lightweight networks for small targets in aerial images, we propose a cross-scale multi-feature fusion target detection method (CMF-YOLOv5s) for aerial images. Based on the original YOLOv5s, a bidirectional cross-scale feature fusion sub-network (BsNet) is constructed, using a newly designed multi-scale fusion module (MFF) and cross-scale feature fusion strategy to enhance the algorithm's ability, that fuses multi-scale feature information and reduces the loss of small target feature information. To improve the problem of the high leakage detection rate of small targets in aerial images, we constructed a multi-scale detection head containing four outputs to improve the network's ability to perceive small targets. To enhance the network's recognition rate of small target samples, we improve the K-means algorithm by introducing a genetic algorithm to optimize the prediction frame size to generate anchor boxes more suitable for aerial images. The experimental results show that on the aerial image small target dataset VisDrone-2019, the proposed method can detect more small targets in aerial images with complex backgrounds. With a detection speed of 116 FPS, compared with the original algorithm, the detection accuracy metrics $\mathrm{mAP}_{0.5}$ and $\mathrm{mAP}_{0.5:0.95}$ for small targets are improved by 5.5% and 3.6%, respectively. Meanwhile, compared with eight advanced lightweight networks such as YOLOv7-Tiny and PP-PicoDet-s, $\mathrm{mAP}_{0.5}$ improves by more than 3.3%, and $\mathrm{mAP}_{0.5:0.95}$ improves by more than 1.9%.

Keywords: aerial images; object detection; YOLOv5s; cross-scale multi-feature fusion

1. Introduction

Object detection plays a vital role in many application scenarios, such as gesture recognition [1], instance segmentation [2] and medical data processing [3]. In recent years, with the rapid development of computer vision and airborne remote sensing technologies, target detection tasks based on aerial images have become a hotspot for research and play an essential role in scenarios such as nautical security [4], traffic management [5] and environmental monitoring [6]. Although much research has been conducted on target detection in aerial images, [7] summarizes in a very comprehensive and detailed way the current development status and the research progress made in aerial image target detection in recent years. However, due to the rapid change in flight altitude of aerial photography equipment, the unique nature of the shooting angle and location and having a complex background and more small targets, the detection algorithms' real-time detection and detection accuracy must also be improved.

Currently, there are two main types of target detection methods based on deep learning. One is the two-stage detection algorithm represented by Fast R-CNN [8]. One is the one-stage detection algorithm YOLO represents [9]. For the two-stage detection algorithm, the first stage uses a region proposal network to generate multiple candidate regions. In the second stage, these candidate regions are screened, categorized and regressed to obtain the final detection results. For the one-stage detection algorithm, there is no need to generate candidate regions but to generate the category probability and location direct coordinate values of the target to be tested, and the final detection results can be directly obtained after a single detection. Compared with the two-phase detection algorithm, the one-phase detection algorithm can maintain a balance between real-time detection and high accuracy and better meet the needs of target detection in unmanned aerial vehicle (UAV) aerial images. The two-phase detection algorithm needs to consume a lot of computational resources when running due to the limitations of the internal framework, and it cannot meet the real-time requirements of computing devices.

In recent years, target detection in aerial images based on single-stage detection algorithms has received more and more attention. Liu et al. [10] proposed UAV-YOLO based on the YOLOv3 detection algorithm, which improves the whole network structure and enriches spatial information by adding shallow convolution. However, there are leakage and false detection problems for small targets. Liang et al. [11] proposed a single-stage detection model FS-SSD based on feature fusion and scale scaling, which utilizes the inverse convolution and feature fusion modules for prediction and further improves detection accuracy through contextual analysis. Liu et al. [12], based on CenterNet, improved the detection accuracy of the network by adding an adaptive base module, a global attention module and a high-quality decoding module. However, the large number of parameters required for the network and the high arithmetic requirement made it challenging to deploy in UAV platforms. Huang et al. [13] improved the detection accuracy of the YOLOv5s network by introducing the shufflenetv2 feature extraction structure, which reduces the computation of the network but causes a large amount of accuracy loss. Xu and Mao [14] proposed a multilayer feature fusion algorithm for UAV aerial image detection based on YOLOv5, which improves the accuracy of small targets in aerial images by fusing different layers of feature maps to aggregate contextual information. However, the computational resources consumed are as high as 109.3 GFLOPs, leading to slow detection speed and failing to meet the real-time detection requirements. Liu et al. [15] proposed a one-stage aerial image target detection algorithm, RelationRS. This framework combines the bi-relational module and the bridging visual representation module to solve the multi-scale fusion problem of aerial images in complex backgrounds, improving the target detection accuracy but not meeting the real-time detection requirements.

For small target detection, researchers have done many studies. Chu et al. [16] proposed a small target detection algorithm based on multi-layer convolution feature fusion based on regional proponent network (R-CNN). By fusing feature information from high and low feature layers to improve the detection accuracy of small targets in intelligent traffic detection scenarios. Sheikhpour et al. [17] proposed a Hessian-based semi-supervised feature selection using a generalized uncorrelated constraint method to help the algorithm eliminate redundant features and extract information-rich favorable features. Lin et al. [18] proposed feature pyramid network (FPN) to achieve multi-scale feature fusion for the first time for different scales feature maps, but the fusion of FPN is unidirectional, and there is inevitably the problem of insufficient fusion of feature information. The path-aggregation network (PANet) proposed by Liu et al. [19] was designed to use top-down and bottom-up bidirectional paths to fuse multi-scale feature maps in an additive manner to enhance the representation of deeper feature information with accurate localization signals. Tan et al. [20] proposed a weighted bidirectional feature pyramid network (BiFPN) based on [19] which enhances the representation of features through weighted fusion and residual connectivity. The above small target detection methods can make up for the differences in spatial and semantic information between the shallow and deep feature maps to a certain extent. However, at the same time, they also introduce more parameters and calculations which cannot meet the real-time detection needs of small target detection in aerial images.

To address the problems of insufficient real-time performance and difficulty in improving the accuracy of small target detection in the current target detection algorithms for aerial images, we propose a cross-scale multi-feature fusion target detection method (CMF-YOLOv5s) for aerial images based on the lightweight target detection algorithm YOLOv5s, and the main contributions are as follows: 1) We constructed a bi-directional cross-scale feature fusion sub-network in the feature fusion stage. Specifically, we design a multi-scale feature fusion module (MFF) to fuse and extract global and local information from different sensory fields of the same feature map to reduce the problem of small object feature information loss due to repeated sampling during the fusion process. At the same time, we use a cross-scale feature fusion strategy to fully fuse the shallow large-scale feature maps that are beneficial for small object detection with other deep small-scale feature maps to increase the importance of small object information in the overall feature map and further enhance the characterization capability of the small object. 2) We constructed a multiscale detection head to improve the multiscale object recognition capability of the model by changing tri-scale detection into quad-scale detection based on the original algorithm to reduce the number of missed detections of small objects. 3) According to the characteristics of aerial images, the K-means clustering algorithm is optimized to generate 16 groups of anchor boxes of different sizes for multi-scale object detection, to improve the recall of small objects.

2. YOLOv5s

The YOLOv5 network is widely used in object detection tasks due to its high accuracy, fast detection speed and ease of deployment on different hardware platforms. YOLOv5s, as a lightweight version of YOLOv5 series algorithms, has a small and streamlined network and high detection accuracy and speed, and it is one of the most popular target detection algorithms in the current development of industrial intelligence. As shown in Figure 1, YOLOv5s uses CSPNet (C3) as the backbone feature extraction network to form a multi-scale feature map and PANet to fuse the multi-scale feature map. The final detection results are output by a triple-scale detection head. This structure is designed to be enough for general image object detection tasks. However, for aerial images containing many dense small-scale objects, YOLOv5s has the problem of easy loss of small object feature information during multi-scale feature fusion, which leads to poor detection accuracy of YOLOv5s on aerial images.

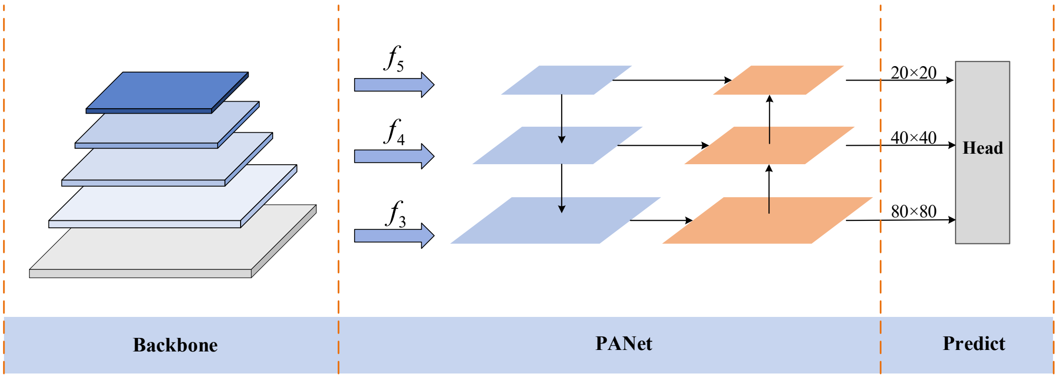


Figure 1. Architecture of YOLOv5s.

3. CMF-YOLOv5s

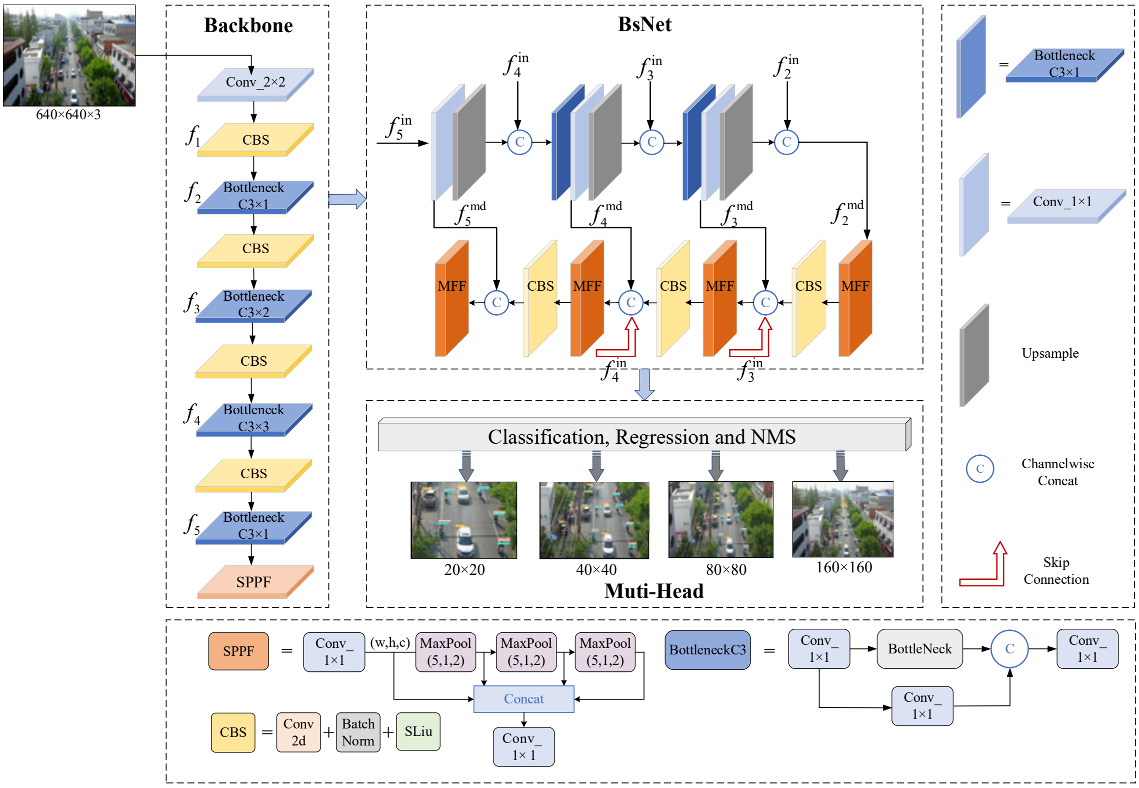


Figure 2. Architecture of CMF-YOLOv5s. Please refer to the "Appendix" at the end of the paper for the detailed composition of each network layer and the input/output parameters of each layer.

We propose a lightweight object detection method for the aerial image based on the YOLOv5s_v6.2 network [21], as shown in Figure 2. The CMF-YOLOv5s comprises a feature extraction network, a bi-directional cross-scale feature fusion network (BsNet) and a multi-scale detection head (Multi-Head). The feature extraction network is CSPNet (C3), mainly composed of BottlenackC3, CBS and SPPF. The function of CSPNet (C3) is to increase the network receptive field, extract each scale object's feature information in the input image and form five different scale feature maps from f_1 to f_5. BsNet fuses the feature maps at different scales to enrich the feature information representation of small object samples by the multi-dimensional cross-scale intermingling of deep and shallow feature information. The multi-scale detection head performs classification, regression and non-maximal suppression (NMS) of category and location information for objects of different sizes in the same image at four scales and outputs the detection results. Specifically, the head of 20 × 20 size is used to detect large-scale objects in the image, and the head of 160 × 160 is taken to detect the smallest-size objects in the image.

3.1. BsNet network

The shallow network extracts object texture edge features with a more comprehensive detail description. The deeper network extracts the rich semantic features of the object. Still, it simultaneously weakens the perception of small object location information and detail information, causing the feature information of the small object in the feature map to be lost [22]. YOLOv5s uses PANet to fuse different scale feature information in each layer by an equal relationship, ignoring the importance of varying depth feature layers for different scale object detection, and only combines multi-scale features for layers f_3 to f_5 without fusing the output of the shallow feature map f_2 which mainly contains small object feature information. To address the above problems, we construct a bi-directional cross-scale feature fusion network, as shown in Figure 3. First, a bottom-up and top-down bi-directional path is used to coarsely fuse the multi-scale feature information of the four-scale feature maps from f_2 to f_5. Meanwhile, a new multi-scale feature fusion module (MFF) is designed to achieve fine-grained feature extraction and fusion from different perceptual fields to improve the characterization capability for small object feature information. In addition, cross-scale feature fusion is performed between the parallel fusion paths of $\{f_3, f_4\}$ and $\{f_4, f_5\}$ to enrich further the semantic information and localization information of the feature maps at each level.

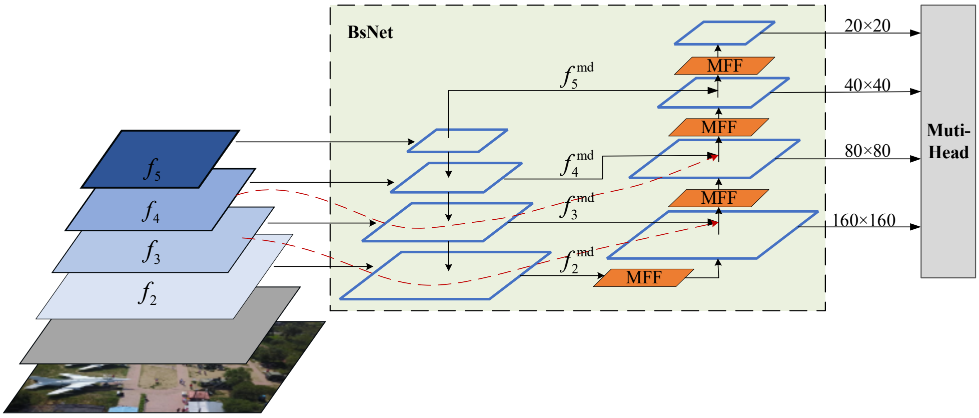


Figure 3. The structure of the BsNet network. The red dotted line in the figure represents cross-scale feature fusion.

3.1.1. MFF module

Please refer to the Symbol List for the symbols covered in this paper and their descriptions, as shown in Table 1.

Table 1. List of symbols.

Symbol	Description
$f_{(x)}^{\text{in}}$	Input feature map, x is the feature layer variable
$f_{(x)}^{\text{md}}$	Intermediate feature map of the fusion process
$f_{(x)}^{\text{out}}$	Output feature map
$Concat(\bullet)$	Feature summation based on channel dimension
$Resize(\bullet)$	Image size scaling function
N_{recall}	Maximum number of ground truths that can be recalled
N_{gt}	Total number of ground truths
IOU	Intersection over Union, the mathematical description: $\frac{A\cap B}{A\cup B}$

Many small objects are in the aerial images, and little feature information is available after extraction. PANet of YOLOv5s fuses features by continuously upsampling and downsampling the feature map to enable multi-scale information. Nevertheless, the repeated sampling operation causes a lot of small object information to be lost. To solve the above problem, we design a multi-scale feature fusion module (MFF), as shown in Figure 4, which consists of Unit A and Unit B in cascade. For the input feature map $f_{(x)}^{\text{md}}$, Unit A extracts the fine-grained features of $f_{(x)}^{\text{md}}$ from different scale perceptual fields and fuses the features again on the channel dimension by "Concat" to form the more detailed information-rich $f_{(x)}^{\text{md}'}$ feature map. Then, in Unit B, local and global information depth extraction is performed on $f_{(x)}^{\text{md}'}$ in two branches to obtain the final enhanced feature map $f_{(x)}^{\text{out}}$ for the detection output. The fusion and re-extraction of feature information by the MFF module can effectively avoid the loss of small object information caused by sampling operations during the fusion process.

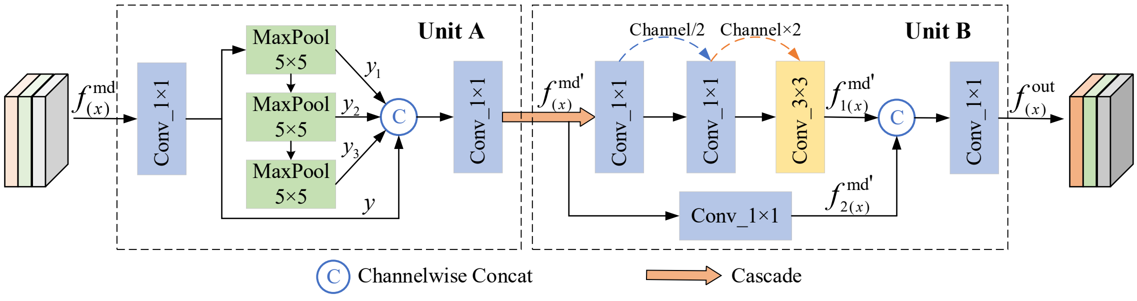


Figure 4. Schematic of the MFF.

The function of Unit A is to fuse the feature information under different receptive fields of the same feature map through "Concat" to enrich the detailed information of the feature map. For the input feature map $f_{(x)}^{\text{md}}$, it first passes through a 1 × 1 convolutional layer to form the feature map y and then serially passes through three convolutional kernel size 5 × 5 MaxPool layers. Three feature maps of different sizes, y_1, y_2 and y_3, are formed, where the size of y_1 is 5 × 5, the size of y_2 is 9 × 9, and the size of y_3 is 13 × 13. Finally, the feature information extracted under the four scale receptive fields is fused to obtain $f_{(x)}^{\text{md}'}$. The fusion process can be described by Eq (3.1), where x indicates the multi-scale feature map variable, and "Concat" indicates the summation of features based on channel dimensions:

$$\begin{cases} f_{(x)}^{\text{md}'} = Conv_{1\times1}[Concat(y, y_1, y_2, y_3)], \\ y = \text{C}onv_{1\times1}(f_{(x)}^{\text{md}}), \\ y_1 = MaxPool_{5\times5}(y), \\ y_2 = MaxPool_{5\times5}(y_1), \\ y_3 = MaxPool_{5\times5}(y_2). \end{cases} \tag{3.1}$$

Unit B's function is to fully extract the global and local information from the same feature map, enhance the features' edge and texture information and improve the recognition rate of small objects. Unit B performs feature extraction on the input feature map $f_{(x)}^{\text{md}'}$ by two branches. On the branch $f_{1(x)}^{\text{md}'}$, the 1 × 1 convolution reduces the number of channels to half the original size, and the 3 × 3 convolution doubles the number of channels. The "downscaling-extraction-Upgrading" pattern will help the network to extract more global information. The $f_{2(x)}^{\text{md}'}$ branch uses only a 1 × 1 convolution, which means that the size of the feature map is not changed, so it keeps the spatial resolution of the feature map from being reduced, thus better preserving the local information of the object. The feature extraction process for Unit B can be described by Eqs (3.2)–(3.4):

$$f_{1(x)}^{\text{md}'} = \{Conv_{3\times3}[Conv_{1\times1}(f_{(x)}^{\text{md}'})]\} \tag{3.2}$$

$$f_{2(x)}^{\text{md}'} = Conv_{1\times1}(f_{(x)}^{\text{md}'}) \tag{3.3}$$

$$f_{(x)}^{\text{out}} = Concat(f_{1(x)}^{\text{md}'}, f_{2(x)}^{\text{md}'}) \tag{3.4}$$

3.1.2. Cross-scale feature fusion

The PANet adds up each multi-scale feature map in the feature fusion process, ignoring the importance of shallow and large-scale feature maps for small object detection. Therefore, we fused the feature maps of layers f_3 and f_4 with the feature maps of layers f_3 and f_4 by using jump connections to fully extract the potential small object information in the feature maps when constructing the BsNet. In the meantime, the fast normalized fusion formula (fast normalized fusion), the calculation formula shown in (3.5), is introduced to assign weights equally to different scale features to enhance the importance of small object feature weights.

$$O = \sum_i \frac{\omega_i}{\varepsilon + \sum_j \omega_j} \cdot I_i \tag{3.5}$$

where I_i is the input value, the O is the output value, the Relu function ensures $\omega \geq 0$, and the ε is a minimal value greater than 0, which is used to avoid numerical instability. Eventually, the value of each weight falls between 0 and 1 by normalization calculation. We take the multi-scale fusion process of the fourth layer feature map as an example, and the calculation process is shown in Eqs (3.6) and (3.7).

$$f_4^{\mathrm{md}} = Conv\left(\frac{\omega_1 \cdot f_4^{\mathrm{in}} + \omega_2 \cdot Resize\left(f_5^{\mathrm{md}} \right)}{\varepsilon + \omega_1 + \omega_2} \right) \tag{3.6}$$

$$f_4^{\mathrm{out}} = Conv\left(\frac{\omega_3 \cdot f_4^{\mathrm{in}} + \omega_4 \cdot f_4^{\mathrm{md}} + \omega_5 \cdot Resize\left(f_3^{\mathrm{out}} \right)}{\varepsilon + \omega_3 + \omega_4 + \omega_5} \right) \tag{3.7}$$

where *Conv* represents the convolution operation, f_4^{in} and f_4^{out} represent the input and output of the layer four feature map, and the *Resize* represents the upsampling operation. The computational flow of cross-scale feature fusion is shown in Figure 5, and the same process is used for all other layers.

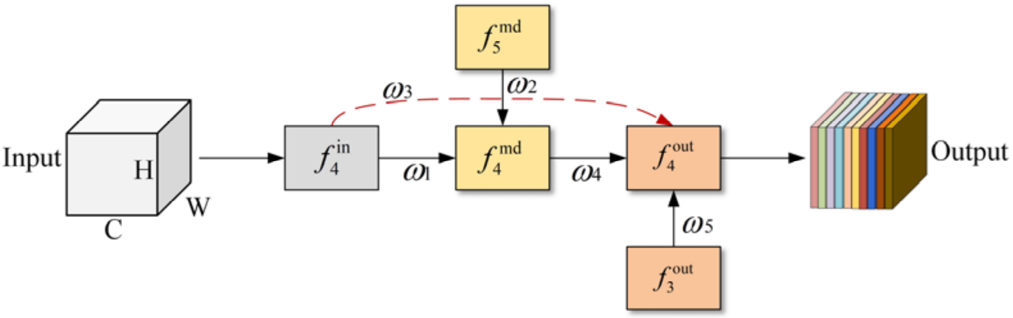


Figure 5. The cross-scale feature fusion process of fourth layer.

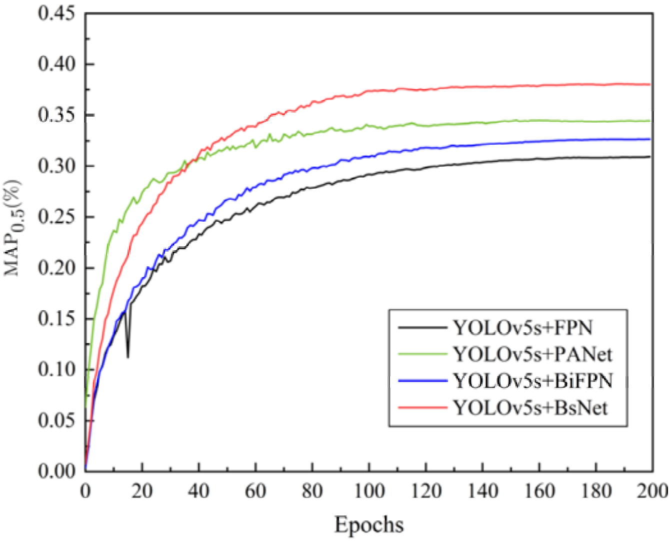


Figure 6. Comparison of the mean accuracies ($\mathrm{mAP}_{0.5}$) of FPN, PANet, BiFPN, and BsNet.

To verify the effectiveness of the BsNet network, BsNet was compared with the classical FPN, PANet and BiFPN feature fusion networks. Keeping the experimental conditions constant, the feature fusion networks of YOLOv5s were replaced with FPN, PANet, BiFPN and BsNet. Figure 6 shows the variation of the average accuracy ($mAP_{0.5}$) with the number of iterations (Epochs) after replacing different feature fusion networks on YOLOv5s.

The experiments show that with the increase of Epochs, the best average detection accuracy is obtained for the trained model when BsNet is used as the feature fusion network. Compared with other feature fusion methods, the BsNet constructed in this paper has better multi-scale feature fusion capability.

3.2. Multi-scale feature head

For object detection, smaller feature maps correspond to more large mapping regions in the original image, allowing the detection of larger objects in the image. However, the large mapping regions lack detailed information, which causes small objects to be less likely to be detected. As shown in Figure 7(a), YOLOv5s uses three scale detection heads of 80 × 80 × 128, 40 × 40 × 256, and 20 × 20 × 512 to detect objects of small, medium and large image sizes. However, the shallow large-scale feature map is more critical in improving detection accuracy for aerial images containing numerous small objects. We constructed a multi-scale detection head to address these issues, as shown in Figure 7(b). A new 160 × 160 × 64 detection head is added for tiny object detection in images based on the head structure of the original YOLOv5s. The four-scale detection head helpfully improves detection accuracy while reducing the missed detection rate of small objects.

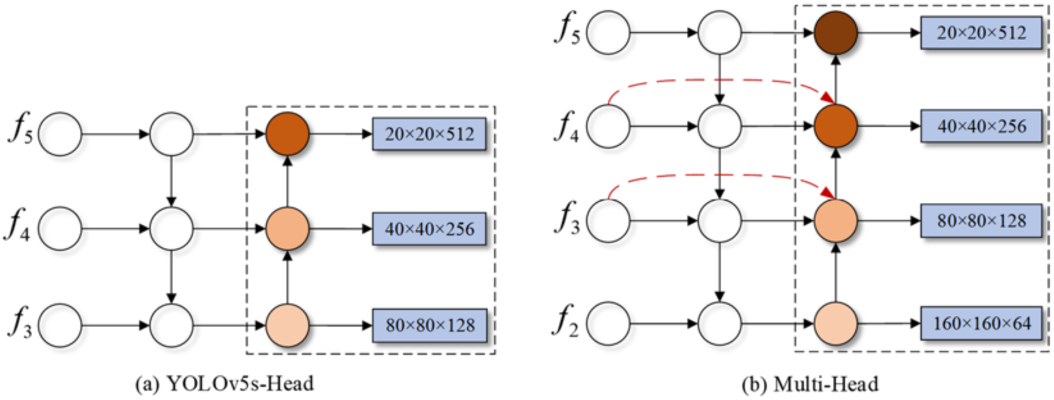
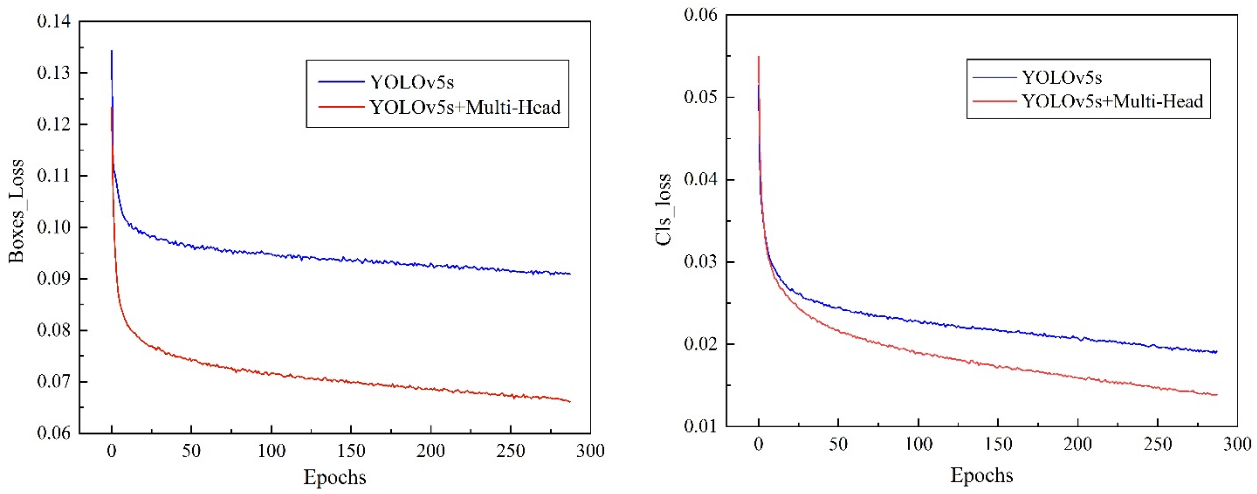


Figure 7. Comparing head structure before and after improvement.

Figure 8 shows the trend of each loss value with the number of iterations (Epochs) during the training process for the original YOLOv5s and the YOLOv5s with the improved head structure, with Boxes_loss indicating the mean loss value of bounding boxes and Clc_loss indicating the mean loss value of classification. Smaller values of Boxes_loss and Cls_loss indicate faster convergence of the model and better performance in object prediction and classification. The experimental results show that the training loss values of YOLOv5s are significantly lower than those of YOLOv5s after replacing Multi-Head in the case of using the same loss function, indicating that the improved Multi-Head structure can effectively improve the accuracy of the network in object classification and prediction.



(a) Train/Boxes_loss

(b) Train/Cls_loss

Figure 8. Comparison of training loss values.

3.3. Optimized anchor boxes

The YOLO series algorithm uses anchor boxes to constrain the range of predicted objects and incorporates the previous experience of setting sizes to improve model learning efficiency while helping the model converge quickly. The original YOLOv5s algorithm uses the K-means clustering algorithm to calculate the distance between the authentic label boxes and the predicted boxes in the dataset based on the Euclidean distance to generate anchor boxes applicable to the current dataset. However, as can be seen from Table 2, the K-means algorithm only generates nine sets of anchor boxes on the three scale feature maps, which cannot meet the detection needs of realistic aerial scenes with significant differences in object scales. In order to better improve the matching between anchor boxes and each size object on each feature map, we use IOU distance [23] to calculate the similarity between actual label boxes and prediction boxes, with BPR [24] (best possible recall) as the index. At the same time, the genetic algorithm is introduced based on the K-means algorithm to optimize the search results and calculate the anchor boxes that better match the current data set. The algorithm flow is shown in Table 2.

Table 2. K-means & Genetic algorithm flow.

```
Algorithm: K-means & Genetic algorithm
Data Generation: H: Height of the sample boxes, W: Width of the sample boxes
Inputs: K: number of clusters, BPR: 0.98, Max: maximum number of iterations (1000)
Outputs: width and height of anchor boxes for BPR greater than or equal to 0.98
While  BPR ≥ 0.98 or Max = 1000 do:
    (1). Initialize K anchor boxes
    (2). Calculate d: 1-IOU (box, centroid) and pick the smallest value of d and Update the value of K
    (3). Calculate BPR
    (4). When BPR < 0.98, the Genetic algorithm is applied to search for the optimal width and height of
anchor boxes, update the width and height of anchor boxes, and iterate continuously
return Optimal Anchor boxes width and height values
```

As shown in Table 3, the optimized anchor boxes have 16 scale values in four rows and four columns. Each detection head has four scales of anchor boxes, and they can be matched to various sizes of objects in the feature map, effectively reducing the miss-detection of small objects.

Table 3. Optimization of the anchor boxes size.

Feature map sizes	Anchor boxes of YOLOv5s	Optimized anchor boxes
160 × 160	None	[3,4; 4,9; 7,6; 7,13]
80 × 80	[3,4; 4,9; 8,6]	[13,7; 12,12; 10,19; 22,11]
40 × 40	[7,14; 15,9; 15,19]	[19,17; 16,26; 33,18; 28,32]
20 × 20	[31,17; 25,37; 55,42]	[47,28; 40,58; 86,53; 97,126]

Equation (3.8) is a formula designed by the authors of YOLOv2 for the target detection problem to calculate the distance between the predicted frame and the truth box using IOU as the metric. Compared with the Euclidean distance calculation formula, the IOU distance formula aligns more with the target detection distance calculation principle. In the formula, "box" denotes the position of the prediction box, and "centroid" denotes the position of the center point of the truth box, and the smaller the value of d(box, centroid) is, the higher the fitness of the anchor boxes to the dataset. Equation (3.9) is the formula for the best possible recall (BPR), which was initially proposed in the Fully convolutional one-stage object detection (FCOS) paper to measure the best recall of a dataset, where a larger value means that the model is more capable of identifying positive samples, and the optimal value is 1. N_{recall} represents the number of labeled boxes of objects that can be recalled in the dataset, and N_{gt} represents the total number of labeled boxes of objects.

$$d(\text{box}, \text{centroid}) = 1 - IOU(\text{box}, \text{centroid}) \tag{3.8}$$

$$\text{BPR} = \frac{N_{\text{recall}}}{N_{\text{gt}}} \times 100\% \tag{3.9}$$

Table 4 shows the BPR values of YOLOv5s's anchor boxes and the optimized anchor boxes. The experiments show that the BPR of the original anchor boxes is only 0.933, which is less than 0.98 and does not meet the requirements. The match between our optimized Anchor boxes and the labeled boxes of objects reached 0.998, indicating that the improved anchor boxes are more suitable for the dataset and can effectively improve the detection precision of the aerial image objects.

Table 4. Comparison of BPR values.

	Anchor boxes of YOLOv5s	Optimized Anchor boxes
BPR/%	0.933	**0.998**

4. Experimental results

4.1. Data sets and evaluation metrics

To verify the performance of CMF-YOLOv5s object detection in aerial images, we conduct experiments on the publicly aerial image dataset VisDrone-2019 [25]. The dataset contains 8629 images with 260 manually annotated label boxes comprising ten categories, including pedestrians, cars, etc. In the COCO dataset [26], objects with an area smaller than 32 × 32 pixels are defined as small objects. According to statistics, the percentage of small objects in the COCO dataset is 21.73% [27], while compared to the distribution of small objects in the COCO dataset, the percentage of objects smaller than 32 × 32 pixels in the VisDrone-2019 dataset is as high as 44.70%. Figure 8 shows the distribution of data label sizes in the VisDrone-2019 training set, where the horizontal coordinate of width is the ratio of the data label to the width of the whole image, and the vertical coordinate of height indicates the ratio of the data label to the height of the whole image. The smaller the value of the coordinate point composed of width and height is, the smaller the size of the label box in the image. It can be seen from Figure 9 that most of the objects in the data set used in the experiment are small objects, which are consistent with the characteristics of natural aerial photography scenes.

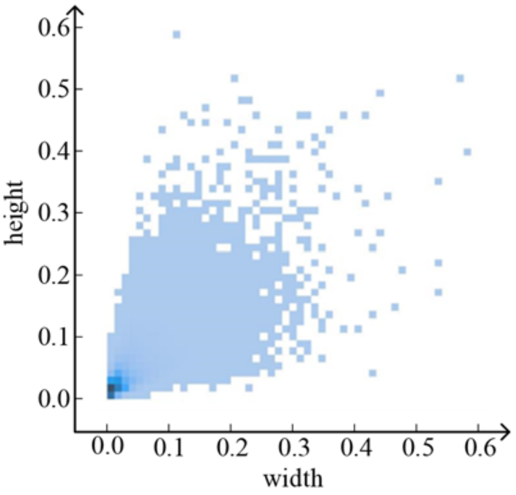


Figure 9. Label size distribution of VisDrone-2019 training set data.

In our experiment, precision (P), recall (R), average precision (AP), mean average precision (mAP) and frames per second (FPS) were used as the evaluation indicators of model performance. $AP_{0.5}$ represents the average detection precision of each category in the data set when the IOU threshold is 0.5, $mAP_{0.5}$ represents the average value of $AP_{0.5}$ when the IOU threshold is 0.5, and $mAP_{0.5:0.95}$ represents the average of 10 mAP obtained for IOU thresholds of 0.5 to 0.95. FPS is a measure of the speed of detection of the algorithm. The time of the algorithm to detect an image includes the image pre-processing (Pre) time, inference (Infer) time and non-maximum suppression (NMS) time. A higher value of FPS on the same hardware device means that the algorithm processes the data faster. The formula for the evaluation metrics is shown in Eqs (4.1)–(4.5).

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

$$\mathrm{AP} = \int_0^1 \mathrm{P(R)}\,\mathrm{dR} \tag{4.3}$$

$$\mathrm{mAP} = \frac{\sum_{i=1}^{\mathrm{N}} \mathrm{AP}_i}{\mathrm{N}} \tag{4.4}$$

$$\mathrm{FPS} = \frac{1}{\mathrm{Pre+\ Infer+\ NMS}} \tag{4.5}$$

TP denotes the number of positive samples correctly predicted by the model, FP denotes the number of negative samples incorrectly predicted by the model and FN denotes the number of positive samples incorrectly predicted by the model in the formula.

4.2. The environment and parameters of the experiment

Table 5. Setting of the main experimental parameters.

Parameter	Value
Epochs	300
Batch size	32
Image size	640 × 640
Initial learning rate	0.01
Optimization algorithm	SGD

The operating system used in this experiment is Ubuntu 22.04, the experimental software was Anaconda 2.1.1 and PyCharm 2022.2.3, and the experimental environment is Python 3.8 + PyTorch 1.10.1 + CUDA 11.1. All algorithms in this experiment were run on an NVIDIA RTX 3060Ti GPU graphics card, and the same hyperparameters are used for training, validation and testing. Table 4 shows the main parameter settings for the algorithm's training. As shown in Table 5, the training model parameters in this paper were set as follows: The SGD optimization algorithm was used with an initial learning rate of 0.01. The activation function is Sigmoid, and the loss function is CIOU. The training batches were 300 times, with 16 images passed in at a time in each training batch for training. The input image size was 640 × 640, and the last iteration model weights and the best performance model weights were stored.

4.3. Comparative experimental analysis

To verify the superiority of this paper's algorithm compared with other algorithms, CMF-YOLOv5s was compared with eight lightweight methods, including YOLOv5s, EfficientNet [28], MobileNet [29], YOLOv3-Tiny [30], YOLOX-s [31], PP-PicoDet-s [32], YOLOv7-Tiny [33] and PP-YOLOE-s [34]. Specifically, we designed four comparative experiments, including a comparison of the detection accuracy of different methods, a comparison of the detection accuracy of different methods for each category of the object, a comparison of the detection accuracy of each version of YOLOv5 and CMF-YOLOv5 and a comparison of visual effects.

4.3.1. Comparison of the detection accuracies of different methods

The results of the detection accuracy metrics comparison between the method in this paper and the eight lightweight methods are shown in Table 6. The experimental results show that the accuracy metrics of CMF-YOLOv5s are better than the other comparison algorithms. Specifically, compared to the other methods, CMF-YOLOv5s improved P and R by at least 2.9% and 0.5%, $mAP_{0.5}$ by more than 3.3%, and $mAP_{0.5:0.95}$ by more than 1.9%. Meanwhile, the $mAP_{0.5}$ and $mAP_{0.5:0.95}$ of this paper's method improved by 5.5% and 3.6% compared to the benchmark method YOLOv5s.

Table 6. Experimental results of the comparison of the detection performances of different methods.

Method	P/%	R/%	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%	FPS
YOLOv5s	50.1	33.6	34.5	18.7	119
EfficientNet	34.5	30.2	26.8	13.3	106
YOLOv7-Tiny	47.2	38.0	35.6	18.9	**140**
PP-PicoDet-s	42.5	37.7	36.0	19.8	48
YOLOX-s	49.4	36.0	36.7	20.4	99
MobileNet	33.3	22.4	20.6	9.69	113
YOLOv3-Tiny	28.3	18.7	15.4	6.60	132
PP-YOLOE-s	41.4	33.8	31.8	16.7	111
Ours	**53.0**	**38.5**	**40.0**	**22.3**	116

4.3.2. Comparison of detection accuracies of different methods on each category of the object

Table 7. Experimental results of comparison of detection accuracy on each category.

Methods	$AP_{0.5}$/%									
	pedestrian	people	bicycle	car	van	truck	tricycle	awning-tri	bus	motor
YOLOv5s	40.2	33.4	12.8	74.3	37.3	32.0	20.9	12.0	42.8	39.7
EfficientNet	34.4	26.8	6.40	69.3	25.4	21.0	13.2	8.40	29.6	33.6
YOLOv7-Tiny	39.3	36.3	10.0	76.8	38.5	31.2	22.4	10.7	47.2	43.8
PP-PicoDet-s	44.6	33.6	10.5	79.3	40.7	30.2	20.5	10.7	47.4	41.8
YOLOX-s	42.9	36.1	14.7	75.9	38.4	32.5	22.3	11.9	47.5	42.6
MobileNet	23.2	20.0	3.90	61.5	19.9	16.1	9.90	5.20	23.2	23.3
YOLOv3-Tiny	17.5	17.1	3.1	49.3	13.0	11.7	7.9	4.3	13.1	17.4
PP-YOLOE-s	39.5	31.1	8.8	73.1	33.6	27.0	17.0	9.9	40.1	38.2
Ours	**47.9**	**36.9**	**15.2**	**81.0**	**42.5**	**36.1**	**26.5**	**12.7**	**54.4**	**44.8**

To intuitively demonstrate the advantages of this paper's method over other methods in terms of detection accuracy, we conducted single-category detection accuracy comparison experiments for each of the ten categories in the VisDrone-2019 dataset, and $AP_{0.5}$ was used to measure the detection accuracy performance of each category, and Table 7 shows the experimental results. The experimental results show that on the VisDrone-2019 dataset, compared with other methods, CMF-YOLOv5s improved detection accuracy for each category in different degrees, with $AP_{0.5}$ improving by at least 0.5–6.9%, further demonstrating that the method in this paper can effectively improve the accuracy of object detection in aerial images.

4.3.3. Comparison of the detection accuracy of YOLOv5 and CMF-YOLOv5

Like previous YOLOv5 networks, YOLOv5_v6.2 has been categorized into five versions: n, s, m, l and x, to satisfy the needs of different detection scenes. To validate the effectiveness of the improvement strategy proposed in this paper on other versions of the YOLOv5 network, the improvement strategy of this paper was applied to four networks of YOLOv5_v6.2, including YOLOv5n, YOLOv5m, YOLOv5l, and YOLOv5x, and compare with the network before improvements. As can be seen from the data in Table 8, the average detection accuracy of CMF-YOLOv5 varies from 3% to 7.3% higher than that of YOLOv5 for the corresponding versions. The experiments demonstrate that the improved strategy in this study is not affected by the network complexity of different versions of YOLOv5 and has a certain degree of generalizability.

Table 8. The detection accuracy results of the comparison of YOLOv5 and CMF-YOLOv5.

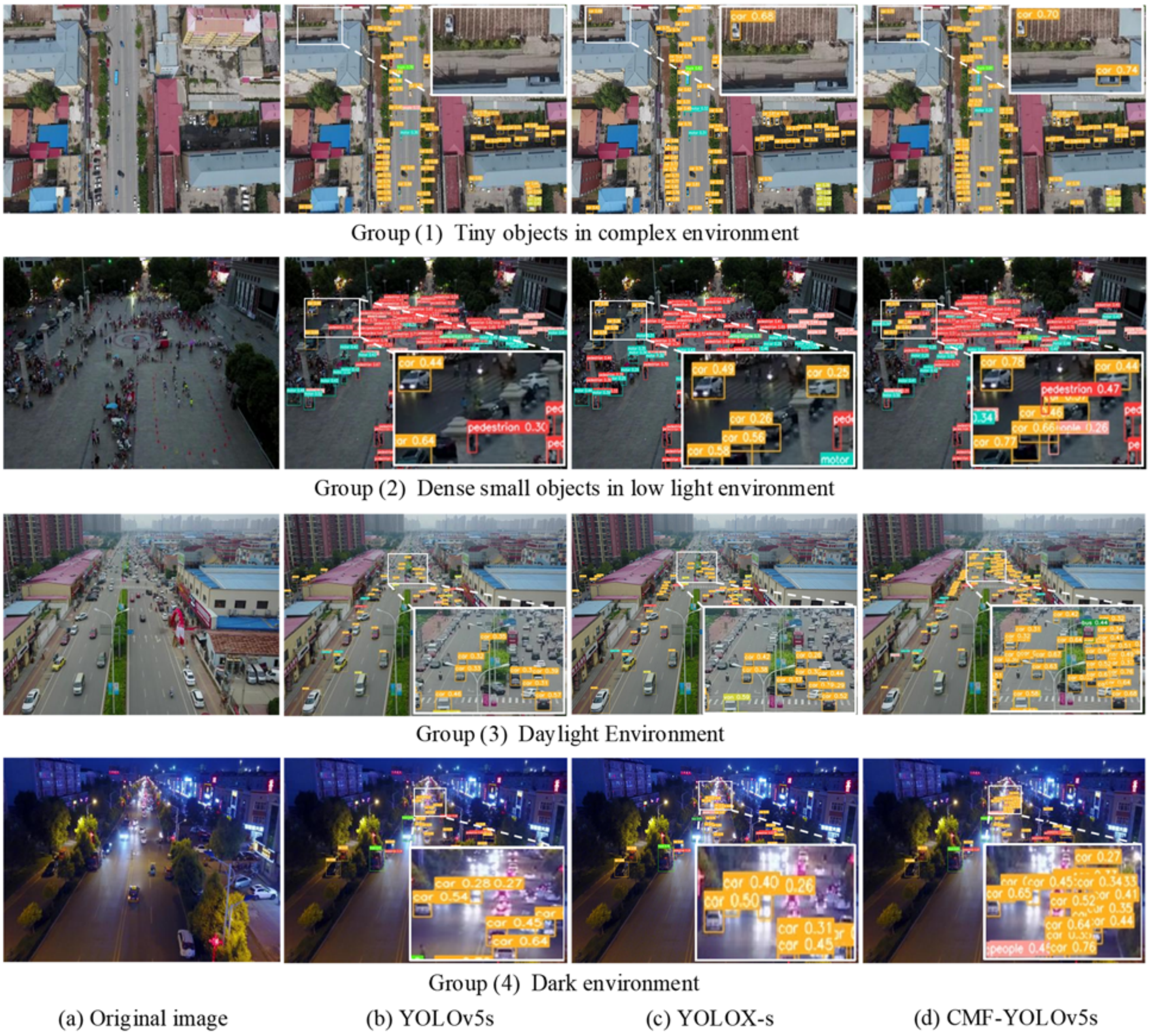
Method	P/%	R/%	$mAP_{0.5}$/%	$mAP_{0.5:0.95}$/%
YOLOv5n	33.0	28.2	24.7	12.1
CMF-YOLOv5n	**41.4**	**31.4**	**30.7**	**15.8**
YOLOv5s	50.1	33.6	34.5	18.7
CMF-YOLOv5s	**53.0**	**38.5**	**40.0**	**22.3**
YOLOv5m	46.2	37.4	35.7	19.7
CMF-YOLOv5m	**53.9**	**40.5**	**42.6**	**24.3**
YOLOv5l	49.2	38.3	38.2	21.7
CMF-YOLOv5l	**56.0**	**43.3**	**45.5**	**26.8**
YOLOv5x	49.4	40.9	39.9	22.8
CMF-YOLOv5x	**54.8**	**46.0**	**46.9**	**27.9**

4.3.4. Comparison of visual effects

To display the detection effect of this paper's method in complex aerial photography scenes, we conducted visual effect comparison experiments on the VisDrone-2019-DET-test-challenge dataset (1580 images), and we selected some representative aerial images scenes from which to display the effect. The selected scenes have in common a complex detection background and numerous small objects that are difficult to detect. A comparison of the detection results is shown in Figure 10, where

(a) is the original input image, (b) is the detection result of the baseline algorithm YOLOv5s, (c) is the detection result of YOLOX-s, which is the second most accurate detection algorithm in the comparison algorithm after the accuracy of this paper, and (d) is the detection result of the method in this paper.

In Figure 10, Group (1) shows the detection result of tiny objects in complex backgrounds, and CMF-YOLOv5s locates the tiny object in the upper left corner of the image that is difficult to detect by vehicles and correctly outputs the object class information car. Group (2) indicates that CMF-YOLOv5s can better detect dense vehicles in low light conditions and obscured pedestrians in the distance compared to the YOLOv5s and YOLOX-s. Groups (3) and (4) display the result in detecting vehicles on urban roads in both daytime and nighttime conditions, and the result indicates our method can detect many dense and small objects in the image with a more distant field of view. In summary, the proposed CMF-YOLOv5s can detect small and densely distributed objects in complex aerial scenes and identify the object category accurately, regardless of whether it is daytime, nighttime or under low light conditions.



Group (1) Tiny objects in complex environment

Group (2) Dense small objects in low light environment

Group (3) Daylight Environment

Group (4) Dark environment

(a) Original image (b) YOLOv5s (c) YOLOX-s (d) CMF-YOLOv5s

Figure 10. Comparison of the detection effects of different aerial photography scenes.

To verify the effectiveness of the method in this paper, we compared the detection effects of CMF-YOLOv5s with six lightweight algorithms, and the detection results are shown in Figure 11. As shown in the figure, compared with the other lightweight algorithms, CMF-YOLOv5s can identify more objects in complex aerial photography scenes and better recognize dense and small objects further away from the field of view. The visual experimental results show that the method in this paper effectively improves the detection accuracy of the lightweight algorithm on small objects in aerial images, the detection time is only about 0.008 s per image, and the detection frame rate per second can reach 116 FPS, meeting the demand for real-time detection of aerial images.

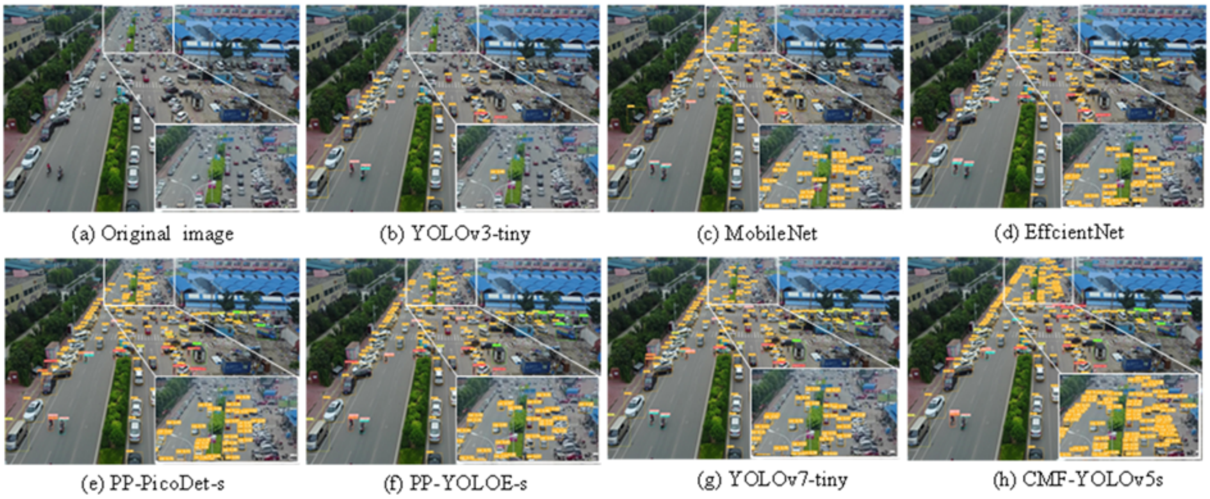

(a) Original image
(b) YOLOv3-tiny
(c) MobileNet
(d) EffcientNet
(e) PP-PicoDet-s
(f) PP-YOLOE-s
(g) YOLOv7-tiny
(h) CMF-YOLOv5s

Figure 11. Comparison of the detection results.

4.4. Ablation experiments analysis

In order to verify the effectiveness of each improvement strategy, we designed ablation experiments, as shown in Table 9 on the VisDrone-2021 dataset, using YOLOv5s as the benchmark network. The " √ " in the table indicates the improved modules. The experimental results show that the detection accuracies of YOLOv5s are improved after each module is added sequentially. Moreover, BsNet not only makes the algorithm's detection accuracy better but also speeds up the algorithm's detection speed. When each improvement strategy is added to YOLOv5s at the same time, the average detection accuracy of the algorithm is improved by 5.5% under the FPS of 116, which is the optimal value that is not achieved by adding each improvement measure alone. In fulfilling the real-time detection requirement, CMF-YOLOv5s gets more accuracy improvement with a slight FPS loss.

Table 9. Results of ablation experiments.

Groups	BsNet	Multi-Head	Optimized anchor boxes	$mAP_{0.5}$/%	FPS
A				34.5	119
B	√			37.3	**120**
C	√	√		38.4	117
D	√	√	√	**40.0**	116

5. Conclusions

For the problem of many small targets in aerial images, which causes difficulty in detection, we propose a small target detection algorithm CMF-YOLOv5s with better performance based on the lightweight network YOLOv5s for enhancing the target detection accuracy of aerial images. First, the ability of the algorithm to fuse multi-scale information is enhanced by constructing a bidirectional cross-scale feature fusion sub-network. Second, the recognition rate of small targets is improved by constructing a multi-scale detection head. Finally, K-means, the algorithm for generating anchor boxes, is improved to obtain anchor boxes more suitable for aerial images. Many experiments have shown that the method proposed in this paper can effectively improve the performance of target detection accuracy in aerial images with complex backgrounds and dense small targets. On the VisDrone-2021 dataset, the proposed method improves the $mAP_{0.5}$ and $mAP_{0.5:0.95}$ by at least 3.3% and 1.9% compared with eight lightweight methods, such as YOLOv7-Tiny. Meanwhile, the detection speed of the proposed method reaches 116 FPS, which meets the real-time demand of aerial image detection and is favorable for applying in actual aerial image detection tasks. Moreover, the improved strategy in this paper has a certain degree of generalization and can be extended to YOLOv5 networks of different sizes to improve the detection performance of the network. However, there are still some problems. For example, the problem of uneven category distribution in the VisDrone-2019 dataset affects the feature learning ability of the model for the target categories with little data to a certain extent. In the future, we will explore unsupervised or semi-supervised model training methods to reduce the impact of data volume on model training performance and improve the detection performance of the algorithm.

Use of AI tools declaration

The authors declare we have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61971339), the Major Research and Development Program of Shaanxi Province (No. 2019GY-113) and the Natural Science Basic Research Program of Shaanxi Province (No. 2019JQ-361).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. D. Christine, A. P. S. Chen, H. J. Christanto, Deep learning for highly accurate hand recognition based on YOLOv7 model, *Big Data Cogn. Comput.*, **7** (2023), 53. https://doi.org/10.3390/bdcc7010053
2. Y. Zhang, J. Chu, L. Leng, J. Miao, Mask-Refined R-CNN: A network for refining object details in instance segmentation, *Sensors*, **20** (2020), 1010. https://doi.org/10.3390/s20041010

3. M. Rostami, S. Forouzandeh, K. Berahmand, M. Soltani, Integration of multi-objective PSO based feature selection and node centrality for medical datasets, *Genomics*, **112** (2020), 3943–3950. https://doi.org/10.1016/j.ygeno.2020.07.027
4. L. A. Varga, B. Kiefer, M. Messmer, A. Zell, SeaDronesSee: A maritime benchmark for detecting humans in open water, in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, (2022), 3686–3696. https://doi.org/10.1109/WACV51458.2022.00374
5. W. Li, J. Qiang, X. Li, P. Guan, Y. Du, UAV image small object detection based on composite backbone network, *Mobile Inf. Syst.*, **2022** (2022), 11. https://doi.org/10.1155/2022/7319529
6. Y. Cheng, H. Xu, Y. Liu, Robust small object detection on the water surface through fusion of camera and millimeter wave radar, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 15243–15252. https://doi.org/10.1109/ICCV48922.2021.01498
7. J. Ding, N. Xue, G. S. Xia, X. Bai, W. Yang, M. Y. Yang, et al., Object detection in aerial images: A large-scale benchmark and challenges, *IEEE Trans. Pattern Anal. Mach. Intell.*, **44** (2022), 7778–7796. https://doi.org/10.1109/TPAMI.2021.3117983
8. 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.*, **36** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
9. 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
10. M. Liu, X. Wang, A. Zhou, X. Fu, Y. Ma, C. Piao, UAV-YOLO: Small object detection on unmanned aerial vehicle perspective, *Sensors*, **20** (2020), 2238. https://doi.org/10.3390/s20082238
11. X. Liang, J. Zhang, L. Zhuo, Y. Li, Q. Tian, Small object detection in unmanned aerial vehicle images using feature fusion and scaling-based single shot detector with spatial context analysis, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2020), 1758–1770. https://doi.org/10.1109/TCSVT.2019.2905881
12. X. Liu, J. Huang, T. Yang, Q. Wang, Improved small object detection for UAV acquisition based on CenterNet, *Comput. Eng. Appl.*, **58** (2022), 96–104.
13. Y. Huang, H. Cui, J. Ma, Y. Hao, Research on an aerial object detection algorithm based on improved YOLOv5, in *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, (2022), 396–400. https://doi.org/10.1109/CVIDLICCEA56201.2022.9825196
14. G. Xu, G. Mao, Aerial image object detection of UAV based on multi-level feature fusion, *J. Front. Comput. Sci. Technol.*, **17** (2023), 635–645. https://doi.org/10.3778/j.issn.1673-9418.2205114
15. Z. Liu, X. Zhang, C. Liu, H. Wang, C. Sun, B. Li, et al., RelationRS: Relationship representation network for object detection in aerial images, *Remote Sens.*, **14** (2022), 1862. https://doi.org/10.3390/rs14081862
16. J. Chu, Z. Guo, L. Leng, Object detection based on multi-layer convolution feature fusion and online hard example mining, *IEEE Access*, **6** (2018), 19959–19967. https://doi.org/10.1109/ACCESS.2018.2815149

17. R. Sheikhpour, K. Berahmand, S. Forouzandeh, Hessian-based semi-supervised feature selection using generalized uncorrelated constraint, *Knowledge-Based Syst.*, **269**, (2023), 110521. https://doi.org/10.1016/j.knosys.2023.110521
18. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 936–944. https://doi.org/10.1109/CVPR.2017.106
19. S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
20. M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079
21. G. Jocher, A. Chaurasia, *New YOLOv5 Classification Models*, 2022. Available from: https://github.com/ultralytics/yolov5/tree/v6.2.
22. S. Liu, D. Huang, Y. Wang, Learning spatial fusion for single-shot object detection, preprint, arXiv:1911.09516.
23. 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
24. 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
25. D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, et al., VisDrone-DET2019: The vision meets drone object detection in image challenge results, in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, (2019), 213–226. https://doi.org/10.1109/ICCVW.2019.00030
26. T. Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, Microsoft COCO: Common objects in context, in *13th European Conference on Computer Vision*, (2014), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
27. Z. Zhang, H. Yi, J. Zheng, Focusing on small objects detector in aerial images, *Acta Electron. Sin.*, **51** (2023), 944–955. https://doi.org/10.12263/DZXB.20220313
28. M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in *International Conference on Machine Learning*, PMLR, (2019), 6105–6114.
29. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, MobileNets: efficient convolutional neural networks for mobile vision applications, preprint, arXiv:1704.04861.
30. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint, arXiv:1804.02767.
31. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding YOLO series in 2021, preprint, arXiv:2107.08430.
32. G. Yu, Q. Chang, W. Lv, C. Xu, C. Cui, W. Ji, et al., PP-PicoDet: A better real-time object detector on mobile devices, preprint, arXiv:2111.00902.
33. C. Y. Wang, A. Bochkovskiy, H. Y. M. 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*, (2023), 7464–7475.

34. S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, et al., PP-YOLOE: An evolved version of YOLO, preprint, arXiv:2203.16250.

Appendix

Table A1. The network structure and parameters of CMF-YOLOv5s.

Layer	Module	Parameter		
		Number	Input	Output
0	Conv_1 × 1	×1	640 × 640	32 × 32
1	CBS	×1	32 × 32	64 × 64
2	BottleneckC3	×1	64 × 64	64 × 64
3	CBS	×1	64 × 64	128 × 128
4	BottleneckC3	×2	128 × 128	128 × 128
5	CBS	×1	128 × 128	256 × 256
6	BottleneckC3	×3	256 × 256	256 × 256
7	CBS	×1	256 × 256	512 × 512
8	BottleneckC3	×1	512 × 512	512 × 512
9	SPPF	×1	512 × 512	512 × 512
10	Conv_1 × 1	×1	512 × 512	256 × 256
11	Upsample	×1	None	None
12	Concat	×1	None	None
13	BottleneckC3	×1	512 × 512	256 × 256
14	Conv_1 × 1	×1	256 × 256	128 × 128
15	Upsample	×1	None	None
16	Concat	×1	None	None
17	BottleneckC3	×1	256 × 256	128 × 128
18	Conv_1 × 1	×1	128 × 128	64 × 64
19	Upsample	×1	None	None
20	Concat	×1	None	None
21	MFF	×1	128 × 128	64 × 64
22	CBS	×1	64 × 64	64 × 64
23	Concat	×1	None	None
24	MFF	×1	256 × 256	128 × 128
25	CBS	×1	128 × 128	128 × 128
26	Concat	×1	None	None
27	MFF	×1	512 × 512	256 × 256
28	CBS	×1	256 × 256	256 × 256
29	Concat	×1	None	None
30	MFF	×1	512 × 512	512 × 512