scientific reports

OPEN

A multi-scale small object detection algorithm SMA-YOLO for UAV remote sensing images

Shilong Zhou[1], Haijin Zhou[1]✉ & Lei Qian[1,2]

Detecting small objects in complex remote sensing environments presents significant challenges, including insufficient extraction of local spatial information, rigid feature fusion, and limited global feature representation. In addition, improving model performance requires a delicate balance between improving accuracy and managing computational complexity. To address these challenges, we propose the SMA-YOLO algorithm. First, we introduce the Non-Semantic Sparse Attention (NSSA) mechanism in the backbone network, which efficiently extracts non-semantic features related to the task, thus improving the model's sensitivity to small objects. In the model's throat, we design a Bidirectional Multi-Branch Auxiliary Feature Pyramid Network (BIMA-FPN), which integrates high-level semantic information with low-level spatial details, improving small object detection while expanding multi-scale receptive fields. Finally, we incorporate a Channel-Space Feature Fusion Adaptive Head (CSFA-Head), which fully handles multi-scale features and adaptively handles consistency problems of different scales, further improving the robustness of the model in complex scenarios. Experimental results on the VisDrone2019 dataset show that SMA-YOLO achieves a 13% improvement in mAP compared to the baseline model, demonstrating exceptional adaptability in small object detection tasks for remote sensing imagery. These results provide valuable insights and new approaches to further advance research in this area.

Keywords Remote sensing images, Object detection, Multi-branch auxiliary, Feature fusion

With the rapid advancement of drone and other remote sensing technology, drones have become an ideal platform for various object detection tasks due to their excellent payload capacity, ease of operation and flexible manoeuvrability[1]. Drones are widely used in scenarios such as traffic patrols[2], environmental monitoring[3] and maritime search and rescue[4], particularly in the areas of crowd and vehicle safety monitoring[5]. However, because remote sensing platforms typically operate in the high spatial domain, the targets in the collected images are often very small and often occupy a very small area of the image, resulting in sparse pixel information and insufficient target detail. At the same time, the complex background, variable lighting conditions, ambient noise and occlusion between targets further aggravate the difficulty of small target detection. Therefore, small object detection[6] not only has to solve the problem of low resolution and limited feature information, but also has to deal with the challenges posed by complex environmental clutter and occlusion. These factors combine to make small object detection an extremely challenging task, and there is an urgent need for specially designed detection algorithms to improve the accuracy and robustness of small object detection.

With the rapid advancement of deep learning, the performance of object detection models has improved significantly. Currently, mainstream object detection algorithms can be divided into two types: two-stage algorithms, and one-stage algorithms. Two-stage detection algorithms typically generate candidate regions first, and then classify and regress these regions. Representative models include the R-CNN series, such as Faster R-CNN[7], Mask R-CNN[8], Cascade R-CNN[9] and R-FCN[10]. These algorithms are known for their high accuracy, particularly in detecting small objects, and their performance can be further enhanced by techniques such as multi-scale feature fusion and model distillation. However, their main drawback is slow processing speed, as they require the generation of numerous candidate regions, leading to higher computational and time complexity. Single-stage object detection algorithms treat object detection as an overall regression problem, directly outputting object categories and location information. Representative methods include anchor-based algorithms such as the YOLO (You Only Look Once) series[11–15], SSD (Single Shot MultiBox Detector)[16] and RetinaNet[17]. These algorithms divide the image into anchor grids and perform classification and regression on

[1]Key Laboratory of Environmental Optics and Technology, Anhui Institute of Optics and Fine Mechanics, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China. [2]University of Science and Technology of China, Hefei 230026, China. ✉email: hjzhou@aiofm.ac.cn

| https://doi.org/10.1038/s41598-025-92344-7

each anchor. Among them, YOLO methods have gained significant attention in recent years due to their fast inference speed and commendable detection accuracy. However, while these methods have become mainstream solutions in the field of object detection, the detection of small objects in drone datasets remains highly complex. The small size of these objects, coupled with limited visual data, often results in these entities being inadvertently missed during detection.

To address the challenges of small object detection, many researchers have proposed various improvement strategies. For example, Liu et al.[18] reduced the inference time by using re-parameterised detection heads and designed a graded loss function to improve the accuracy of small object detection, but their method may not perform well when dealing with high-density targets. Zhu et al.[19] introduced the Transformer Prediction Head (TPH) and Convolutional Block Attention Module (CBAM)[20] in YOLOv5, which optimises cross-scale object discrimination and effectively identifies key regions in dense object scenes, although it requires high computational resources. Zhao et al.[21] proposed MS-YOLOv7, which extends YOLOv7 by adding a fourth detection head to extract features at different scales, although this significantly increases the parameter size of the model. In addition, they enhanced the network's neck features with Swin transformers, W-MSA, SW-MSA[22] and CBAM attention mechanisms, and adopted SoftNMS and the Mish activation function to significantly improve the handling of occluded targets in dense target detection. Zhang et al.[23] introduced Drone-YOLO, which uses a three-layer PAFPN structure and detection heads specifically designed for small targets, significantly improving the small object detection capabilities. The model adopts a sandwich fusion module with a large receptive field and low model parameters, and enhances the backbone downsampling layers with the RepVGG reparameterisation convolution module, increasing the efficiency of multi-scale feature learning. However, the computational efficiency of the model may decrease when dealing with large or dense target scenes. Wang et al.[24] UAV-YOLOv8 integrates the BiFormer attention mechanism to optimise the backbone network and designs a feature processing module called Focal FasterNet Block (FFNB), while introducing two new detection scales to effectively merge shallow and deep feature information. However, this may lead to information redundancy between targets of different scales. Qi et al.[25] proposed MSFE-YOLO, which extends the symmetric feature extraction branches to build the symmetric C2f (SCF) module, thus improving the feature extraction capabilities in both the backbone and neck networks. They also used the Efficient Multi-Scale Attention (EMA) module for cross-channel and cross-space learning, improving the relevance of local features. However, this increased the model's computational load, which could affect real-time detection performance.

Despite these advances, challenges remain due to the complex backgrounds, significant spatial resolution differences, and irregular arrangement of small objects in remote sensing imagery. To further improve the object detection performance in remote sensing images, this paper proposes an improved object detection algorithm with the following improvements:

- Proposed a Non-Semantic Sparse Attention (NSSA) mechanism that suppresses the expression of semantic information in sparse attention blocks and adaptively extracts non-semantic features relevant to operations. This mechanism enhances the sensitivity to small objects while maintaining a low number of parameters.
- Introduced a new small object detection head tailored for aerial imagery and proposed a bidirectional multi-branch auxiliary feature pyramid network. This network employs an SDF (Semantic-Detail Fusion) method to combine structural information from shallow features with semantic information from deep features, further improving the model's small object detection capability and effectively expanding multi-scale receptive fields.
- Addressing the inconsistency of features across different scales in aerial images, the study introduces a CS-FA-Head. By making full use of multi-scale features and adaptively handling the consistency problem of different scales, the network can effectively filter the conflict problem so that the network can more accurately identify and locate objects of different scales.
- Experimental results demonstrate that SMA-YOLO effectively enhances feature extraction and fusion capabilities, significantly improving the ability to detect small objects in remote sensing imagery. It achieves a notable increase in detection accuracy while maintaining a low number of model parameters.

Related work

Attention mechanisms

Attention is a technology that mimics the human cognitive mechanism of selectively focusing on specific information and enhancing critical details to better understand the essence of data. By introducing attention mechanisms[26], deep learning models have significantly improved their performance, especially in the field of natural language processing (NLP). Transformers[27] and their variants have demonstrated unprecedented performance on several NLP benchmark tasks, far outperforming earlier models such as recurrent neural networks (RNNs)[28] and convolutional neural networks (CNNs)[29]. In computer vision, the Squeeze and Excitation (SE)[30] module enhances representational capability by dynamically adjusting channel characteristics. This module uses global average pooling to compress the features of each channel into a single value, which is then non-linearly transformed through a fully connected layer. The resulting weight vector is multiplied by the input channel features on a per-element basis, emphasising critical features. To further optimize computational efficiency, the Efficient Channel Attention (ECA)[31] module replaces the fully connected layer with one-dimensional convolution, reducing parameter redundancy while effectively capturing dependencies between channels. However, despite the strong performance of SE and ECA modules in channel attention, they fail to adequately consider spatial information. To address this limitation, the Convolutional Block Attention Module (CBAM) combines channel and spatial attention by leveraging large-kernel convolutions to aggregate local positional information, thereby enhancing feature representation. However, single-layer large-kernel convolutions can only capture limited local positional dependencies and fail to reflect global positional information adequately.

To solve this problem, the Coordinate Attention (CA)[32] module introduces positional information into channel attention, enabling precise capture of global positional dependencies. Integrating the CA module into CSPNet[33] can significantly improve the detection accuracy of aerial targets while keeping computational costs manageable. In addition, the SE, CBAM and CA modules all enhance semantic features by focusing on channel and spatial relationships. Vision Transformers (ViTs)[34], as a novel model architecture[35,36], process images as a series of patches, successfully bridging the gap between image classification tasks and Transformer structures. The ViT network has achieved state-of-the-art accuracy on the ImageNet classification task, with its core component solely comprising Multi-Head Self-Attention (MHSA), showcasing a robust feature modeling capability.

Multi-scale features for object detection

The construction of an efficient feature pyramid network is crucial for improving the efficiency and accuracy of multi-scale object detection[37]. Deep networks have a large receptive field and strong semantic representation capabilities, but suffer from low resolution feature maps, resulting in weaker representation of spatial geometric information. In contrast, shallow networks offer higher resolution and capture more spatial geometric details, but have limited semantic representation capabilities. Thus, the integration of multi-scale features from deep and shallow networks can achieve comprehensive information extraction that includes both global information and local details. FPN[38] achieves this by employing a top-down architecture combined with lateral connections, effectively injecting high-level semantic information into lower-level features. This facilitates the exchange of information between feature maps of different scales, significantly reducing the loss of semantic information caused by stacked convolutions and pooling operations, thereby greatly improving object recognition accuracy. NAS-FPN[39] optimises the construction of the feature pyramid by autonomously searching for network architectures, thereby improving object recognition performance. The Adaptive Spatial Feature Fusion (ASFF)[40] mechanism dynamically adjusts the feature weights for each position and adaptively refines the fusion strategy based on the position and size of the objects. This approach effectively improves the robustness of the model in complex scenes and under object occlusion. BiFPN[41] further optimises feature fusion by allowing features to flow bidirectionally - top-down and bottom-up - and assigning weights to each input feature to optimise the fusion process. This design better adapts to different input resolutions and object sizes, improving recognition performance. In addition, PRB-FPN (Parallel Bidirectional FPN)[42] introduces a parallel bidirectional feature fusion structure to retain high-quality features for precise localisation. However, during layer-by-layer feature extraction and spatial transformation, these basic building blocks can still result in significant information loss, especially when detecting small objects. To address this issue, further optimisation of feature fusion strategies is needed to improve the capability of feature pyramid networks to detect small objects and handle complex scenarios. The introduction of positionally enhanced attention mechanisms or more efficient feature flow strategies can further improve detection performance, especially in small object recognition tasks.

Methods

YOLOv8 consists of three main components: the backbone, the neck and the sensing head. It comes in five versions: YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x, which differ in channel width, depth and maximum number of channels. The backbone network extracts features from the input image through multiple convolutional operations, generating three detection heads at different scales $(80 \times 80, 40 \times 40, \text{and } 20 \times 20)$. The backbone uses the Cross Stage Partial Darknet (CSPdarknet) structure, replacing the original Cross Stage Partial (CSP) module with the C2f module to improve gradient flow. In addition, a Spatial Pyramid Pooling Fast (SPPF) module is used at the end of the backbone to pool the feature maps into a fixed size to accommodate different output dimensions. The neck uses a PAN-FPN structure, which builds a network with both top-down and bottom-up pathways to effectively fuse multi-scale information. The detection head features a decoupled design, with two independent branches for target classification and bounding box regression prediction. Different loss functions are employed for each task: Binary Cross Entropy (BCE) loss for classification, and Distribution Focal Loss (DFL Loss)[43] and Complete IOU Loss (CIOU Loss)[44] for regression. To address the challenges of detecting small and multi-scale objects in YOLOv8, we propose an optimised detection model, SMA-YOLO, based on YOLOv8n, tailored to the specific needs of remote sensing imagery. The detailed network structure is shown in Fig. 1, with the individual modules of the improvements outlined below.

Non-semantic sparse attention mechanism

Inspired by ViT, the attention mechanism allows the model to focus on the relationships between different locations (or patches) in the image, thereby capturing spatial dependencies within the image. However, this mechanism can also lead the model to overemphasise semantic information. Specifically, during global interactions, the model considers features from all regions of the image, such as colour and shape, to gain a comprehensive understanding of the overall content. However, this approach can miss the local inconsistencies of non-semantic information, making it difficult for the model to effectively handle fine details or complex local variations. In order to improve the model's sensitivity to small objects and to efficiently extract and manipulate task-relevant non-semantic features, particular attention is paid to local geometric and textural patterns. We introduce a NSSA mechanism into the backbone network as shown in Fig. 2.

Given an input feature map $X \in \mathbb{R}^{H \times W \times C}$, it is divided into two branches. One branch undergoes convolution, mean pooling and sigmoid operations to generate attention weights, while the other branch divides the original feature map into tensor blocks of the form $(C, S \times S, H/S \times W/S)$, Where S is the sparse coefficient. That is, the feature map is decomposed into non-overlapping tensor blocks of size $S \times S$, with each block having spatial dimensions of $H/S \times W/S$, and the self-attention mechanism is computed separately within these tensor blocks. Only tensor blocks labelled with the same colour perform self-attention computations. This design effectively suppresses the expression of semantic information within the sparse

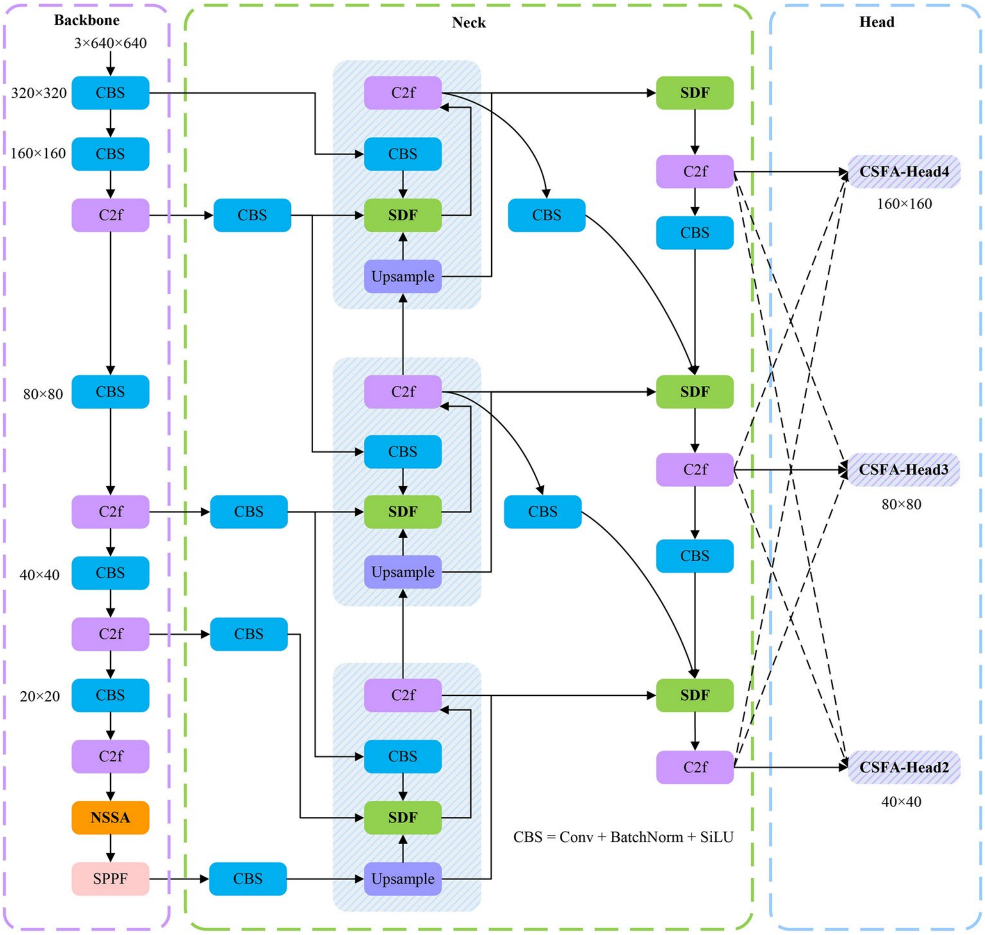


Fig. 1. The overall structure of SMA-YOLO.

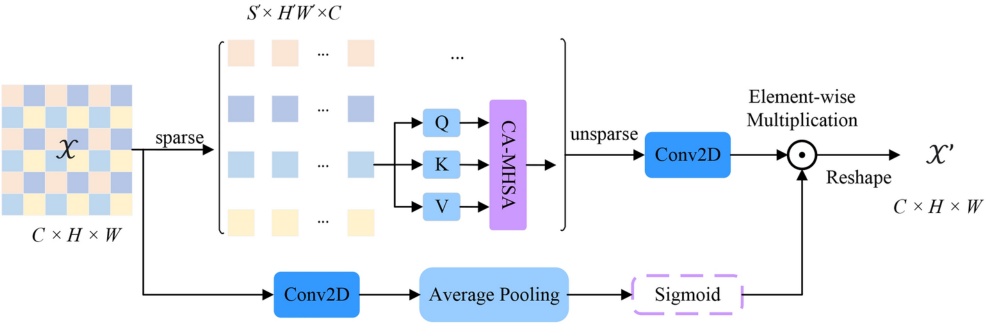


Fig. 2. Structure of NSSA, where S' represents $S \times S$, and H' and W' represent H/S and W/S respectively. The figure illustrates the case when the sparse coefficient S is 2, dividing the input features into 4 non-overlapping tensor blocks of different colors. When the sparse coefficient S is 1, it is equivalent to a single block, performing global attention.

attention blocks, allowing the model to focus more on extracting non-semantic features. The recovered features are then multiplied element-wise by the corresponding elements from previous branch and reshaped back to the original shape to meet the spatial dimensional requirements. The implementation details of NSSA are as follows:

$$F = \delta\left(\text{AP}\left(\text{Conv}(X)\right)\right) \tag{1}$$

$$Q, K, V = \text{Sparse}_Q(X), \text{Sparse}_K(X), \text{Sparse}_V(X) \tag{2}$$

$$F' = \text{Conv}\left(\text{Unsparse}\left(\text{CA-MHSA}(Q, K, V)\right)\right) \tag{3}$$

$$X' = \text{Reshape}\left(F \otimes F'\right) \tag{4}$$

where AP stands for Average Pooling, $\text{Sparse}(\cdot)$ refers to the sparse linear projection to generate queries, keys, and values, CA-MHSA refers to Channel-wise Multi-Head Self-Attention, and Unsparse indicates restoring the shape prior to sparsification.

Sparsification of tensor blocks in the feature map effectively eliminates the need for attentional computations involving many irrelevant key-value pairs during operation localisation, thus improving local representational ability with low parameters, allowing the model to pay more attention to non-semantic information in the image.

Bidirectional multi-branch auxiliary feature pyramid network

The original YOLOv8 network uses the Path Aggregation Network (PAN-FPN) as its neck, which performs bidirectional feature aggregation along both bottom-up and top-down paths. This allows the deeper layers to capture semantic features while the shallower layers respond to image details. This bidirectional information flow design facilitates the fusion of low-level and high-level information, effectively shortening the information path between them. During upsampling and downsampling, features of the same dimension are stacked to preserve the features of small objects. However, for small targets, especially those with pixel-level resolution, the feature maps typically have lower spatial resolution, limiting the geometric information available for accurate object detection and failing to fully satisfy the demand for positional information. On the other hand, the feature reuse rate is low, and after long paths of upsampling and downsampling, some information in the original features is easily lost. To address these issues and improve the detection performance of small low-resolution targets and multidimensional feature fusion, we propose the BIMA-FPN feature pyramid network, as shown in the neck part of Fig. 1.

First, we introduce a higher resolution (160 × 160 pixels) detection head, which enhances the model's performance in detecting small objects by utilizing lower-level feature layers. To reduce computational load, we also remove the detection branch for large objects. Additionally, we incorporate six CBS modules with a stride of 1 and a kernel size of 1, which serve as containers to store the backbone feature information. These containers provide a new input stream of backbone feature information to subsequent network layers, effectively aggregating shallow feature maps (low resolution but weak semantic information) and deep feature maps (high resolution but rich semantic information). This aggregation captures multi-scale feature information, addressing issues related to excessive parameter increase, model bloat, gradient vanishing, and feature degradation.

During feature fusion, we also propose a Semantic Detail Fusion module, which aims to effectively capture local details and texture information to further improve detection performance. The structure is shown in Fig. 3.

For feature maps of different sizes, we first represent the feature at each level *i* as f_i^0, then use convolution to align the channels of f_i^0 to *c*, resulting in the feature map denoted as $f_i^1 \in \mathbb{R}^{H_i \times W_i \times c}$, where H_i, W_i, and c represent the width, height, and channels of f_i^1, respectively. Next, we adjust the sizes of the feature maps at each level *j* to match the resolution of f_i^1, formulated as:

$$f_{ij}^2 = \begin{cases} (\mathrm{D})(f_j^1, (H_i, W_i)) & \text{if } j < i, \\ (\mathrm{C})(f_j^1) & \text{if } j = i, \\ (\mathrm{U})(f_j^1, (H_i, W_i)) & \text{if } j > i, \end{cases} \tag{5}$$

where D represents adaptive average pooling, C represents convolution, and U represents bilinearly interpolating f_j^1 to the resolution (H_i, W_i), with $1 \leq i, j \leq N$.

Next, the feature enhancement process can be summarized as follows:

$$F = \text{Concat}(f_{i1}^2, \ldots, f_{iN}^2) \tag{6}$$

$$\hat{F} = F \oplus (F \otimes \delta(\text{Pw-Conv}(GAP(F)))) \tag{7}$$

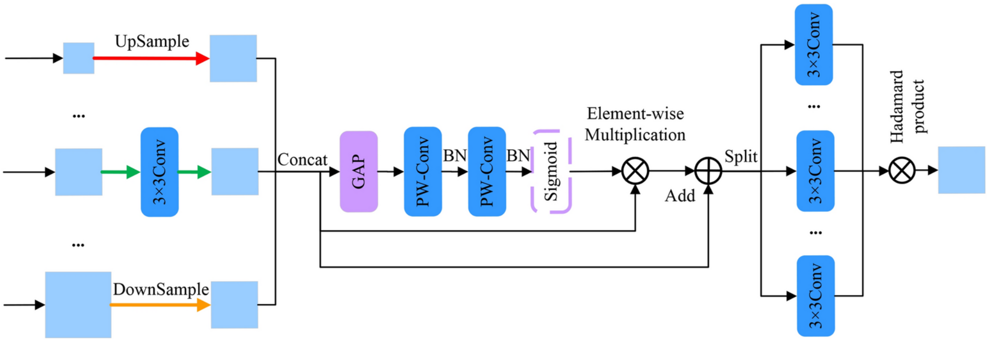


Fig. 3. Structure of semantic and detail fusion modules.

where $\oplus$ denotes element-wise summation, $\otimes$ denotes element-wise multiplication, $\mathrm{Pw\text{-}Conv}(\cdot)$ represents a pointwise convolution layer, and $\delta(\cdot)$ and $GAP(\cdot)$ represent an S-shaped function and global average pooling, respectively. After that, a splitting operation is performed:

$$f_{i1}^3, \ldots, f_{iN}^3 = \mathrm{Split}(\hat{F}) \tag{8}$$

In the formula 9, θ_{ij} represents the parameters of the smooth convolution, and f_{ij}^4 is the *j*-th smoothed feature map at the *i*-th level, formulated as:

$$f_{ij}^4 = \theta_{ij}(f_{ij}^3) \tag{9}$$

Finally, we apply the element-wise Hadamard product to all the resized feature maps to enhance the *i*-th level features with both more semantic information and finer details, as:

$$f_i^5 = H([f_{i1}^4, f_{i2}^4, \ldots, f_{iN}^4]) \tag{10}$$

where $H(\cdot)$ denotes the Hadamard product operation.

BIMA-FPN integrates top-down feature propagation from high-level features and bottom-up feature propagation from low-level features, enabling bidirectional information flow. This effectively addresses the issue of contextual information loss in the model, leading to a significant improvement in small object detection accuracy.

Channel-space feature fusion adaptive head

To further improve the scale invariance and object detection capability of features, we have improved the detection head of YOLOv8. This study proposes a mechanism called CSFA-Head, this mechanism is inspired by the ASFF strategy and dynamically learns the fusion weights of each scale feature map, adjusts the different scale feature layers to a unified size, and adaptively fuses them, as shown in the Fig. 4.

We define $x^{n \to l}$ as the scaled feature vector, *n*, *l* denote the indices of the source and target layers, respectively, which is made uniform in size by convolution operations. To ensure that the original layer features have a larger weight in the detection head, we introduce α_s^l, β_s^l, and $\gamma_s^l \in [0,1]$ as the learned weights[40]. α_s^l, β_s^l, and γ_s^l are defined using the softmax function with λ_α^l, λ_β^l, and λ_γ^l as control parameters, respectively. The formulas for the weight calculation are as follows:

$$\alpha_s^l = \frac{e^{\lambda_\alpha^l}}{e^{\lambda_\alpha^l} + e^{\lambda_\beta^l} + e^{\lambda_\gamma^l}}, \beta_s^l = \frac{e^{\lambda_\beta^l}}{e^{\lambda_\alpha^l} + e^{\lambda_\beta^l} + e^{\lambda_\gamma^l}}, \gamma_s^l = \frac{e^{\lambda_\gamma^l}}{e^{\lambda_\alpha^l} + e^{\lambda_\beta^l} + e^{\lambda_\gamma^l}}, \tag{11}$$

ensuring that

$$\alpha_s^l + \beta_s^l + \gamma_s^l = 1. \tag{12}$$

To adjust the importance of channels and suppress irrelevant information, we use channel attention maps to obtain another set of weight parameters, thereby improving model performance. The specific operations are as follows:

$$s(i) = \mathrm{DW} - \mathrm{Conv}(\mathrm{MaxPool}(f(i)) + \mathrm{AvgPool}(f(i))), \tag{13}$$

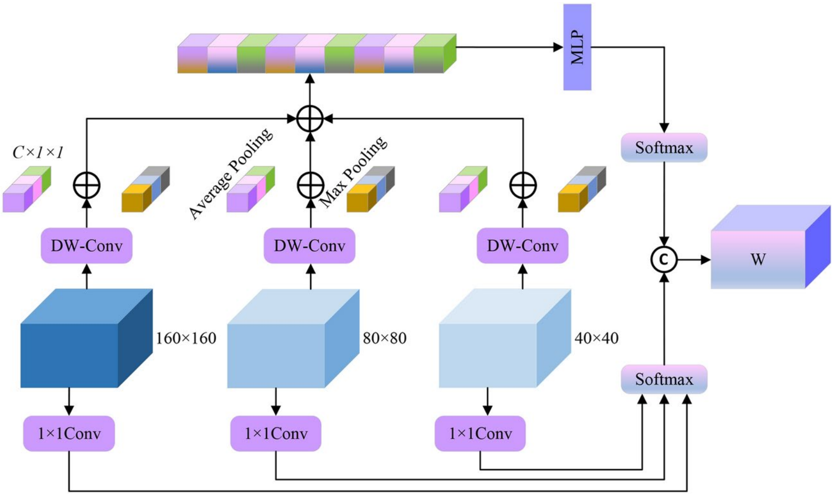


Fig. 4. The Dynamic Fusion Weights Mechanism of CSFA-Head.

$$c(i) = \mathrm{MLP}(s(i)), \tag{14}$$

where the symbol $\mathrm{FC}(\cdot)$ represents the module of a fully-connected (FC) layer, and $\delta(\cdot)$ represents the ReLU function. By applying the softmax function to $c(2)$, $c(3)$, and $c(4)$, we obtain three weights α_c^l, β_c^l, and γ_c^l. The final weighting parameters α^l are calculated as follows:

$$\alpha^l = w_1\alpha_c^l + w_2\alpha_s^l, \tag{15}$$

with w_1 and w_2 being learnable parameters, the computation for β^l and γ^l follows a similar approach.

Using these learned weights for weighted fusion, we obtain a new multiscale feature mapping tensor:

$$y^l = \alpha^l x^{P2\to l} + \beta^l x^{P3\to l} + \gamma^l x^{P4\to l}. \tag{16}$$

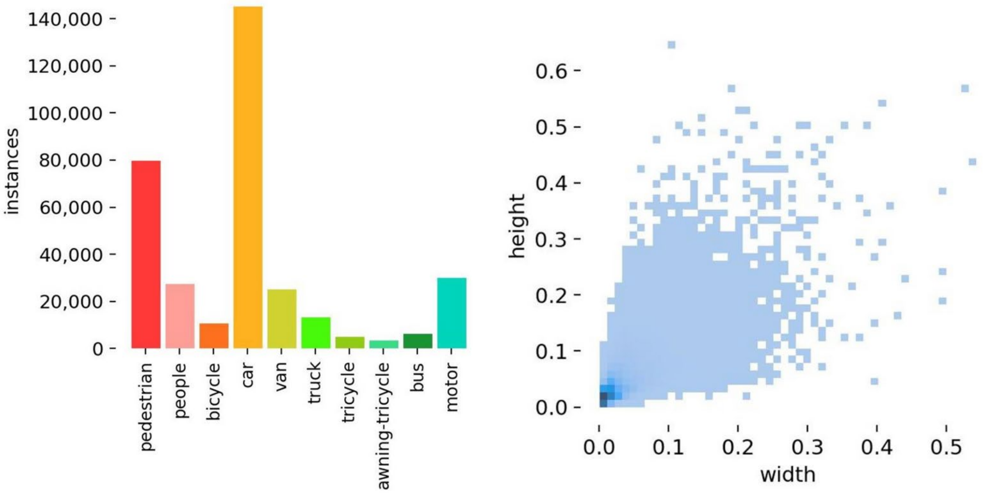
This weighted fusion method not only effectively preserves the advantages of features at different scales but also enhances the model's ability to extract channel information, thereby improving overall detection performance.

Experiments

Datasets and implementation details

We use the VisDrone2019 dataset[45] to evaluate the performance of the model. This dataset contains real-world images captured from drones at various angles and tasks, with a wide range of scale variations and rich small object features, as shown in Fig. 5. The dataset covers 10 categories, including pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. The training set contains 6,471 images, the validation set contains 548 images, and the test set contains 1,610 images. Notably, the dataset exhibits class imbalance and contains small objects. According to the COCO standard[46], objects are classified into three sizes: small, medium and large. Specifically, objects with a bounding box area smaller than 32×32 pixels are categorised as small objects, objects with an area between 32×32 and 96×96 pixels are categorised as medium objects, and objects larger than 96×96 pixels are categorised as large objects. Table 1 shows the distribution of small, medium and large objects within each category. It can be seen that small objects account for 60.49% of the targets in the VisDrone2019 dataset. The high proportion of small objects and complex scenes make this dataset an ideal choice for evaluating small object detection performance.

Our baseline model is YOLOv8, version Ultralytics 8.0.225. In terms of hardware, we used an Intel(R) Xeon(R) Gold 6330 CPU with 28 cores and 56 threads, clocked at 2.0 GHz, and equipped with 32 GB of RAM. Additionally, we employed a GeForce RTX 4090 GPU with 24 GB of video memory. The deep learning framework used was Python 3.9, PyTorch 1.13.1, and CUDA 11.7. For the experiments, we set the batch size to 8 and trained for 200 epochs. We chose stochastic gradient descent (SGD) as the optimizer with an initial learning rate of 0.01, a learning rate momentum of 0.937, and a weight decay coefficient of 0.0005. All input images were standardized to 640×640 pixels and no pretrained weights were used for training. To ensure fairness in comparison and ablation experiments, the same experimental environment and hyperparameters were applied throughout the training and testing phases.



(a) (b)

Fig. 5. The distribution of the objects in the dataset: (**a**) distribution of the number of classes; (**b**) the width and height distribution of the target; the concentration of the distribution is indicated by the color gradient from light white to dark blue, indicating that the distribution becomes more and more concentrated.

Category	Small		Medium		Large	
	Number	Proportion (%)	Number	Proportion (%)	Number	Proportion (%)
Pedestrian	65227	82.22	13804	17.40	306	0.39
People	23497	86.84	3463	12.80	99	0.37
Bicycle	7111	67.85	3244	30.95	125	1.19
Car	69862	48.22	63079	43.54	11926	8.23
Van	10751	43.08	11517	46.15	2688	10.77
Truck	3982	30.93	6725	52.23	2168	16.84
Tricycle	2176	45.22	2425	50.39	211	4.38
Awning Tricycle	1360	41.90	1699	52.34	187	5.76
Bus	1661	28.03	3265	55.10	1000	16.87
Motor	21986	74.16	7395	25.00	266	0.90
Total	207613	60.49	116616	33.98	18976	5.53

Table 1. Distribution of target sizes in the VisDrone2019 dataset.

Evaluation metrics

This paper primarily evaluates the model performance using precision (P), recall (R), mAP@0.5, mAP@0.5:0.95, AP_{small}, model parameters, GFLOPs and FPS. Precision is defined as the ratio of the number of correctly detected objects to the total number of predicted objects. Recall is the ratio of the number of correctly detected objects to the total number of actual objects. AP (Average Precision) evaluates the performance of object detection models for a particular class. It is computed as the area under the Precision-Recall (P-R) curve, with recall plotted on the x-axis and precision on the y-axis:

$$AP = \int_0^1 P(R)\,dR \tag{17}$$

mAP (mean Average Precision) signifies the average precision across all categories, serving as a comprehensive performance metric for multi-class detection tasks:

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{18}$$

mAP@0.5 and mAP@0.5:0.95 represent the mean AP for each class when the Intersection over Union (IoU) threshold is 0.5 and when IoU ranges from 0.5 to 0.95, respectively, where IoU is the overlap rate between the predicted bounding box and the true bounding box, i.e. the ratio of intersection and union, which is used to evaluate the accuracy of object detection. AP_{small} refers to the Average Precision for small objects with bounding box areas less than 32×32 pixels. The model parameter size (M) refers to the total number of parameters in millions. Computational cost is measured using floating-point operations per second (FLOPs), which indicates the number of floating-point operations performed per second. The formula for calculating FLOPs for convolution operations is:

$$\text{FLOPs} = 2HW(C_i K^2 + 1)C_o \tag{19}$$

where H and W are the height and width of the input feature map, C_i is the number of input channels, K is the size of the convolution kernel, and C_o is the number of output channels. The total FLOPs of the model is the sum of the computational cost of all convolution layers. The computational unit used in this paper is GFLOPs, representing the number of floating-point operations per second in billions. FPS is used to measure the number of frames processed per second.

Ablation experiment

To evaluate the impact of adding and modifying modules on the performance of the baseline model, we decompose the structure of each module and perform comprehensive ablation experiments for evaluation and comparison. We choose YOLOv8n as the baseline model, and train and test it under the same experimental conditions using the VisDrone2019 dataset and consistent training parameters. The results of these experiments are detailed in Table 2.

The experimental results presented in the table show that each enhancement strategy applied to the baseline model improved recognition performance to varying degrees. Specifically, the introduction of the non-semantic sparse attention mechanism led to a 1.3% increase in mAP@0.5, improving the model's local representation ability and allowing it to focus more on non-semantic information in the image. Following the addition of BIMA-FPN, both mAP@0.5 and AP_{small} showed significant improvements. This was attributed to the addition of a small object detection head, a bidirectional multi-branch auxiliary feature pyramid network, and a semantic detail fusion module. To better illustrate the contributions of the SDF module and the higher resolution (160×160 pixels) detection head, a detailed analysis is provided, as shown in the Table 3. These components

Model	NSSA	BIMA-FPN	CSFA-Head	P	R	mAP@0.5	mAP@0.5:0.95	$\mathrm{AP}_{\mathrm{small}}$	GFLOPs	Parameters(M)
1				0.437	0.329	0.329	0.191	0.089	8.1	3
2	✓			0.449	0.338	0.342	0.203	0.096	9.2	3.9
3		✓		0.547	0.439	0.441	0.266	0.141	20.8	2.2
4			✓	0.468	0.357	0.358	0.214	0.118	10.4	4.4
5	✓	✓		0.552	0.435	0.446	0.269	0.147	21.4	3
6	✓		✓	0.473	0.365	0.378	0.225	0.124	11.4	5.3
7		✓	✓	0.566	0.438	0.45	0.274	0.153	24.3	2.5
8	✓	✓	✓	0.573	0.447	0.459	0.282	0.162	24.9	3.3

Table 2. Ablation experimental data shows that each result obtained is optimal. We use YOLOv8n as the baseline model.

Model	mAP@0.5	mAP@0.5:0.95	GFLOPs	Parameters(M)
BIMFPN w/o SDF	0.433	0.257	21.7	2.3
BIMFPN w/o 160 × 160	0.346	0.21	8.5	2.2
BIMFPN	0.441	0.266	20.8	2.2

Table 3. In the ablation study of BIMFPN, "BIMFPN w/o SDF" indicates the removal of the SDF module, while "BIMFPN w/o 160 × 160" indicates the removal of the 160 × 160 detection head, keeping the original detection head size of the baseline model.

facilitated bidirectional information flow, effectively addressing the issue of contextual information loss in the model, while maintaining a lower parameter count by removing the detection branch for large object detection. In addition, the pyramid network implementation significantly improved object detection accuracy, albeit at the cost of increased computational complexity. Finally, the incorporation of the CSFA head module effectively exploited multi-scale features and adaptively resolved the consistency problem between different feature scales, effectively mitigating conflicts between multi-scale information. Compared to the baseline model, our proposed method achieved a 13% improvement in mAP@0.5 and a 7.3% increase in AP_{small}, significantly improving the accuracy of small object detection. These experimental results highlight the significant improvement in model learning efficiency through careful optimisation at each stage of the algorithm, and validate the effectiveness of each optimisation measure.

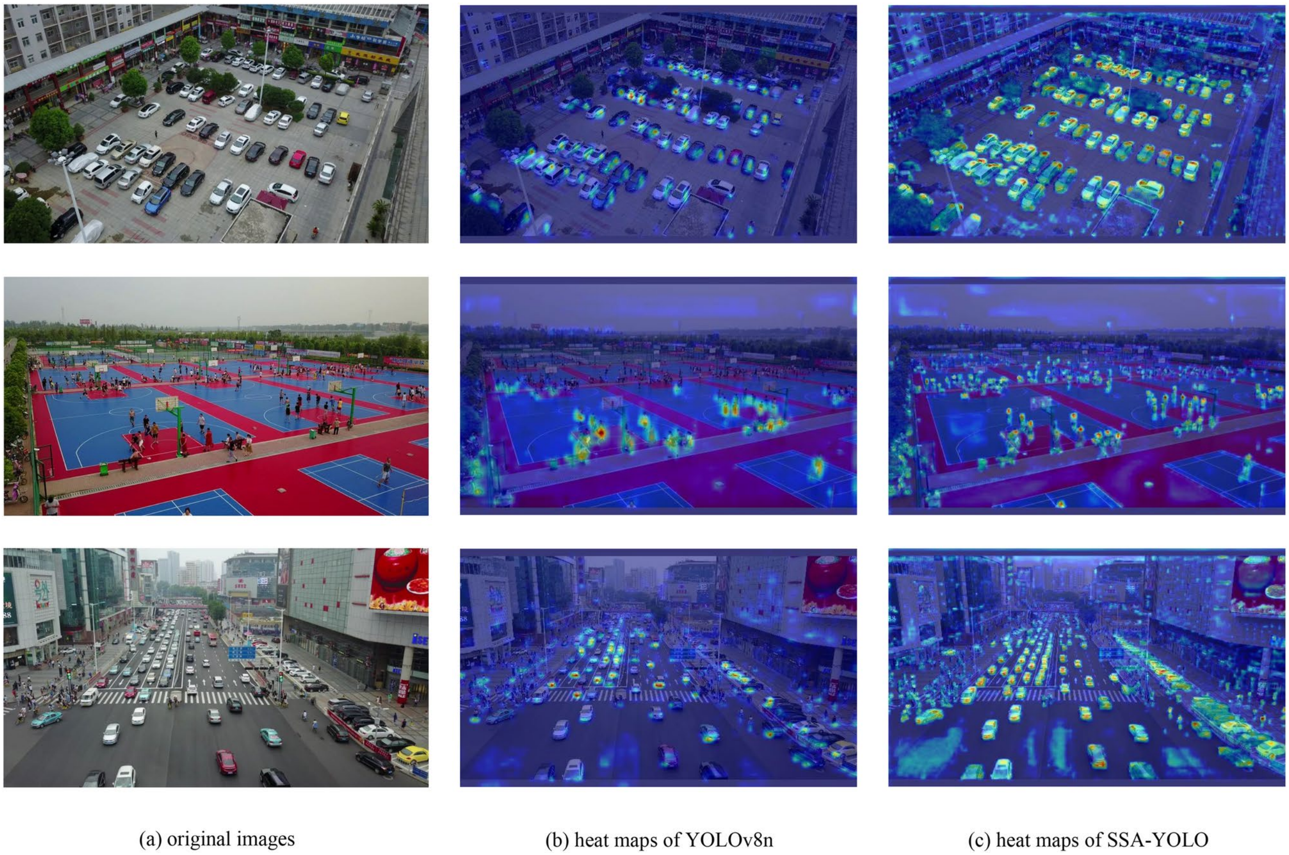
To highlight the regions of interest in the feature map, we use heatmaps for visualization, as shown in Fig. 6. Before optimizing YOLOv8n, the model exhibited insufficient attention to small and very small targets. This was primarily due to the dense distribution of these targets and the complexity of the environment, which resulted in reduced sensitivity to distant objects. After optimization, the model's attention to these targets improved significantly, enhancing its ability to detect small objects and mitigating the influence of external background elements, demonstrating a clear contrast in performance.

Comparative experiment

To further validate the performance advantages of the proposed method in remote sensing small object detection, a comparative analysis was conducted using YOLOv8n as the baseline, alongside other existing models. The performance of each model on the VisDrone2019 validation set is shown in the Table 4. The results indicate that our proposed model achieves the best performance in terms of Precision (P), Recall (R), and mAP@0.5. Although GFLOPs have increased compared to the baseline model and FPS does not have certain advantages, our model outperforms other models with higher mAP@0.5, such as Edge-YOLO and PVswin-YOLOv8s, with improvements of 1.1% and 2.6% in mAP@0.5, respectively. Furthermore, in terms of parameter count and GFLOPs, our model still maintains a certain advantage. Our model also shows clear advantages over the Transformer-based end-to-end object detection model, RT-DETR.

The experimental results fully demonstrate the effectiveness of the improved model in UAV object detection. Overall, our model not only achieves high accuracy, but also maintains a low number of parameters and a certain inference speed, which meets the basic requirements of embedded remote sensing platforms. This makes it more suitable for multi-object recognition tasks in high-precision UAV images, providing an innovative solution for this field.

To further verify the robustness of the model, we perform a comparative analysis with other object detection algorithms on the SSDD[56] and RSOD[57] datasets. The SSDD dataset is specifically designed for ship detection in satellite imagery and contains 1,160 high-resolution remotely sensed images with one category: ships, containing 2,456 ship instances. The dataset is divided in a ratio of 8:1:1, with 928 images used for training, 116 for validation and 116 for testing. The RSOD dataset is used for object detection in remote sensing images, including four target categories: aircraft, oil tanks, playgrounds, and overpasses, containing 4,993 aircraft instances, 1,586 oil tank instances, 191 playground instances, and 180 overpass instances. The dataset is also split in the same 8:1:1


(a) original images

(b) heat maps of YOLOv8n

(c) heat maps of SSA-YOLO

Fig. 6. Compare thermal maps generated for different model detection results: (**a**) Original images. (**b**) Heat maps of YOLOv8n. (**c**) Heat maps of SMA-YOLO.

ratio. Both datasets contain small and very small targets, making detection more difficult and prone to false positives and missed detections.

As shown in the Table 5, the proposed SMA-YOLO algorithm achieves the highest precision, recall, mAP@0.5 and mAP@0.5:0.95 on both the SSDD and RSOD datasets compared to Faster R-CNN, YOLOv3-tiny, YOLOv5s, YOLOv6, YOLOv7-tiny, YOLOv8n and YOLOv11n. This further confirms that the SMA-YOLO model introduced in our research is not limited to specific datasets, but demonstrates excellent recognition capability across different remote sensing imagery scenarios.

Discussion and visualization

To vividly demonstrate the detection performance of our method, we utilized visualization techniques to compare the detection results of SMA-YOLO with the baseline method. We selected several representative images from the VisDrone, SSDD, and RSOD datasets as experimental data. The first two images are from the VisDrone dataset, and the last two are from the SSDD and RSOD datasets, respectively. These images cover various lighting conditions, background complexities, and varying densities of small objects, making them ideal for comparative analysis. Figure 7 presents these comparison results.

In the first image, the YOLOv8n model mistakenly detects a negative sample as "pedestrian." For small targets at the pixel level, YOLOv8 struggles to focus on large-scale feature maps, which leads to inadequate positional information and lower detection quality. The improved model effectively avoids this issue. However, in dense target scenarios, the improved model still faces significant challenges, resulting in some false detections. From an aerial perspective, the pixel information for trucks and buses appears similar, which could lead to potential misidentifications. The model also performs weakly in detecting low-resolution and distant targets, especially in scenarios with large-scale background interference. To address these issues, future research will further improve the multi-scale feature fusion and background suppression capabilities of the model, and consider introducing a multi-view fusion strategy[58] to improve the robustness and detection accuracy of the model. In the second image, the improved model shows a significant reduction in false negatives for the 'pedestrian' category, and it can also be seen that the improved SMA-YOLO algorithm can detect the low visibility target better than the baseline model. This demonstrates the effectiveness of our model in detecting small objects even when they are occluded or overlapped. The third and fourth images highlight 'ship' and 'aircraft' targets that the baseline model failed to detect, further demonstrating the benefits of our method. Our algorithm achieves excellent detection accuracy in both complex and simple environments while maintaining low computational cost, which has important practical implications for the widespread adoption of object detection. The results clearly show

Model	P	R	mAP@0.5	Params (M)	GFLOPs	FPS
YOLOv3-tiny	0.39	0.241	0.236	12.1	19.1	**165.9**
YOLOv5m	0.467	0.382	0.373	9.1	24.1	58.6
YOLOv5l	0.516	0.387	0.395	25.1	64.4	64.8
TPH-YOLOv5	0.496	0.4	0.394	41.6	160.1	–
YOLOv6	0.387	0.301	0.292	4.3	11.9	81.0
YOLOv7-tiny	0.488	0.37	0.358	6.0	13.2	59.5
YOLOv8n	0.437	0.329	0.329	**3.0**	**8.2**	76.2
YOLOv8s	0.481	0.394	0.393	11.1	28.7	68.9
YOLOv8m	0.52	0.416	0.422	25.9	79.1	57.7
Modified-YOLOv8	–	–	0.337	9.66	–	143
TOOD[47]	0.522	0.401	0.413	32.0	199	–
RTMDet[48]	0.534	0.413	0.428	52.3	80	–
Drone-YOLO-n	–	–	0.381	3.1	–	–
GOLD-YOLO[49]	–	–	0.409	5.6	12.1	–
RT-DETR[50]	–	–	0.422	32.68	85.1	–
Edge-YOLO[51]	–	–	0.448	40.5	–	34
PVswin-YOLOv8s[52]	–	–	0.433	41.8	–	–
MSFE-YOLO-m[53]	–	–	0.445	–	–	47.3
YOLOv9s[54]	–	–	0.412	13.7	60.4	70.4
YOLOv11s[55]	–	–	0.364	9.4	21.5	21.3
Ours	**0.573**	**0.447**	**0.459**	3.3	24.9	50.1

Table 4. Compare results with other models on VisDrone2019-val; we use YOLOv8n as the baseline model. The '–' indicates difficult to obtain data. The black bold numbers in the table indicate the best results.

Dataset	Model	P	R	mAP@0.5	mAP@0.5:0.95
SSDD	Faster R-CNN	0.824	0.856	0.926	0.649
	YOLOv3-tiny	0.947	0.840	0.938	0.666
	YOLOv5s	0.956	0.957	0.978	0.707
	YOLOv6	0.955	0.927	0.976	0.695
	YOLOv7-tiny	0.930	0.917	0.957	0.588
	YOLOv8n	0.940	0.931	0.969	0.701
	YOLOv11n	0.952	0.934	0.974	0.698
	Ours	0.955	0.941	0.979	0.699
RSOD	Faster R-CNN	0.918	0.915	0.899	0.574
	YOLOv3-tiny	0.938	0.661	0.730	0.465
	YOLOv5s	0.926	0.920	0.950	0.605
	YOLOv6	0.881	0.933	0.939	0.645
	YOLOv7-tiny	0.912	0.908	0.915	0.586
	YOLOv8n	0.925	0.873	0.919	0.608
	YOLOv11n	0.928	0.875	0.920	0.606
	Ours	0.930	0.921	0.954	0.611

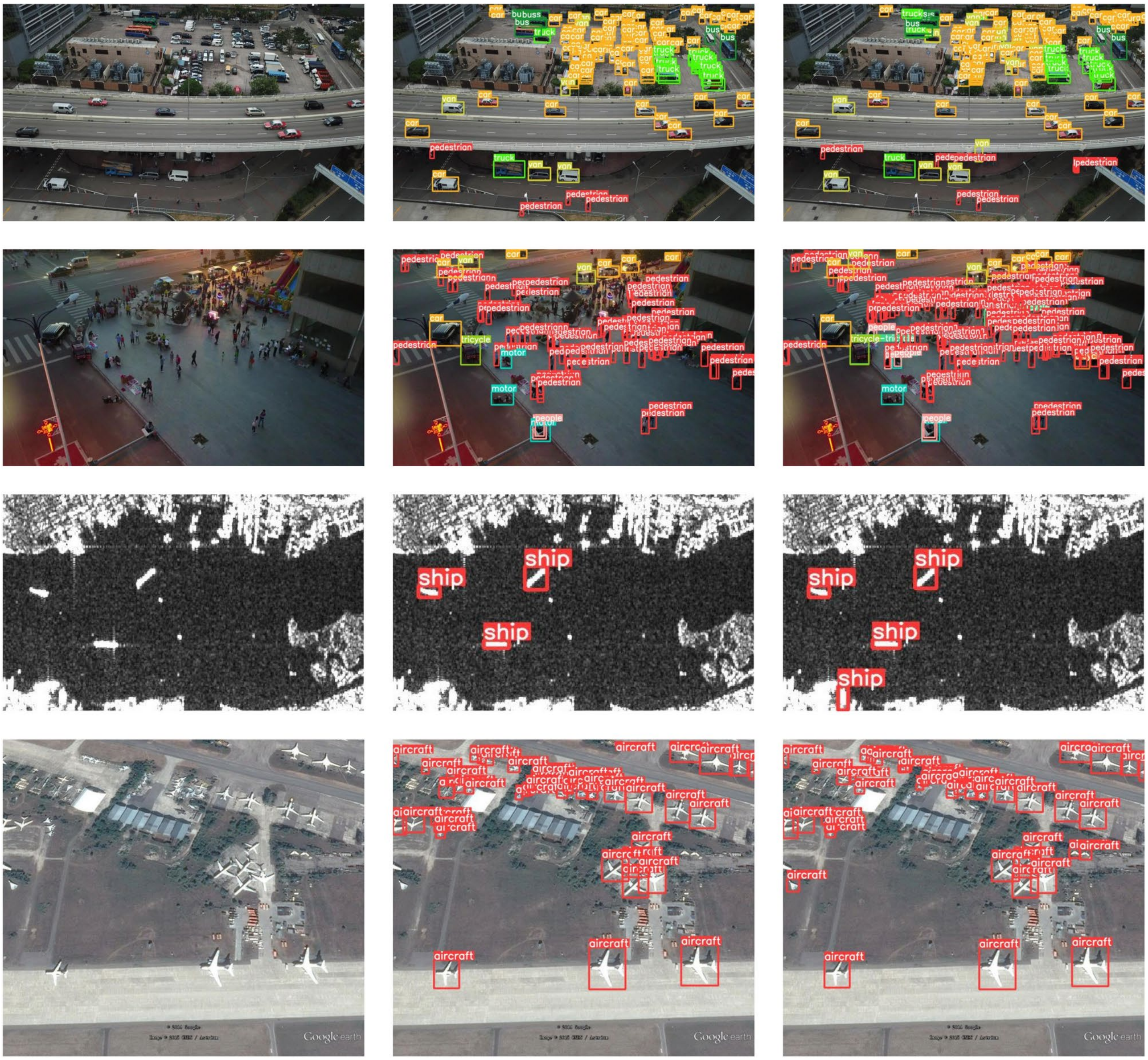
Table 5. Comparison results on other datasets.

that our method effectively reduces false negatives and false positives compared to YOLOv8n, demonstrating superior effectiveness and robustness in various complex scenarios.

These visualisations show that SMA-YOLO can effectively guide the detector to focus on challenging areas, thereby improving the model's attention to critical information. As a result, it shows excellent detection performance in real-world scenarios with varying lighting conditions, complex backgrounds and large changes in object size.

Conclusion

To address the poor performance of remote sensing based small object detection algorithms, we have developed an improved small object detection algorithm based on YOLOv8n. First, inspired by ViT, the algorithm introduces a non-semantic sparse attention mechanism into the backbone network, which allows efficient extraction of task-relevant non-semantic features and significantly improves sensitivity to small objects. In



(a) original images

(b) YOLOv8n

(c) SSA-YOLO

Fig. 7. The visualization results between SMA-YOLO and YOLOv8n are compared on VisDrone2019, SSDD, and RSOD datasets.

addition, we have added a detection branch specifically for small objects and removed the branch originally designed to detect large objects. To further improve the model's ability to detect small objects, we developed a BIMA-FPN feature pyramid network. This module integrates top-down propagation of high-level features and bottom-up propagation of low-level features, effectively compensating for the loss of contextual information, and expanding the multiscale receptive field.Finally, we optimised the detection head by fully exploiting the multiscale features and adaptively dealing with the consistency problem between different feature scales, thus improving the model's adaptability to complex scenarios. The experimental results show that the improved model achieves 45.9% mAP@0.5, which is 13% higher than the basic model, and the AP_{small} accuracy is improved by 7.3% while maintaining a low parameter amount, highlighting its significant advantages in detecting small objects in remote sensing images.

Looking ahead, our research will continue to focus on improving the model's light weight, detection accuracy and processing speed, in order to more effectively deploy and improve the performance of small object detection. However, current methods still face challenges such as domain adaptation, scale change and deployment efficiency. In the future, we will solve these problems through transfer learning and optimisation of multi-scale feature fusion, and further improve the robustness and generalisation ability of the model under different weather conditions to promote its wide use in practical applications. We also plan to combine new models from the YOLO family or other architectures with the methods in this study to further improve detection performance and practicality.

Data availability

All data generated or analysed during this study are included in this published article.

Received: 28 January 2025; Accepted: 26 February 2025
Published online: 18 March 2025

References

1. Hasan, A. F. *et al.* Fractional order extended state observer enhances the performance of controlled tri-copter uav based on active disturbance rejection control. in *Mobile Robot: Motion Control and Path Planning*, 439–487 (Springer, 2023).
2. Beg, A., Qureshi, A. R., Sheltami, T. & Yasar, A. UAV-enabled intelligent traffic policing and emergency response handling system for the smart city. *Personal Ubiquit. Comput.* **25**, 33–50 (2021).
3. Liu, K. & Zheng, J. UAV trajectory optimization for time-constrained data collection in UAV-enabled environmental monitoring systems. *IEEE Internet Things J.* **9**, 24300–24314 (2022).
4. Yan, Y., Chen, X., Shi, M. & Li, R. A decision support system architecture for intelligent driven unmanned aerial vehicles maritime search and rescue. in *2024 10th international symposium on system security, safety, and reliability (ISSSR)*, 424–428 (IEEE, 2024).
5. Wu, J., Wang, H. & Zhang, M. Urban crowd surveillance in an emergency using unmanned air vehicles. *J. Guid. Control. Dyn.* **43**, 838–846 (2020).
6. Zhao, D. et al. A small object detection method for drone-captured images based on improved yolov7. *Remote Sens.* **16**, 1002 (2024).
7. Ren, S., He, K., Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
8. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. in *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (2017).
9. Cai, Z. & Vasconcelos, N. Cascade r-cnn: delving into high quality object detection. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 6154–6162 (2018).
10. Dai, J., Li, Y., He, K. & Sun, J. R-fcn: object detection via region-based fully convolutional networks. *Advances in neural information processing systems* **29** (2016).
11. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788 (2016).
12. Redmon, J. & Farhadi, A. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018).
13. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7464–7475 (2023).
14. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020).
15. Alhawsawi, A. N., Khan, S. D. & Rehman, F. U. Enhanced yolov8-based model with context enrichment module for crowd counting in complex drone imagery. *Remote Sens.* **16**, 4175 (2024).
16. Liu, W. *et al.* Ssd: Single shot multibox detector. in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37 (Springer, 2016).
17. Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. Focal loss for dense object detection. in *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988 (2017).
18. Liu, S., Zha, J., Sun, J., Li, Z. & Wang, G. Edgeyolo: An edge-real-time object detector. in *2023 42nd Chinese Control Conference (CCC)*, 7507–7512 (IEEE, 2023).
19. Zhu, X., Lyu, S., Wang, X. & Zhao, Q. Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2778–2788 (2021).
20. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. Cbam: Convolutional block attention module. in *Proceedings of the European Conference on Computer Vision (ECCV)*, 3–19 (2018).
21. Zhao, L. & Zhu, M. Ms-yolov7: Yolov7 based on multi-scale for object detection on uav aerial photography. *Drones* **7**, 188 (2023).
22. Zhang, C., Wang, L., Cheng, S. & Li, Y. Swinsunet: Pure transformer network for remote sensing image change detection. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13 (2022).
23. Zhang, Z. Drone-yolo: An efficient neural network method for target detection in drone images. *Drones* **7**, 526 (2023).
24. Wang, G. et al. UAV-yolov8: A small-object-detection model based on improved yolov8 for UAV aerial photography scenarios. *Sensors* **23**, 7190 (2023).
25. Qi, S., Song, X., Shang, T., Hu, X. & Han, K. Msfe-yolo: An improved yolov8 network for object detection on drone view. *IEEE Geoscience and Remote Sensing Letters* (2024).
26. Li, K., Wang, D., Xu, H., Zhong, H. & Wang, C. Language-guided progressive attention for visual grounding in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* https://doi.org/10.1109/TGRS.2024.3423663 (2024).
27. Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
28. Bahdanau, D. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014).
29. Li, K. *et al.* Unleashing channel potential: Space-frequency selection convolution for sar object detection. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17323–17332 (2024).
30. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141 (2018).
31. Wang, Q. *et al.* Eca-net: Efficient channel attention for deep convolutional neural networks. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1534–11542 (2020).
32. Hou, Q., Zhou, D. & Feng, J. Coordinate attention for efficient mobile network design. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13713–13722 (2021).
33. Wang, C.-Y. *et al.* Cspnet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 390–391 (2020).
34. Dosovitskiy, A. *et al.* An image is worth 16x16 words. arXiv preprint arXiv:2010.119297 (2020).
35. Song, H. et al. Qaga-net: Enhanced vision transformer-based object detection for remote sensing images. *Int. J. Intell. Comput. Cybernet.* https://doi.org/10.1108/IJICC-08-2024-0383 (2024).
36. Song, H., Yuan, Y., Ouyang, Z., Yang, Y. & Xiang, H. Efficient knowledge distillation for hybrid models: A vision transformer-convolutional neural network to convolutional neural network approach for classifying remote sensing images. *IET Cyber-Syst. Robot.* **6**, e12120 (2024).
37. Khan, S. D., Alarabi, L. & Basalamah, S. A unified deep learning framework of multi-scale detectors for geo-spatial object detection in high-resolution satellite images. *Arab. J. Sci. Eng.* **47**, 9489–9504 (2022).

38. Lin, T.-Y. *et al.* Feature pyramid networks for object detection. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2117–2125 (2017).
39. Ghiasi, G., Lin, T.-Y. & Le, Q. V. Nas-fpn: Learning scalable feature pyramid architecture for object detection. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7036–7045 (2019).
40. Liu, S., Huang, D. & Wang, Y. Learning spatial fusion for single-shot object detection. arXiv preprint arXiv:1911.09516 (2019).
41. Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587 (2014).
42. Chen, P.-Y., Chang, M.-C., Hsieh, J.-W. & Chen, Y.-S. Parallel residual bi-fusion feature pyramid network for accurate single-shot object detection. *IEEE Trans. Image Process.* **30**, 9099–9111 (2021).
43. Zheng, Z. et al. Distance-iou loss: Faster and better learning for bounding box regression. *Proceed. AAAI Confer. Art. Intell.* **34**, 12993–13000 (2020).
44. Su, L. *et al.* Can we get rid of handcrafted feature extractors? sparsevit: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer. arXiv preprint arXiv:2412.14598 (2024).
45. Du, D. *et al.* Visdrone-det2019: The vision meets drone object detection in image challenge results. in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops* (2019).
46. Lin, T.-Y. *et al.* Microsoft coco: Common objects in context. in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, 740–755 (Springer, 2014).
47. Feng, C., Zhong, Y., Gao, Y., Scott, M. R. & Huang, W. Tood: Task-aligned one-stage object detection. in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3490–3499 (IEEE Computer Society, 2021).
48. Lyu, C. *et al.* Rtmdet: An empirical study of designing real-time object detectors. arXiv preprint arXiv:2212.07784 (2022).
49. Wang, C. *et al.* Gold-yolo: Efficient object detector via gather-and-distribute mechanism. *Advances in Neural Information Processing Systems*, **36** (2024).
50. Zhao, Y. *et al.* Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16965–16974 (2024).
51. Liang, S. et al. Edge yolo: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **23**, 25345–25360 (2022).
52. Tahir, N. U. A., Long, Z., Zhang, Z., Asim, M. & ELAffendi, M. Pvswin-yolov8s: UAV-based pedestrian and vehicle detection for traffic management in smart cities using improved yolov8. *Drones* **8**, 84 (2024).
53. Qi, S., Song, X., Shang, T., Hu, X. & Han, K. Msfe-yolo: An improved yolov8 network for object detection on drone view. *IEEE Geosci. Remote Sens. Lett.* https://doi.org/10.1109/LGRS.2024.3432536 (2024).
54. Wang, C.-Y., Yeh, I.-H. & Liao, H.-Y. M. Yolov9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2402.13616 (2024).
55. Khanam, R. & Hussain, M. Yolov11: An overview of the key architectural enhancements. arXiv preprint arXiv:2410.17725 (2024).
56. Zhang, T. et al. Sar ship detection dataset (ssdd): Official release and comprehensive data analysis. *Remote Sens.* **13**, 3690 (2021).
57. Long, Y., Gong, Y., Xiao, Z. & Liu, Q. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **55**, 2486–2498 (2017).
58. Wang, H. et al. Manifold-based incomplete multi-view clustering via bi-consistency guidance. *IEEE Trans. Multim.* https://doi.org/10.1109/TMM.2024.3405650 (2024).

Author contributions

Conceptualization, S.Z.; methodology, S.Z.; investigation, S.Z. and H.Z.; validation, H.Z. and L.Q.; formal analysis, S.Z. and H.Z.; data curation, S.Z.; writing-original draft preparation, S.Z.; writing-review and editing, S.Z., and H.Z.; visualization, S.Z.; supervision, H.Z. and L.Q.; project administration, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding

This work was funded by the National Key R&D Program of China, No:2022YFB3904800.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

The corresponding author is responsible for submitting a competing interests statement on behalf of all authors of the paper. This statement must be included in the submitted article file.

Additional information

Correspondence and requests for materials should be addressed to H.Z.