electronics

MDPI

Article

AID-YOLO: An Efficient and Lightweight Network Method for Small Target Detector in Aerial Images

Yuwen Li [1], Jiashuo Zheng [1,*], Shaokun Li [2], Chunxi Wang [3,*], Zimu Zhang [3] and Xiujian Zhang [3]

[1] School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; liyuwen@seu.edu.cn

[2] School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China; 3120221325@bit.edu.cn

[3] Beijing Aerospace Institute for Metrology and Measurement Technology, Beijing 100076, China; zhangzimu@nudt.edu.cn (Z.Z.); caltzzm@126.com (X.Z.)

* Correspondence: 220223358.seu@vip.163.com (J.Z.); chunxiww@163.com (C.W.)

Abstract: The progress of object detection technology is crucial for obtaining extensive scene information from aerial perspectives based on computer vision. However, aerial image detection presents many challenges, such as large image background sizes, small object sizes, and dense distributions. This research addresses the specific challenges relating to small object detection in aerial images and proposes an improved YOLOv8s-based detector named Aerial Images Detector-YOLO(AID-YOLO). Specifically, this study adopts the General Efficient Layer Aggregation Network (GELAN) from YOLOv9 as a reference and designs a four-branch skip-layer connection and split operation module Re-parameterization-Net with Cross-Stage Partial CSP and Efficient Layer Aggregation Networks (RepNCSPELAN4) to achieve a lightweight network while capturing richer feature information. To fuse multi-scale features and focus more on the target detection regions, a new multi-channel feature extraction module named Convolutional Block Attention Module with Two Convolutions Efficient Layer Aggregation Net-works (C2FCBAM) is designed in the neck part of the network. In addition, to reduce the sensitivity to position bias of small objects, a new function, Normalized Weighted Distance Complete Intersection over Union (NWD-CIoU_Loss) weight adaptive loss function, was designed in this study. We evaluate the proposed AID-YOLO method through ablation experiments and comparisons with other advanced models on the VEDAI (512, 1024) and DOTAv1.0 datasets. The results show that compared to the Yolov8s baseline model, AID-YOLO improves the mAP@0.5 metric by 7.36% on the VEDAI dataset. Simultaneously, the parameters are reduced by 31.7%, achieving a good balance between accuracy and parameter quantity. The Average Precision (AP) for small objects has improved by 8.9% compared to the baseline model (YOLOv8s), making it one of the top performers among all compared models. Furthermore, the FPS metric is also well-suited for real-time detection in aerial image scenarios. The AID-YOLO method also demonstrates excellent performance on infrared images in the VEDAI1024 (IR) dataset, with a 2.9% improvement in the mAP@0.5 metric. We further validate the superior detection and generalization performance of AID-YOLO in multi-modal and multi-task scenarios through comparisons with other methods on different resolution images, SODA-A and the DOTAv1.0 datasets. In summary, the results of this study confirm that the AID-YOLO method significantly improves model detection performance while maintaining a reduced number of parameters, making it applicable to practical engineering tasks in aerial image object detection.

Keywords: small object detection; aerial images; four-branch skip-layer connection and split operation module; convolutional block attention module with two convolutions efficient layer aggregation networks; weight-assignment-regression cost function

check for updates

Citation: Li, Y.; Zheng, J.; Li, S.; Wang, C.; Zhang, Z.; Zhang, X. AID-YOLO: An Efficient and Lightweight Network Method for Small Target Detector in Aerial Images. *Electronics* **2024**, *13*, 3564. https://doi.org/10.3390/electronics13173564

Academic Editor: Pawel Strumillo

Received: 29 July 2024
Revised: 5 September 2024
Accepted: 6 September 2024
Published: 8 September 2024

1. Introduction

In recent years, with the rapid development of UAV remote sensing technology, drones equipped with various imaging systems, such as high-resolution cameras and infrared sensors, can capture extensive scene information. This aerial image object detection technology, aimed at air-to-ground applications, has become a crucial component in modern military systems for intelligence gathering [1]. In the civilian sector, as computer vision technology matures and aerial images offer advantages such as wide viewing angles, large monitoring areas, ease of operation, and mobile deployment [2], they have been widely used and recognized in fields such as perimeter surveying, natural resource management, disaster response, emergency rescue, and ground traffic vehicle detection (Figure 1). Therefore, utilizing UAV remote sensing platforms for aerial image object detection to capture more real-time ground information can enhance the applicability and convenience of ground object detection. The goal of object detection is to identify and locate instances of interest. Although general object detection has been widely applied in many fields, aerial image detection faces more challenges than ordinary image detection.

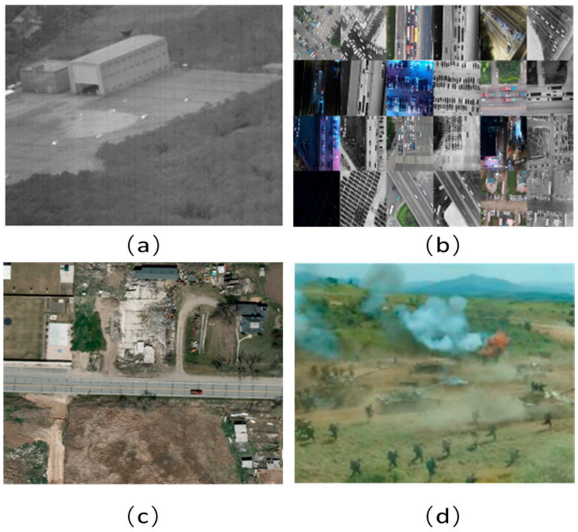


Figure 1. Application scenarios of aerial images: (**a**) infrared unmanned aerial vehicle detection in high-altitude military scenarios; (**b**) high-altitude drone-based traffic vehicle detection; (**c**) high-altitude ground target detection in aerial remote sensing images; (**d**) high-altitude drone battlefield environment monitoring.

On the one hand, small object detection in air-to-ground scenes is challenging mainly due to the significant external condition variations, complex backgrounds, small target imaging proportions, significant differences in the sizes of the targets, extensive redundant background information, changes in lighting conditions, and similar target colors. These factors challenge the detector's ability to distinguish and recognize small ground targets. Currently, most object detection techniques are applied in singular contexts and exhibit poor generalization. Aerial imagery covers a wide range of targets and includes numerous background objects, making it suitable for diverse application scenarios due to its vast shooting scenes and variable angles [3]. In detecting small objects in aerial images, it is crucial to enhance network architectures to improve detection accuracy and utilize multi-scale feature fusion mechanisms to comprehensively improve the performance of object detection tasks in aerial imagery [4].

On the other hand, current general solutions for small ground object detection mainly involve designing a complex deep neural network to separate strong feature representations of objects from the background. It often leads to a heavy computational burden, large model structures, and complex parameters, requiring more computational resources and large computing devices [5], which fails to meet the lightweight and real-time requirements for edge deployment on small embedded mobile devices in practical engineering.

Therefore, neural network methods for object detection in aerial images must be adapted to the specific characteristics of these challenges. The focus of this paper is on small object detection in aerial images. We have developed a recognition model with high accuracy, fast real-time detection, reduced parameter count, and strong generalization capabilities. This innovative approach significantly enhances the ability to detect and recognize ground objects in aerial images, addressing practical needs. To this end, we designed an innovative detector named AID-YOLO specifically for small object detection in aerial imagery. A key innovation of this work is that the AID-YOLO model maintains stable accuracy improvements with a reduced parameter count, achieving an optimal balance between model complexity and performance.

We trained and evaluated AID-YOLO using the VEDAI (512, 1024) and DOTAv1.0 datasets, which comprise aerial viewpoint images annotated with diverse labels for small objects. The results demonstrate that compared to the current state-of-the-art methods, AID-YOLO improves the detection capability of small ground objects while reducing the number of parameters.

Our contributions are summarized as follows:

(1) To better extract small object feature information, we utilized the concept of module splitting and reorganization along with efficient the layer aggregation networks (ELAN hierarchical processing approach to reconstruct a four-branch skip-layer connection and split operation feature extraction module, RepNCSPELAN4, to replace the CSPDarknet53 to 2-Stage FPN (C2F) module in YOLOv8. It allows the model to comprehensively improve small object detection performance in terms of being lightweight, having a better inference speed, and enhanced accuracy.

(2) To integrate multi-scale feature information and emphasize the detection of target regions, we introduce the Multi-Spatial Channel Enhanced Convolutional Block Attention Module (MSCE-CBAM) in the neck section. Consequently, a novel feature extraction module, C2FCBAM, is formed at the neck, enabling the network to focus more on detecting small target regions while learning richer features.

(3) To tackle the discrepancies in the sensitivity of the Intersection Over Union (IoU) loss function towards objects of varying scales, we have introduced a novel weight-assignment-regression cost function called NWD-CIoU_Loss. We enhanced the detection performance of small objects by introducing the Normalized Weighted Distance (NWD) loss function and assigning weight distribution coefficients to adjust the relative importance between the two losses.

2. Related Work

In aerial devices, object detection aims to identify objects from various complex ground backgrounds, determine their locations and categories, and thus complete subsequent tasks such as information collection and ground surveying to meet the requirements. The rise of deep learning technology has led to significant progress in deep neural network-based object detection methods, which perform exceptionally well in complex scenes. The mainstream object detection algorithms mainly include the two-stage detectors represented by the Faster R-CNN [6] series, the one-stage detectors represented by the YOLO [7] series and SSD [8], and the Transformer-based DETR series [9] and the latest Mamba [10] methods. Transformer architectures in the natural language processing domain are too complex to apply directly to engineering applications due to their large parameter volumes and training difficulties. One-stage object detection models, which input images directly into the network and output classification and regression results from a single network model, significantly enhance detection efficiency and are more suitable for real-time detection requirements. Therefore, the YOLO series remains the most widely used algorithm. We selected YOLOv8 [11] as the baseline for improvement in this paper because it has demonstrated effectiveness and power in numerous computer vision tasks.

Given the extensive view, complex background, small target imaging proportions, mutual occlusion between different target sizes, and significant differences in aerial images for ground–air scenes, many researchers have applied deep learning models to UAV remote sensing images for target feature analysis to improve the accuracy of detection algorithms. To effectively address the complex background problem of aerial images, in 2019, Xue Yang et al. proposed the SCRDet method [12]. This approach effectively distinguishes the salient features of targets from the surrounding distractions, resulting in improved accuracy and robustness in aerial image object detection tasks. Chen et al. [13] have achieved remarkable detection performance in real-world unmanned aerial vehicle (UAV) scenarios by optimizing the residual connection modules within convolutional networks and augmenting the number of convolutional kernels, thereby enhancing the network's capability in feature extraction from high-resolution aerial imagery.

Jiang et al. [14] proposed a UAV target detection framework for infrared images and videos, extracting features from ground targets and using an improved YOLOv5s for ground object recognition. This algorithm achieved high recognition accuracy and fast recognition speed. Li et al. [15] proposed an enhanced small parameter target detection network based on the YOLOv8s model. The authors replaced the PAFPN structure of YOLOv8 with Bi-FPN and improved the backbone with the Ghostblock [16] module, achieving a neural network method with fewer parameters but better detection performance. Wang et al. [17] enhanced the YOLOv8 model by incorporating a small target structure (STC) into the neck of the network. This modification addresses the challenges of detecting small targets in aerial images by capturing contextual information and minimizing detection information loss. However, this approach increases the number of parameters. In a related study, Pan et al. [18] improved upon the YOLOv8 model by modifying the Conv and C2f layers with the integration of RFCBAM and the adoption of an enhanced inner-MPDIoU as the model's bounding box regression loss. These enhancements boost the model's ability to learn from complex small samples. Despite these improvements, further parameter optimization remains necessary. Applying transformer-based methods for aerial image object detection has garnered significant attention as a research hotspot in computer vision. DETR (Detection Transformer) [19] has pioneered the introduction of transformers into the domain of object detection, successfully integrating Convolutional Neural Networks (CNNs) with transformers. Subsequently, addressing the shortcomings of DETR in terms of model convergence speed and resource consumption during training, improved models, such as efficient DETR [20], PnP DETR [21], and sparse DETR [22], have emerged. While these advancements have enhanced recognition accuracy for aerial remote sensing tasks, they still need to improve algorithmic real-time performance and substantial parameter sizes, posing challenges for deployment on edge devices. Therefore, there is an ongoing need to develop more efficient and lightweight transformer-based solutions that balance high accuracy and reduced computational complexity, enabling practical applications in resource-constrained environments.

In recent years, rotating object detection has emerged as a popular research direction. The center point, width, height, and rotation angle typically define rotating bounding boxes. Rotating boxes can more closely surround the targets for complex-shaped objects while aligning with their shapes and orientations. Sharma et al. [23] introduced the Yolors model, which enhances the detection of rotated and closely spaced small objects in aerial imagery, facilitating real-time target detection in more significant aerial scenes. Yao et al. [24] addressed the issues of boundary discontinuity in two-stage oriented bounding box (OBB) detection by proposing a simple and effective bounding box representation inspired by polar coordinates, integrating it into both detection stages. This method achieves a good balance between accuracy and speed among mainstream two-stage oriented detectors. However, the meticulous labeling of rotating boxes for vast imagery is time-consuming and labor-intensive. Luo et al. [25] proposed a method for directed object detection based on single-point labels to tackle the high costs of rotating box annotations. By designing a multi-view cooperative optimization strategy, they effectively predict the rotating bounding boxes from point labels, significantly reducing annotation costs while achieving competitive performance on two remote sensing object detection datasets.

Although these methods have significantly improved when applied to aerial image object detection, effectively detecting small targets from wide-area backgrounds still faces many challenges. The main challenges are as follows: (1) the diminutive size of ground targets results in their limited presence within aerial images, while the background occupies a substantial portion, thereby providing restricted detection information; (2) the down-sampling process within the detection network may lead to the disappearance of crucial features necessary for small object detection; (3) small objects within aerial images may encounter challenges such as interference from background colors, occlusion, and varying angles, making their differentiation from the background or similar objects difficult; (4) avoid designing large and complex deep networks for small object extraction. The AID-YOLO method, as outlined in this paper, tackles the previously mentioned issues by employing hierarchical processing for module splitting and reorganization, as well as multi-scale fusion. This approach enables the detection of small recognition targets, making it well-suited for multi-task cross-modal scenarios in aerial image object detection tasks.

3. Methods

3.1. Overview of AID-YOLO

AID-YOLO is an improved version of the YOLOv8 model, comprising three main components, as illustrated in Figure 2. In the backbone of the network, AID-YOLO incorporates the following key components: A convolutional layer, batch normalization, a sigmoid linear unit (CBS), RepNCSPELAN4, and Spatial Pyramid Pooling Fast (SPPF). The significant improvements in the backbone manifest in the reconstructed four-branch skip layer connections and the split operation feature extraction module, RepNCSPELAN4, which replaces the C2F module in the original YOLOv8. This modification reduces the model's weight while facilitating more effective propagation and aggregation of feature information across different levels, thereby enhancing feature extraction. In the neck of the network, while the overall architecture still utilizes a Feature Pyramid Network (FPN) and a Path Aggregation Network (PAN) for top–down and bottom–up feature pyramid structures, the C2F module incorporates a multi-space channel enhancement convolutional attention mechanism to form the C2FCBAM module. It facilitates feature fusion across different scales, improving the model's focus and recognition capabilities for small objects. In the head of the network, improvements are constructed through the regression branch, which employs Distribution Focal Loss (DFL) and introduces a weight allocation method using NWD-CIoU loss. It addresses the issue of the IoU loss function's varying sensitivity to objects of different scales. By assigning weighting coefficients, the relative importance of the two loss functions is adjusted, ultimately enhancing the detection performance for small objects.

3.2. Backbone Network

The proposed backbone network model primarily includes convolutional layers and RepNCSPELAN4 components. Each stage block contains a Conv block with a stride of 2 and a RepNCSPELAN4 module (Figure 3a) for down-sampling and feature extraction. The most critical component for feature map extraction in the backbone is the RepNCSPELAN4 module. This design integrates the RepConv structure and draws inspiration from Cross-Stage Partial Network (CSPNet) and ELAN [26], generally adopting a four-branch skip-layer connection and split operation for feature extraction. Replacing the original C2F module in YOLOv8 with this design has yielded significant improvements.

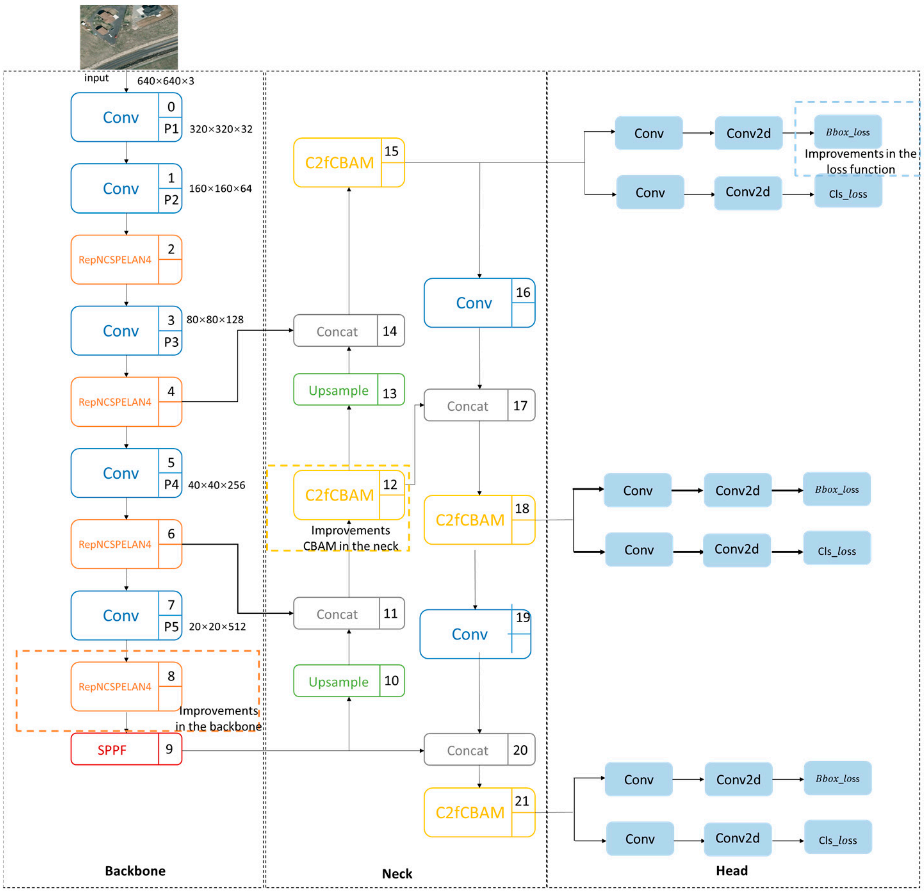
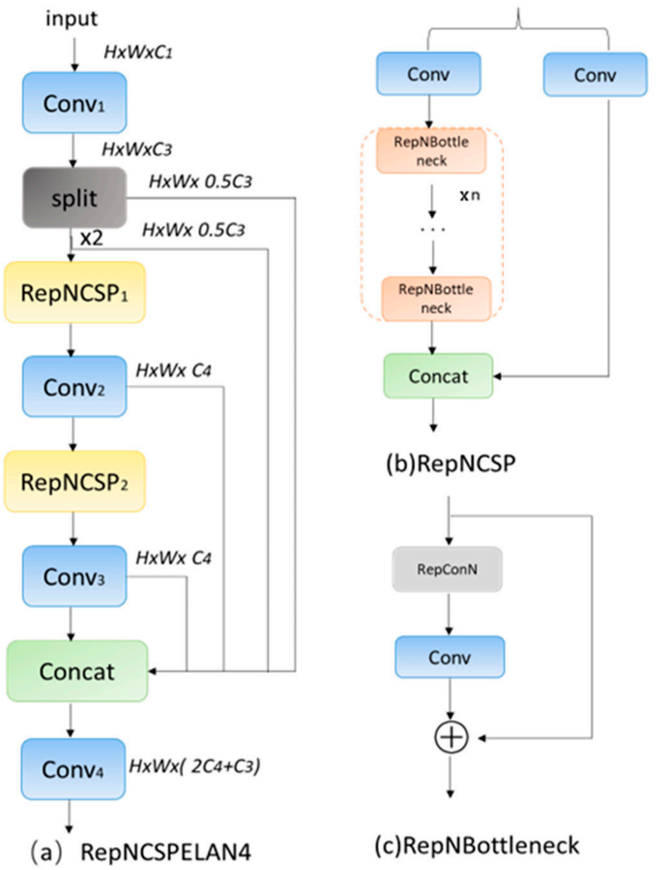


Figure 2. AID-YOLO model overall design framework. In the designed network architecture, the numbers 0–21 represent the layers specified in the configuration file. The notations P3–P5 denote the three output detection layers. Figures 320 × 320 × 32 indicate that the size of the feature map input into the network after down-sampling is 320 × 320, with 32 channels. Other numbers follow this pattern.

In YOLOv9 [27], the authors introduced an improved Generalized Efficient Layer Aggregation Network (GELAN), which integrates the CSPNet and ELAN models to optimize gradient pathways for enhanced feature information propagation and aggregation. The design principle of CSPNet involves partitioning the feature map into two segments: one undergoes convolution. In contrast, the other merges with the upper-layer convolution results through a cross-stage method. This separation of gradient flow reduces computational complexity and enriches branch fusion information. ELAN, by contrast, enhances gradient flow and aggregates features from multiple levels, ensuring that each layer incorporates a “Resnet” pathway. This design improves the model’s receptive field and feature representation capabilities, effectively mitigating the challenges associated with increased training difficulty. The GELAN significantly reduces computational load and parameter count while maintaining detection performance.

We draw inspiration from the GELAN module proposed in YOLOv9, employing the concepts of segmentation and recombination while introducing a hierarchical processing approach. This design enhances feature extraction capabilities alongside network channel expansion. Consequently, we developed the RepNCSPELAN4 feature extraction module, which leverages comprehensive gradient flow information to improve the extraction of features related to small targets for detection tasks. Specifically, the module incorporates a four-branch skip layer connection and split feature extraction operations, allowing RepNCSPELAN4 to be defined as follows:

$$RepNCSPELAN4(Conv_1(C_1)) = Conv_4(\text{concat}(\text{split}_1(0.5C_3), \text{split}_2(0.5C_3), Conv_2(C_4), Conv_3(C_4))) \quad (1)$$



(a) RepNCSPELAN4

(b)RepNCSP

(c)RepNBottleneck

Figure 3. The module of RepNCSPELAN4: (**a**) RepNCSPLEN4; (**b**) RepNCSP; (**c**) RepNBottleneck.

In this module, C_1 serves as the input, which passes through the layer of Conv$_1$ and produces an output of C_3. Subsequently, a split convolution operation is performed, where split1(0.5C_3) and split2(0.5C_3) represent the first two segments of the Conv output, divided along the channel dimension, with no further operations applied before outputting these segments. The outputs generated by Conv$_2$ (C_4) and Conv$_3$ (C_4) result from independent convolution and pooling operations applied to their respective channels. Finally, the four segments are concatenated along the dimension to produce the final output features. The specific process is delineated as follows:

First, in the initial two branches, namely split$_1$ and split$_2$, the absence of any operational transformations leads to a direct output equivalent to 2 × 0.5 C_3. In the subsequent feature extraction modules, the third branch undergoes processing through the Re-parameterization-Net with Cross-Stage Partial (RepNCSP$_1$) submodule (Figure 3b). RepNCSP$_1$ divides the input features into two segments: one segment undergoes feature extraction via the Re-parameterization Bottleneck without Identity Connection (RepNBottleneck), while the other employs conventional convolution operations. These two segments are concatenated to augment the network's feature extraction capabilities and performance, enhancing its overall efficacy in capturing relevant information. The architecture of the RepNBottleneck submodule (Figure 3c) references the ResNet structure, where one branch maintains the output channel count at $C_4/2$. In contrast, the other branch is processed through the Re-parameterization Convolution without Identity Connection (RepConvN) submodule (Figure 4). The core idea of RepConvN is to utilize two distinct convolution kernel sizes, 3 × 3 and 1 × 1, for feature extraction. During the inference phase, structural re-parameterization merges the 1 × 1 and 3 × 3 convolution kernels into a single 3 × 3 kernel. Specifically, the 1 × 1 kernel is padded to match the size of the 3 × 3 kernel, allowing the padded kernel to be added to the original 3 × 3 kernel based on the principle of additivity of kernels of the same size, thereby forming a 3 × 3 convolution kernel for inference. Applying RepConvN within the RepNBottleneck submodule enhances the model's efficiency and performance. After the nested RepNBottleneck feature extraction is completed, the process returns to the RepNCSP_1 submodule, where N operations of the RepNBottleneck submodule are sequentially executed. The output from this module is then concatenated with the original convolution channel, resulting in an output feature size of C_4 after passing through Conv_2. Subsequently, the output features from the third branch serve as input for the fourth branch, which again enters the RepNCSP_2 submodule, repeating the operations above. Ultimately, the output features from the RepNCSP_2 submodule, after passing through Conv_3, also yield a size of C_4.

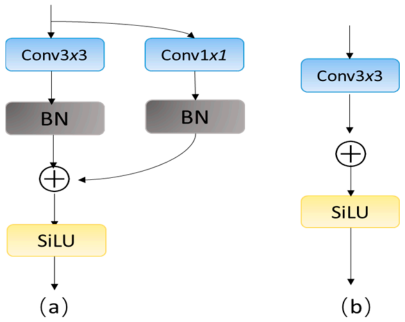


Figure 4. RepConv module structure: (**a**) RepConv layer in training; (**b**) RepConv layer in inference.

The final output consists of four distinct channels: split_1 and split_2, each representing the initial two outputs with channel dimensions of 0.5 C_3, respectively. These are subsequently processed through the RepNCSP_1 submodule and Conv_2 block, leading to the third output feature with C_4 channels. Further, the fourth path involves the RepNCSP_2 submodule and Conv_3 block, resulting in an output feature of C_4 channels. Ultimately, these four channels undergo a Concat operation, yielding a final output feature of $C_1 = 0.5\ C_3 + 0.5C_3 + C_4 + C_4 = C_3 + 2 \times C_4$. This module's innovative design lies in its capability to amplify channel dimensions while effectively learning multi-scale small object features and expanding the receptive field through intra-module feature vector splitting and multi-level nested convolutions. This approach not only enhances the network's efficiency but also addresses the issue of excessive parameter counts in the original C2F module of Yolov8, which arises from fusing features from different hierarchical levels. Consequently, the proposed module achieves a comprehensive improvement characterized by reduced parameter counts, heightened detection accuracy, and better training generalization. It offers a more streamlined and efficient alternative to conventional methods, thereby contributing to advancements in object detection performance.

3.3. Neck Network

In YOLOv8, the neck network maintains the PAFPN structure, which can fuse feature maps of different scales and provide richer feature representations. At this point, the C2f module simultaneously fuses low-resolution and high-resolution feature maps to enhance detection accuracy. In recent years, attention mechanisms have been widely introduced into object detection architectures to optimize models and achieve significant results. Through learning and model training, deep neural networks can learn which regions need specific attention in each new image, forming the necessary attention. Among these, self-attention, spatial attention, temporal attention mechanisms, and branch attention are the most typical attention mechanisms. Therefore, adding attention mechanisms in the neck not only allows the network to focus more on the target regions and model them more finely but also helps the network focus on edge, texture, and other detail information of small target objects in the image data, thereby improving the overall recall and precision of object detection.

The Convolutional Block Attention Module (CBAM) is an attention mechanism used in computer vision tasks, particularly suitable for Convolutional Neural Networks (CNNs). The principle underlying the CBAM involves the fusion of channel and spatial attention, considering the significance of features in both the channel and spatial dimensions. This approach refines the input feature map in two stages. By jointly using these two attention mechanisms, it can better capture the key features of small targets in images. Fundamentally, the channel attention mechanism first focuses on "which channels are important", using parallel operations of average pooling and max pooling to integrate the spatial information in the input feature map, obtaining dual feature maps. These dual feature maps are then fed into a shared multilayer perceptron, adding the output features of the two multilayer perceptrons one by one and generating the channel attention map through a sigmoid activation function. The spatial attention mechanism first focuses on "where are important", performing parallel operations of global max pooling and global average pooling at the channel level on the input feature map to obtain a pair of feature maps. Next, these feature maps are concatenated along the channel axis and convolved to reduce parameter count. Subsequently, a sigmoid operation generates spatial attention features. This mechanism adapts to improve the model's focus on critical features, enhancing recognition ability. More importantly, the CBAM is a lightweight, universal module that enhances training efficiency without adding computational burden to the network, making it simple and efficient.

Therefore, this paper also designs a multi-spatial channel-enhanced C2FCBAM module in AID-YOLO, as shown in Figure 5, aiming to improve the detection of small objects. As shown in Figure 4, the neck network C2f module plays a crucial role in the CSP Bottleneck structure. Through feature transformation, branch processing, and feature fusion operations, it can extract and transform the features of the input data, generating more representative outputs. It aids in enhancing the performance and representation capability of the network, thereby facilitating its improved adaptability to intricate data tasks. Hence, in the CBAM attention mechanism, based on assigning convolutional attention weights in both spatial and channel dimensions, we cascade and enhance the two CBAM modules and combine them with the Bottleneck module at the neck network's C2F part. The Bottleneck module is still based on two convolutional modules, first passing through the initial convolutional layer and then replacing the second convolutional layer with the second nested and MSCE-CBAM. This internal and external nested double residual information linkage further enhances the model's ability to focus on crucial attributes of the detection object, improving detection performance.

3.4. Head Network

The detection samples in this dataset primarily focus on small targets. Tiny objects generally have dimensions smaller than 16x16 pixels, providing extremely restricted visual information. The elevated complexity imposed on the network model hampers its ability to acquire discriminative features for detecting diminutive targets, resulting in an elevated rate of missed detections. Currently, mainstream bounding boxes adopt BCE (Binary Cross-Entropy) as the classification loss and IoU (Intersection over Union) as the regression loss. Despite many modifications, IoU's sensitivity to objects of different scales varies greatly, as shown in Figure 6. For example, with small objects in the recognition dataset, minor positional deviations can cause significant IoU drops, resulting in inaccurate positive and negative sample label assignments. However, the Intersection over Union (IoU) demonstrates minimal variations for larger objects, suggesting discretization of IoU measurements when accounting for objects of diverse scales and positional deviations. Therefore, using the IoU series as a loss function for small object detection models can lead to insufficient feature information feedback for small targets, causing the model to focus only on larger targets while neglecting small target feature learning, making it difficult for the model to converge. Although the CIoU loss function combines the characteristics of Generalized Intersection over Union (GIoU)and IoU loss functions, considering the area and center distance of the bounding box. Since CIoU is designed based on the object's area, the influence of larger objects becomes more significant, potentially leading to excessive correction for smaller objects. Thus, using different loss weights for recognition targets of varying sizes to improve this issue is crucial.

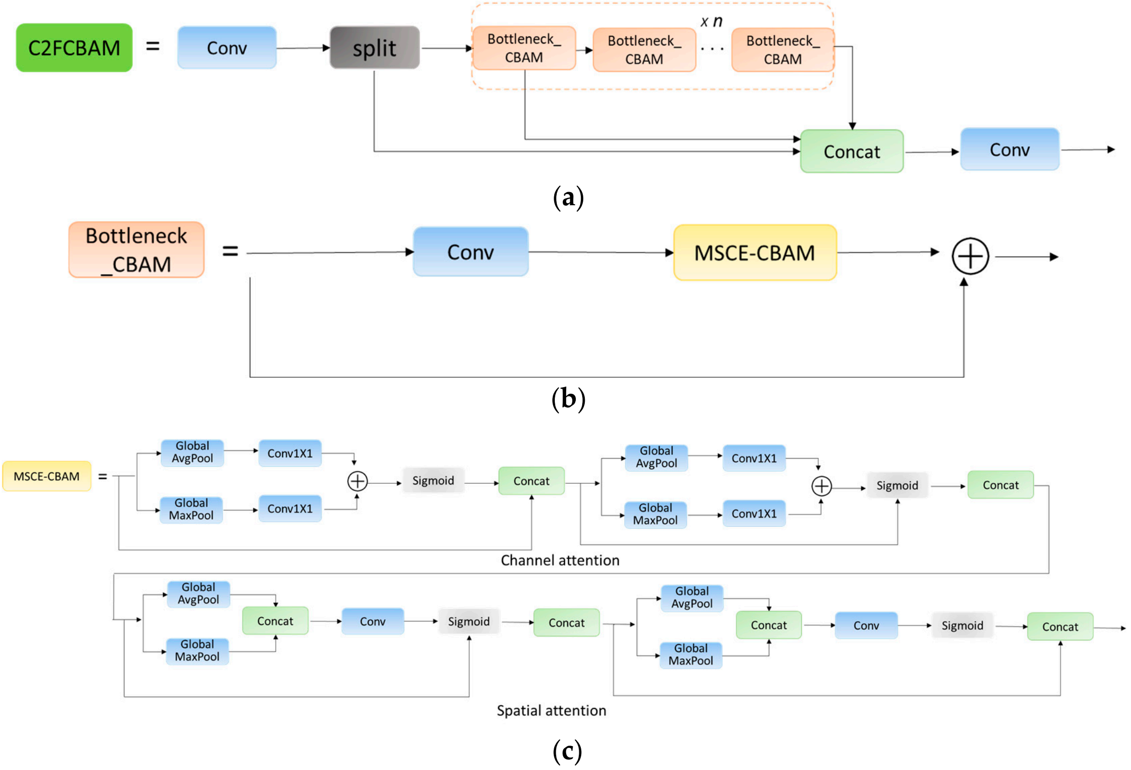


Figure 5. Structure diagram of C2FCBAM and its sub-modules: (**a**) overall structure of C2FCBAM; (**b**) overall structure of Bottleneck_CBAM; (**c**) connection method of MSCE-CBAM.

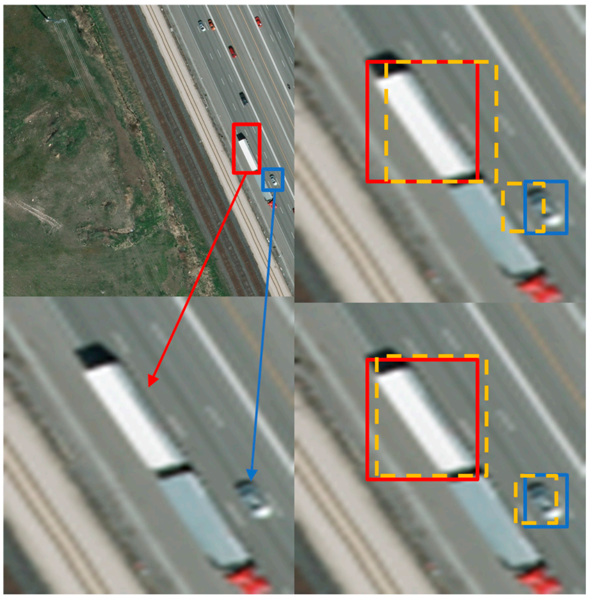


Figure 6. Comparison of sensitivity analysis for detection of object position deviation with varying scales. Red and blue boxes represent ground truth bounding boxes of objects of different sizes, while the yellow dashed box represents the detected bounding box.

We propose a new weighted regression cost function, NWD-CIoU_Loss, based on bounding box loss to solve this problem. This function integrates the NWD_loss function, specifically designed for small targets, with the CIoU function. By allocating a weight distribution coefficient to modify the relative significance of the two losses, the detection performance for minuscule objects is enhanced. The NWD loss function comprises the following steps:

First, to better describe the weight of different pixels within the bounding box, the bounding box is modeled as a two-dimensional Gaussian distribution. The calculation process is as follows: The horizontal bounding box $R = (c_x, c_y, w, h)$, where (c_x, c_y), w, and h represent the center coordinates, width, and height. Its inscribed ellipse equation is

$$\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} = 1 \tag{2}$$

In the equation, (u_x, u_y) are the center coordinates of the ellipse, and δx and δy are the semi-axis lengths along the x and y axes. Thus, $u_x = c_x$, $u_y = c_y$, $\delta_x = w_2$, and $\delta_y = h_2$, $\delta_{\mathrm{y}} = \mathrm{h}_2$. The probability density function of the two-dimensional Gaussian distribution is as follows:

$$f(x|\mu, \sum) = \frac{\exp(-\frac{1}{2}(x-\mu)^T \sum^{-1}(x-\mu))}{2\pi\sqrt{\sum}} \tag{3}$$

where (x, y) represents the coordinates of the Gaussian distribution, u is the mean vector, and Σ is the covariance matrix.

Next, the distribution distance is calculated using the Wasserstein distance from optimal transport theory. For two-dimensional Gaussian distributions, $\mu_1 = N\ (m_1, \Sigma_1)$ and $\mu_2 = N\ (m_2, \Sigma_2)$, the second-order Wasserstein distance between μ_1 and μ_2 is defined as follows:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \left\| \sum_1^{\frac{1}{2}} - \sum_2^{\frac{1}{2}} \right\|_F^2 \tag{4}$$

where $\|\bullet\|_F$ denotes the Frobenius norm.

For bounding box modeling, the Gaussian distributions modeled by the bounding boxes $A = (c_{xa}, c_{ya}, w_a, h_a)$ and B = $(c_{xb}, c_{yb}, w_b, h_b)$ modeled Gaussian distribution Na, Nb, can simplify the second-order Wasserstein distance to

$$W_2^2(N_a, N_b) = \left\| \left(\left[c_{xa}, c_{ya}, \frac{w_a}{2}, \frac{h_a}{2} \right]^T, \left[c_{xb}, c_{yb}, \frac{w_b}{2}, \frac{h_b}{2} \right]^T \right\|_2^2 \tag{5}$$

Since $W_2(N_a, N_b)$ is a distance metric, and the original IoU is a similarity metric for bounding boxes, a new metric called the normalized Wasserstein distance (NWD) is obtained through exponential normalization:

$$NWD(N_a, N_b) = \exp(-\frac{\sqrt{W_2^2(N_a, N_b)}}{C}) \tag{6}$$

Here, C represents a constant closely tied to the dataset, conventionally set to 12.8.

Finally, since IoU_Loss cannot provide gradient transformations for optimizing the network when there is no overlap between the predicted bounding box P and the ground truth G (i.e., $P \cap G = 0$) or when P and G are mutually inclusive (i.e., $|P \cap G| = P$ or G), the NWD metric is designed as a loss function to better deal with small object detection as follows:

$$L_{NWD} = 1 - NWD(N_P, N_g) \tag{7}$$

where N_P is the Gaussian distribution model of the predicted box P, and N_{g} is the Gaussian distribution model of the ground truth box G. In summary, considering the inconsistent distribution of targets of different scales in ground-based scenes, the ratio of NWD to

IoU metrics is set to α: β to achieve better detection of targets of diverse scales. The final bounding box regression loss function is as follows:

$$NWD - CLoU\_loss = \alpha \times NWD\_loss + \beta \times CLoU\_loss \tag{8}$$

where α + β = 1 represents the adjusted weight range of the bounding box loss, considering that small objects are more sensitive to displacement from the center point, and α is set to be greater than β in the experiments.

4. Experiments and Results

4.1. Dataset and Experimental Settings

The VEDAI dataset [28] is an aerial image dataset designed to detect various small vehicles. Each image in the dataset includes visible (RGB) and infrared (IR), with two available resolutions: 1024 × 1024 and 512 × 512. The object size varies between various model sizes, mainly in the small size of the target object (in Figure 7). In our experiments, we mostly use the 1024-resolution visible light version to test the performance of the proposed model. However, we also validated the 512-resolution versions (the abbreviation for VEDAI1024 and VEDAI512). This dataset encompasses small aerial vehicles and showcases a range of diversities, encompassing multiple orientations, variations in illumination and shadows, high reflectivity, and instances of occlusion. Their presence underscores the dataset's complexity and richness in capturing real-world scenarios. The original data consist of 1271 aligned visible and infrared images, focusing on detecting small vehicles in remote sensing images. The image backgrounds encompass complex scenes such as forests, cities, roads, parking lots, and fields. During preprocessing, categories with fewer than 50 instances were removed and classified as "Other". Subsequently, in response to the issue of misalignment between labels and data, images labeled as 0 were excluded from the original dataset, and the experiment transformed annotations of the VEDAI dataset into YOLO format. The center coordinates of the bounding boxes were normalized, and the length and width of the detection boxes were also normalized to [0, 1]. As the center coordinates of the bounding boxes were normalized, the lengths and widths of the detection boxes were also normalized to fall within the range of [0, 1]. In the label processing phase, values of the bounding boxes that exceeded the normalized boundaries were clipped (i.e., we set values less than 0 to 0 and greater than 1 to 1). Ultimately, the model selected 1246 experimental images with corresponding data and labels. The final recognition categories were car, pickup, camping, truck, other, tractor, boat, and van, with class IDs converted to 0, 1, . . ., and 7, respectively, making N = 8. The dataset was split into a training set and a validation set in a ratio of 0.8:0.2, resulting in 996 images for training and 250 for validation.

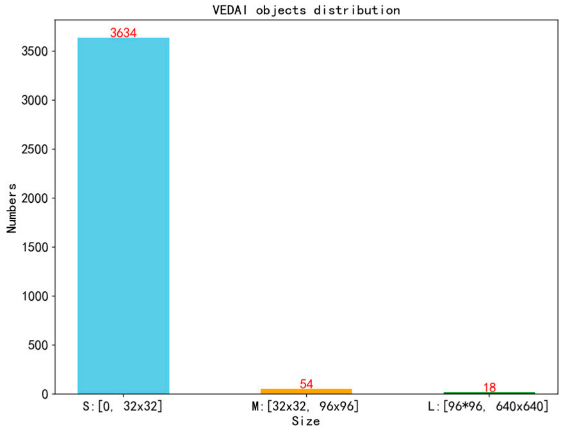


Figure 7. VEDAI objects distribution. S: Number of small target objects (areas less than 32 × 32 pixels); M: number of medium target objects (areas between 32 × 32 and 96 × 96 pixels); L: number of large target objects (areas greater than 96 × 96 pixels).

We used the improved Yolov8s as the primary network framework, with an experimental environment consisting of Ubuntu 20.04 OS, Python 3.8.18, Torch 2.2.1, and Cuda4.11.0. The improved network in this experiment did not utilize pre-trained weights. The hardware setup included two NVIDIA GTX 3090 GPUs, and we modified the experiment's code based on version 8.1.25 of Ultralytics. Throughout the experiment, consistent hyperparameters were maintained for training, testing, and validation. The training epochs were set to 300, with a learning rate of 0.01, momentum of 0.937, a batch size of 64, and input images resized to 640 × 640 for network input. The presented results comprise detection outputs from Yolov8s and the proposed AID-YOLO network, alongside comparative data from relevant referenced papers. Table 1 summarizes the experimental environment and parameter settings.

Table 1. Experimental environment and parameter settings.

Setting	Parameters
CPU Intel	I9-10920X
System	Ubuntu20.04
GPU	RTX3090*2
Python	3.8.18
Torch	2.2.1
Training Epochs	300
Weight_decay	0.0005
Momentum	0.937

4.2. Evaluation Metrics

The experiments evaluate the proposed detection method based on detection performance and model parameter size. The evaluation metrics include precision (P), recall (R), Average Precision (AP), Mean Average Precision (mAP), and millions of parameters (M). Precision is the proportion of correctly predicted targets among all detected targets, while recall measures the proportion of correctly detected targets among all actual targets.

The calculation methods for precision and recall are defined as follows:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

TP and TN represent correct predictions, and FP and FN represent incorrect results. The meanings of these metrics are as follows:

True Positive (TP): The box is correctly classified as a positive sample, and it is indeed a positive sample.

True Negative (TN): The box is correctly classified as a negative sample and, indeed, a negative sample.

False Positive (TN): The box is incorrectly classified as a positive sample, but it is a negative sample.

False Negative (FN): The box is incorrectly classified as a negative sample, but it is a positive sample. It means that it represents the actual object that went undetected.

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{1}{k}\sum_{i=1}^{k} AP_i = \frac{\int_0^1 P(R)dR}{N} \tag{12}$$

AP is the area under the precision–recall (P-R) curve. The closer the AP value is to 1, the better the detection performance of the algorithm. The calculation process of AP can be summarized as follows: mAP is the average of AP values for each class. It is a comprehensive metric used for fairly measuring the performance of multi-class object detection tasks. Therefore, the mAP is used to evaluate the detection accuracy in our experiments. Our metrics include two different Average Precision scores: mAP@0.5 (mAP@0.5 represents the average accuracy at an IOU threshold of 0.5) and mAP@0.5–0.95 (mAP@0.5:0.95 means the average mAP calculated at multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05). Here, P represents precision, R represents recall, and N is the number of recognition categories. Giga Floating-point Operations Per second (GFLOPs) and parameter size (parameters) are used to measure the complexity and computational cost of the model.

In this experiment, the dataset primarily consists of aerial images, where the ground objects vary significantly in scale. To better evaluate the performance of the AID-YOLO model in detecting small, medium, and large objects, we incorporate the evaluation metrics AP_S, AP_m, and AP_l. The specific definitions of these metrics are as follows:

AP_s (Average Precision for small objects): Average Precision for small objects (areas less than 32 × 32 pixels).

AP_m (Average Precision for medium objects): Average Precision for medium objects (areas between 32 × 32 and 96 × 96 pixels).

AP_l (Average Precision for large objects): Average Precision for large objects (areas greater than 96 × 96 pixels).

4.3. Experimental Results

4.3.1. Validation of the Benchmark Framework

The Table 2 comprehensively evaluates the model size and inference capabilities of different baseline frameworks based on the number of layers, parameter size (parameters), GFLOPs, and inference speed (FPS). The detection performance of these models is measured by mAP@0.5. Although YOLOv8X achieved the best detection performance, it has 169 more layers than YOLOv8s (393 vs. 224), its parameter size is 6.13 times that of YOLOv8s (68.1M vs. 11.1M), and its GFLOPs are 9.03 times that of YOLOv8s (257.4 vs. 28.5). In the case of YOLOv8s, its mAP@0.5 is lower than that of frameworks such as YOLOv8x; it is selected as the baseline framework due to its significant advantages in terms of layers, parameter size, GFLOPs, FPS, and overall performance for edge deployment. The experiments above validate the rationality of selecting YOLOv8s as the baseline detection framework.

Table 2. Comparison of model size and inference capability of different baseline YOLO frameworks in VEDAI1024 (RGB).

Method	LAYERS	Parameters (M)	GFLOPs	FPS	mAP@0.5
Yolov5s	193	9.1M	23.8	196	0.632
Yolov6s	142	16.3M	44	200	0.57
Yolov8s	**168**	**11.1M**	**28.5**	**238**	**0.639**
Yolov8l	268	43.6M	164.8	123.4	0.667
Yolov8m	295	25.8M	78.7	158.7	0.662
Yolov8x	365	68.1M	257.4	90	0.68

Bold represents the overall performance display of Yolov8s.

4.3.2. Ablation Experiment

In our ablation study (Table 3), we compared the baseline method using the YOLOv8s model to verify the effectiveness of our various enhancements for detecting small objects. This study progressively combines each optimization measure, and Table 3 presents the detailed results of these experiments. (1) Backbone Improvement: The improved model employs a four-branch skip layer connection and the split operation module RepNCSPELAN4 to replace the C2F module in the YOLOv8s backbone network (YOLOv8s + RepNCSPELAN4); (2) Neck Improvement: An improved C2FCBAM module is combined with the Bottleneck module at the C2F location in the neck network (YOLOv8s + RepNCSPELAN4 + C2FCBAM); (3) Loss Function Improvement: The introduction of the NWD-CIoU_loss function enhances the weight distribution loss for detecting small objects (YOLOv8s + RepNCSPELAN4 + C2FCBAM + NWD-CIoU_loss). Through these three stages of improvement, we developed the proposed AID-YOLO detection model. Each model was evaluated using multiple metrics from the VEDAI dataset. In this experiment, all hyperparameters were kept constant. During training, the input image size was 640 × 640, the batch size was 64, and the training ran for 300 epochs.

Table 3. Comparison table of VEDAI1024 ablation experiment results.

Method	P	R	mAP@0.5	mAP@ 0.5–0.95	Parameter (M)
Yolov8s	0.698	0.559	0.639	0.392	11.1M
Yolov8s+ RepNCSPELAN4	0.559	0.632	0.647↑	0.405	**7.56M**
Yolov8s+ RepNCSPELAN4 + C2FCBAM	0.625	0.609	0.654	0.405	7.59M
Yolov8s+ RepNCSPELAN4 +C2FCBAM + NWD-CIOU_loss	**0.706**	**0.632**	**0.686**	**0.412**	7.59M

Bold represents the maximum or minimum value of the column.

The experimental results are shown in Table 3. The results demonstrate that the proposed modules improved recognition accuracy.

Analysis of RepNCSPELAN4: When employing RepNCSPELAN4 as the feature extraction module in the backbone, substantial enhancements were observed in recognition accuracy and the network model's lightweight nature. The mAP@0.5 metric exhibited a 1.25% increase compared to the baseline, while the mAP@0.5–0.95 metric experienced a 3.32% improvement. Additionally, the parameter count decreased by 32%, achieving a commendable balance between accuracy and parameter efficiency.

Analysis of C2FCBAM: After adding the C2FCBAM module, although the improvement observed on the dataset was relatively modest, the mAP@0.5 metric still showed a slight increase, contributing an increase of 1.1%. It indicates that C2FCBAM enables the network to focus on critical regions favorable for small object detection and enhances multi-scale fusion characteristics. It improves the model's ability to model image features and helps the network better focus on crucial feature parts to improve small object detection accuracy.

Analysis of NWD-CIoU_loss: The most significant improvement in recognition accuracy for the AID-YOLO model was achieved by introducing the NWD-CIoU_loss weight distribution loss function. The mAP@0.5 metric increased by 7.36%, and the strict mAP@0.5–0.95 metric improved the most, with an increase of 5.1%. It indicates the importance of considering the varying scales of objects and analyzing the sensitivity of small objects to positional deviations in this task. This section achieved the best detection effect and improved small object recognition ability when a more extensive weight was applied to the positional deviation term, with α:β approximately equal to 4:1.

Based on the training curves from mAP@0.5 (Figure 8), it is clear that as the RepNCSPELAN4 module, C2FCBAM, and NWD-CIoU loss function modules are successively added during the ablation experiments, the recognition accuracy of the AID-YOLO detector gradually increases. The improvement rates for these additions are 1.25%, 2.35%, and 7.36%, respectively.

Similarly, based on the training curves from mAP@0.5–0.95 (Figure 9), it is evident that the recognition accuracy of the AID-YOLO model increases by 3.32% and 5.1% successively with the addition of the RepNCSPELAN4 module, C2FCBAM, and NWD-CIoU_loss function modules.

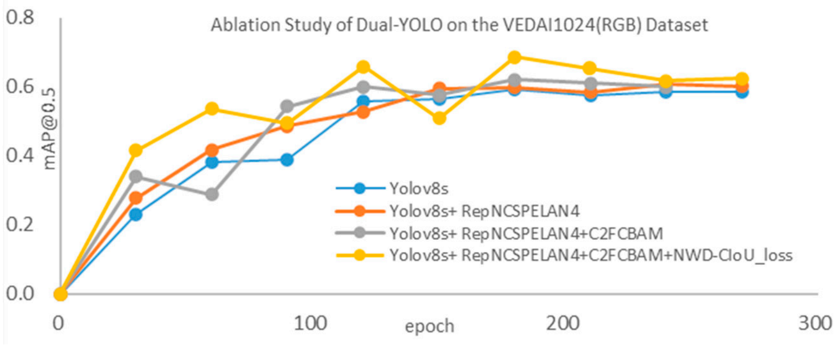


Figure 8. The mAP@0.5 performance curve of AID-YOLO during training.

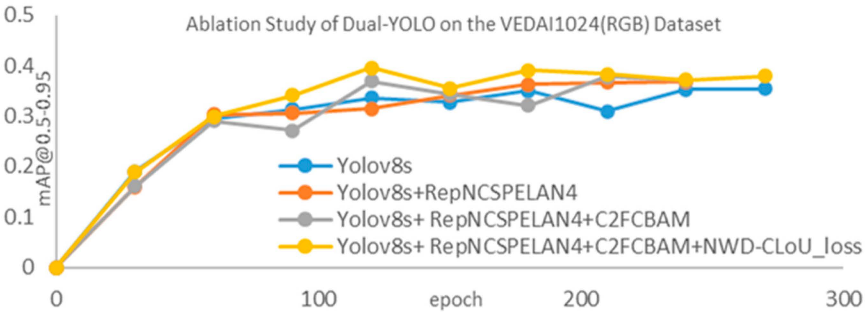


Figure 9. The mAP@0.5–0.95 performance curve of AID-YOLO during training.

4.4. Algorithm Comparison

4.4.1. Numerical Comparisons

We compared our experimental results with YOLO series algorithms, the Yolov8s baseline model, the classic SSD algorithm, and the latest detection models, RT-DETR, Yolov9s, and Yolov10s. The numerical comparison results are presented in Table 4 below. The results for the visual detection performance of the baseline YOLO models and the AID-YOLO are presented in Table 4. It can be observed that this method accurately detects categories of objects that were either not detected or incorrectly predicted by the YOLO series algorithms. Especially for categories such as pickup and car or van and boat, the similarities between these objects can lead to confusion during detection. The model also demonstrates commendable performance for visually challenging and unevenly distributed categories, such as boat and tractor. Table 4 summarizes the performance of the YOLOv5s, YOLOv6s, YOLOv8 series, SSD, YOLOv9s, YOLOv10s, RT-DETR models, and the proposed AID-YOLO across eight recognition categories on the VEDAI1024 dataset. It also comprehensively compares metrics such as mAP@0.5, mAP@0.5–0.95, parameters (M), APs, GFLOPs, and more.

Table 4. Performance comparison of different algorithms in VEDAI 1024(RGB) for object recognition.

Methods	**Car**	**Pick Up**	**Camping**	**Truck**	**Other**	**Tractor**	**Boat**	**Van**	**Map0.5**	**AP$_s$**	**AP$_m$**	**AP$_l$**	**FPS**	**Parameters (M)**	**GFlops**
Yolov5s	0.791	0.73	0.698	0.672	0.433	0.534	0.482	0.712	0.632	0.195	0.287	0.459	196	9.11M	23.8
Yolov6s	0.756	0.687	0.636	0.55	0.47	0.464	0.416	0.585	0.57	0.146	0.267	0.357	200	16.3M	44
Yolov8s	0.828	0.753	0.668	0.677	0.448	0.601	0.447	0.688	0.639	0.202	0.282	0.252	**238**	11.1M	28.5
Yolov8m	0.823	0.764	0.656	0.672	0.486	0.596	0.532	0.766	0.662	0.22	0.303	**0.51**	158.7	25.84M	78.7
Yolov8l	**0.831**	0.796	0.659	0.617	0.533	0.637	0.571	0.69	0.667	0.178	0.294	0.459	132.4	43.61M	164.8
Yolov8x	0.81	**0.805**	0.687	0.628	0.582	0.672	0.516	0.74	0.68	0.186	0.303	0.459	90	68.13M	257.4
Yolov9s	0.844	0.766	0.683	0.626	0.483	0.609	0.413	0.662	0.636	0.195	0.296	0.204	188	**7.29M**	26.7
Yolov10s	0.799	0.691	0.776	0.554	0.52	0.653	0.474	0.735	0.65	0.217	0.287	0.383	250	8.04M	24.5
Rtdetr-resnet50	0.766	0.696	0.639	0.584	0.523	0.609	0.472	0.53	0.602	0.205	0.277	0.51	140.8	41.95M	125.7
SSD	0.755	0.741	0.736	**0.756**	**0.632**	**0.823**	0.459	0.386	0.661	---	---	---	---	---	----
retinannet	0.17	0.37	0.6	0.45	0.5	0.27	0.46	0.03	0.389	0.091	0.283	0.6	46.2	---	---
FCOS	0.78	0.73	0.74	0.55	0.49	0.74	0.56	0.49	0.627	0.161	0.394	0.8	52	---	---
Ghost	0.757	0.637	0.706	0.489	0.361	0.388	0.476	0.711	0.566	---	---	---	263	5.92M	16.1
Yolov8-P2	0.814	0.736	0.71	0.509	0.493	0.538	0.479	0.677	0.62	---	---	---	182	10.6M	36.7
YoloX	0.82	0.79	**0.81**	0.58	0.55	0.74	0.50	0.5	0.674	0.22	**0.422**	0.3	59.95	---	---
AID-Yolo	0.826	0.726	0.701	0.731	0.496	0.649	**0.575**	**0.782**	**0.686**	**0.22**	0.30	0.408	208.3	7.59M	30.6

Bold represents the maximum or minimum value of the column.

The experiments were conducted on the VEDAI1024 dataset by scaling the input image size to 640 × 640, as shown in Table 4. The proposed AID-YOLO model achieved the best mAP@ 0.5, with a value of 0.686. Two categories achieved the highest accuracy among the eight object recognition categories tested. Compared with the baseline method, YOLOv8s, the AID-YOLO model improved the mAP@0.5 from 0.639 to 0.686, representing an overall increase of 7.36%. Additionally, the number of parameters was reduced from 11.1M to 7.59M, a decrease of 31.7%. Furthermore, comparisons with the state-of-the-art models, YOLOv9s and YOLOv10s, showed improved recognition precision of 7.8% and 5.54%, respectively. The model parameters of AID-YOLO increased by only 4% compared to YOLOv9s while decreasing by 5.93% compared to YOLOv10s. Notably, for small object detection, specifically for objects with a pixel area smaller than 32×32, the Average Precision (AP) improved by 8.9% over the baseline model (YOLOv8s), making AID-YOLO one of the best performers among all compared models. Although the model did not achieve the best results for medium and large object detection, the AP for medium target detection was only 1% lower than that of single-stage YOLO series algorithms (e.g., YOLOv8m) while requiring only one-third of the parameters. While some two-stage detection methods (e.g., FOS) performed better for medium and large object detection, their slower inference speeds rendered them less effective for real-time target detection tasks. Although the modified model showed a decrease in frames per second (FPS), it remained above 200, making it suitable for real-time aerial image detection scenarios. In summary, the AID-YOLO model demonstrates comprehensive improvements in accuracy, parameter efficiency, and detection speed, achieving favorable results for aerial image detection tasks.

4.4.2. Heatmap Comparison

To provide a more intuitive visual comparison, Figure 10 presents the feature displays of the original image alongside YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, and AID-YOLO. This section utilizes Grad-CAM heatmaps to visualize the prediction process of the networks, highlighting key feature regions in the specific prediction images generated by the YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, and AID-YOLO detection algorithms. The heatmap feature maps are the same size as the input network images, specifically 640 × 640 pixels. The experimental setup specifies the extraction of features and heatmap displays from the model networks' 9th, 12th, 15th, 18th, and 21st layers, with the configuration set to exclude bounding boxes in the visualized images. The generated heatmaps are shown below:

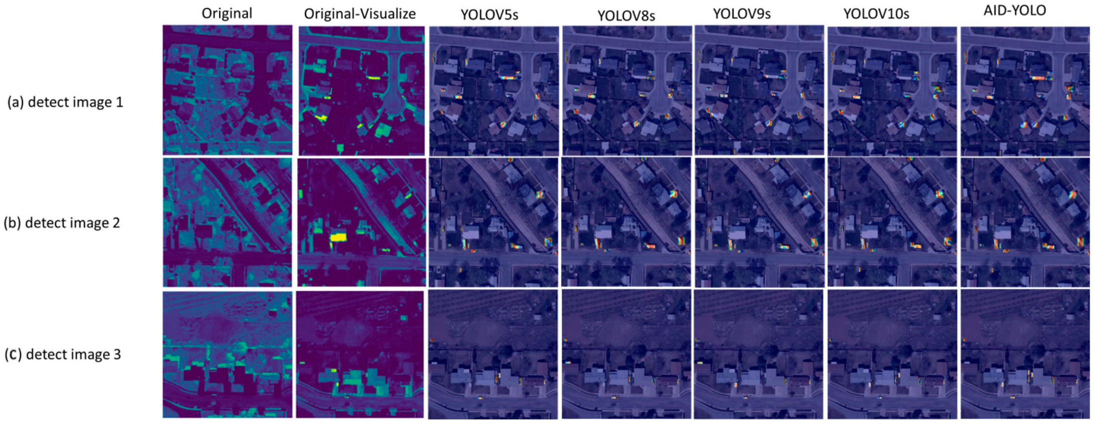


Figure 10. Heatmap comparison between AID-YOLO and other networks. The term "original" refers to the original distribution image, while "original visualize" indicates the output display of the original network. The heatmaps for YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, and AID-YOLO represent the visualization of these algorithms using the best pre-trained weights. (**a**–**c**) represent different test images.

By comparing the visualized images shown in Figure 10, it is evident that the AID-YOLO model represents more obvious object structures, more accurately identifies regions of interest, and highlights distinctive feature areas. Compared to other YOLO series algorithms, AID-YOLO demonstrates significant advantages in recognition performance.

4.4.3. Visualization Comparison

The following analysis highlights the differences in object detection results from images of various scenes, shooting angles, and object types between YOLOv5s, YOLOv8s (baseline model), YOLOv9s, YOLOv10s, and AID-YOLO.

In Figure 11a, under conditions of dense and occluded environments, AID-YOLO successfully identified categories such as boat (light green box), van (dark green box), and camping (orange box), which YOLOv5s, YOLOv8s, or YOLOv9s did not detect. Additionally, YOLOv5s and YOLOv8s incorrectly classified the car (red box) as a pickup (light pink box), while YOLOv10s mistakenly identified the pickup (light blue box) as a car (dark blue box). In Figure 11b, in scenarios with dense objects and similar sizes, AID-YOLO recognized car (red box), camping (orange box), and tractor (green box), which were not detected by YOLOv5s and YOLOv8s, as well as by YOLOv9s and YOLOv10s. In Figure 11c, in low-light conditions with smaller object sizes, the AID-YOLO algorithm successfully identified all categories, including car (red box), tractor (green box), and pickup (light pink box). In contrast, YOLOv5s and YOLOv8s incorrectly classified pickup (light pink box) as a car (red box), while YOLOv9s failed to detect pickup (light blue box), and YOLOv10s did not recognize the tractor category.

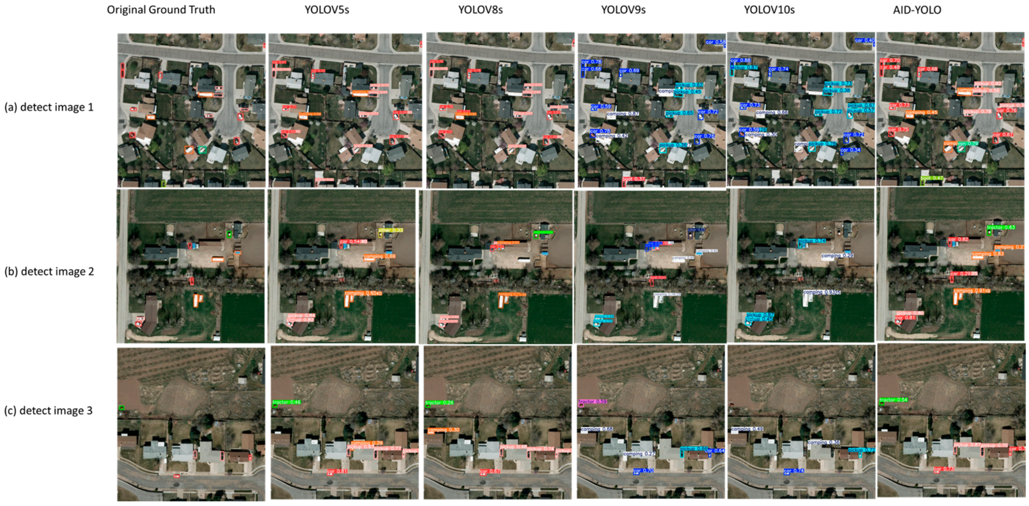


Figure 11. Visual comparison between AID-YOLO and other networks. Original Ground Truth" represents the true annotation information in the original image, with different colors indicating different categories. YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, and AID-YOLO illustrate the visualization of detection results under various model conditions. (**a**–**c**) represent different test images.

4.4.4. Infrared Image Validation Comparison

The AID-YOLO model was also validated on infrared images (IR) from the VEDAI1024 dataset, showing excellent performance. Compared to YOLOv5s, the mAP@0.5 metric improved from 0.609 to 0.633, an increase of 3.94% in recognition accuracy. Compared to the YOLOv8s baseline model, the mAP@0.5 metric (Figure 12) improved from 0.615 to 0.633, an increase of 2.9%, and the mAP@0.5–0.95 metric (Figure 13) improved from 0.364 to 0.383, an increase of 5.21%. These results indicate that the model is also suitable for multi-modal object recognition tasks.

The accuracy comparison curves are shown below:

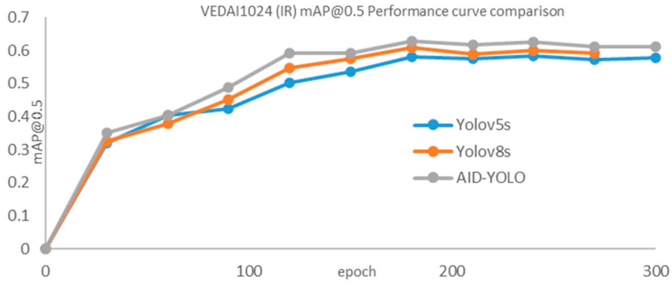


Figure 12. VEDAI (1024) (IR) mAP@0.5 performance curve comparison.

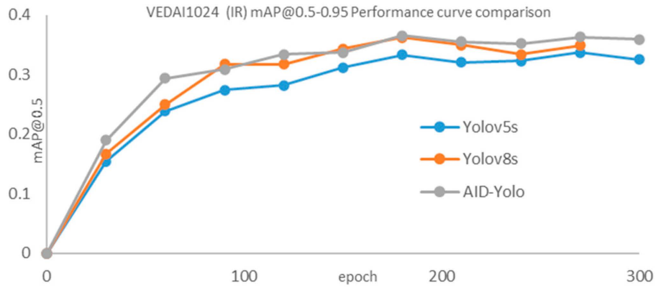


Figure 13. VEDAI (1024) (IR) mAP@0.5-0.95 performance curve comparison.

4.4.5. Verification and Comparison of Images at Different Resolutions

To verify the proposed AID-YOLO model's advantage across different image resolutions, this section uses the VEDAI512 dataset for comparative validation. The study considered other single-stage and two-stage object detection methods, such as SSD, Faster R-CNN (including the VGG16 backbone), RetinaNet [29] (with the ResNet-50 backbone), and EfficientDet [30]. Although the experiment employed different implementation frameworks, we configured all networks with the same parameters to ensure a fair comparison. The input size for all networks was fixed at 512 × 512, with a batch size of 64. The learning rate for all detectors was set to 0.001, with a momentum of 0.9, and it conducted training over a fixed 200 epochs. The results are as follows in Table 5:

Table 5. Performance comparison of different algorithms in VEDAI 512 for object recognition.

Method	Precision	Recall	mAP@0.5
SSD	0.69	0.51	0.543
EfficientDet(D0)	0.58	0.54	0.375
EfficientDet(D1)	0.60	0.68	0.514
Faster R CNN	0.48	0.71	0.509
RetinaNet(50)	0.47	0.59	0.403
AID-YOLO	0.628	0.507	0.592

Experiments demonstrate that the AID-YOLO model is also suitable for object detection tasks with different image resolutions. Although the VEDAI512 dataset has half the resolution of VEDAI1024, and despite a reduction in recognition accuracy, the proposed model still shows a detection advantage in comparison experiments with related algorithms. The mAP@0.5 metric remains higher than some classic algorithms. The AID-YOLO model exhibits relatively low recall rates, attributed to the diverse and imbalanced class distribution within the VEDAI dataset. For instance, the number of cases for the "car" category is approximately eight times that of the "van" category. This disparity results in insufficient sample representation for specific classes, reducing recognition capabilities. Additionally, the model prioritizes improving prediction accuracy, which may sacrifice some of its ability to detect positive samples, thereby contributing to the lower recall rate observed.

4.4.6. Comparison of DOTA-v1.0

To validate the generalization of our proposed network, we compared AID-YOLO with different one-stage or two-stage methods using data from a single modality, including a large-scale dataset for object detection in aerial images (DOTA). The DOTAv1.0 dataset contains rich scene variations in aerial images with extensive target scale and orientation variations, consisting of 2806 large images and 188,282 instances across 15 categories. For the setup, we cropped the images to 1024 × 1024 pixels. Half of the original images were selected as the training set, and the experiment used 1/6 for the validation set. The input size for the images was fixed at 512 × 512, with training epochs set to 100 and a batch size of 16. To validate the superiority of our proposed AID-YOLO, six classic methods were selected for comparison: single-stage algorithms (Yolov8s, RetainNet GFL [31]), a two-stage method (Faster R-CNN), and lightweight models (MobileNetV2 [32] and ShuffleNet [33]. The experimental results are presented in the table below:

As shown in Table 6, the AID-YOLO achieved the best detection results on the DOTA-v1.0 dataset: mAP@0.5. Regardless of whether compared with two-stage, single-stage, or lightweight methods, the model parameters (7.59M) and GFLOP (30.6) are significantly smaller than other algorithm detectors. The above experiments demonstrate that the algorithm possesses both scene detection generalization and meets the requirements for downsizing and lightweight implementation in engineering applications.

Table 6. Performance comparison of different algorithms in DOTA-v1.0 for object recognition.

Method	mAP@0.5	Parameters (M)	GFLOPs
RetainNet [25]	0.504	55.39	293.36
GFL [26]	0.665	19.13	159.18
Faster R-CNN	0.606	60.19	289.25
MobileNetV2 [27]	0.569	10.30	124.24
ShuffleNet [28]	0.577	12.11	142.60
Yolov8s	0.649	11.9	28.5
Yolov10s	0.639	8.04	24.5
Ghost	0.625	9.47	16.1
AID-Yolo	0.653	7.59	30.6

4.4.7. Comparison of SODA-A

This experiment aimed to validate the universality of the proposed AID-YOLO model. To achieve this, we conducted tests using the large-scale dataset SODA-A, which focuses on small object detection in aerial remote sensing images. The SODA-A dataset comprises a rich collection of remote-sensing images with diverse scene transformations and a wide range of target scales and orientations. The original dataset provided by the authors includes 2512 ultra-high-resolution images across ten categories. The original images were cropped into 640 × 640 pixel sub-images in the preprocessing stage. A threshold size of 0.01 was set to remove duplicate or inaccurate labels and merge similar ones. Ultimately, the training set comprised 11,837 images, while the validation set contained 3309 images. The experiments were conducted with 100 epochs and a batch size of 32 without employing any pre-trained network weights. The experimental results are presented in Table 7:

Table 7. Performance comparison of different algorithms in SODA-A for object recognition.

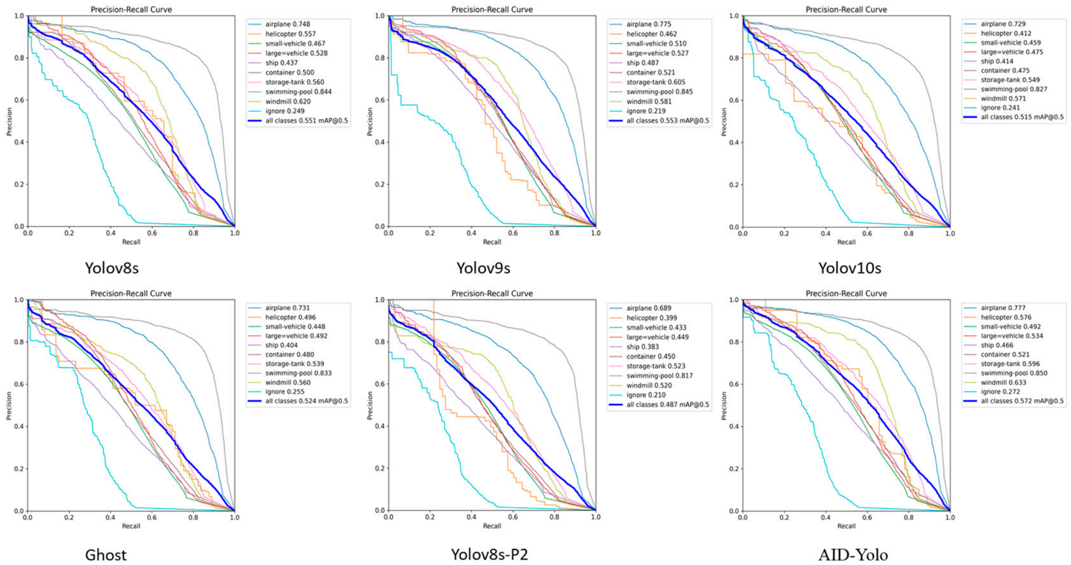
	Yolov8s	Yolov9s	Yolov10s	Ghost	Yolov8s-P2	AID-YOLO
Airplane	0.748	0.775	0.729	0.731	0.689	**0.777**
Helicopter	0.557	0.462	0.412	0.496	0.399	**0.576**
Small-vehicle	0.467	**0.51**	0.459	0.448	0.433	0.492

Table 7. *Cont.*

	Yolov8s	Yolov9s	Yolov10s	Ghost	Yolov8s-P2	AID-YOLO
Large-vehicle	0.528	0.527	0.475	0.492	0.449	**0.534**
Ship	0.437	**0.487**	0.414	0.404	0.383	0.466
Container	0.5	0.521	0.475	0.48	0.45	0.521
Storage-tank	0.56	**0.605**	0.549	0.539	0.523	0.596
Swimming-pool	0.844	**0.845**	0.827	0.833	0.817	**0.85**
Windmill	0.62	0.581	0.571	0.56	0.52	**0.633**
(ignore)	0.249	0.219	0.241	0.255	0.21	0.272
mAP@0.5	0.551	0.553	0.515	0.524	0.487	**0.572**
mAP@0.5–0.95	0.304	0.302	0.284	0.282	0.265	**0.319**
Precision	0.601	**0.629**	0.569	0.583	0.547	0.628
Recall	0.542	0.527	0.519	0.523	0.502	**0.547**

Bold represents the maximum value of the column.

In analyzing the overall results in Table 7 and Figure 14, it was observed that the original dataset underwent a reduction in image resolution during the cropping process, which led to some small target objects being unevenly segmented. Additionally, numerous classes with uneven distribution contributed to the SODA-A dataset's relatively low overall detection accuracy. However, comparative experimental data indicate that testing on this dataset still reflects the advantages of the AID-YOLO model. Specifically, the AID-YOLO model achieved the highest precision for four detection categories, with a mAP@0.5 that exceeds the parameter-comparable YOLOv9s model by approximately 3.4%. Furthermore, the recall rate was also the highest among the compared algorithms. Although the accuracy could have been more optimal, the overall evaluation metrics demonstrated strong performance. Among the other comparison models, YOLOv9s also exhibited good detection results on this dataset, suggesting that the design of the AID-YOLO model, which draws on the YOLOv9 architecture, is reasonably justified.



Yolov8s Yolov9s Yolov10s

Ghost Yolov8s-P2 AID-Yolo

Figure 14. Precision–recall curve of different algorithms on SODA-A.

5. Conclusions

This paper proposes a real-time lightweight end-to-end detection network, AID-YOLO, for small object detection in aerial images. The detector benefits from the design of four-branch skip layer connections and the split operation feature extraction module RepNCSPELAN4, enabling the model to comprehensively improve small object detection performance in lightweight, inference speed, and accuracy enhancement. We integrated the designed C2FCBAM into the model, allowing the network to learn more features while focusing more on detecting small object areas. Furthermore, considering the sensitivity to slight positional deviations of small objects, we devised a weighted allocation regression cost function, namely NWD-CIoU_Loss. By determining the optimal coefficient distribution ratio based on model characteristics, we enhanced the detection of small objects.

In this experiment, with the above improvements, the proposed AID-YOLO achieved 68.6% mAP@0.5 on the VEDAI1024 dataset with lower computational costs, which is 7.36% higher than YOLOv8s; meanwhile, the parameter count is reduced by 31.7% compared to YOLOv8s, achieving a good balance between accuracy and parameter count. In terms of small object detection metrics, the AID-YOLO model shows an improvement of 8.9% in Average Precision (AP) compared to the baseline model (YOLOv8s), positioning it as one of the top performers among all compared models. Additionally, the FPS metric remains suitable for real-time detection in aerial image scenarios. Moreover, comparative experimental results using this model on infrared images and other datasets indicate that the model also has advantages in detection performance and generalization.

Next, we plan to implement aerial image object detection tasks on high-performance embedded platforms in engineering applications to evaluate the proposed method's practical deployment effects. Meanwhile, to better adapt to the impact of light changes on detection effects, we will continue to research multi-modal fusion networks to better meet the requirements of aerial image applications in practical scenarios.

Author Contributions: Conceptualization, J.Z., Z.Z. and Y.L.; methodology, Y.L., J.Z. and S.L.; software, J.Z. and S.L.; validation, J.Z., C.W. and X.Z.; formal analysis, Y.L.; investigation, Y.L., J.Z. and Z.Z.; resources, Y.L., C.W. and X.Z.; data curation, C.W. and X.Z.; writing—original draft preparation, Y.L. and J.Z.; writing—review and editing, J.Z.; visualization, J.Z.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the 173 Basic Strengthening Program under Grant 2023-JCJQ-JJ-0724 and the fund of Laboratory for Advanced Computing and Intelligence Engineering.

Data Availability Statement: The experimental data used for verification in this article can be publicly obtained on the extreme mart, with the identifier: https://downloads.greyc.fr/vedai/ (accessed on 5 September 2024).

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

Abbreviations

AID-YOLO	you only look once based aerial image detector
GELANs	Generalized Efficient Layer Aggregation Networks
ELAN	Efficient Layer Aggregation Networks
C2F	CSPDarknet53 to 2-Stage FPN
C2FCBAM	Convolutional Block Attention Module with Two Convolution Efficient Layer Aggregation Networks
RepNCSPELAN4	Re-parameterization-net with Cross-Stage Partial CSP and Efficient Layer Aggregation Networks
MSCE-CBAM	Multi-Spatial Channel Enhanced Convolutional Block Attention Module
NWD-CIoU_Loss	Normalized Weighted Distance Complete Intersection Over Union
CBS conv2d	Batch normalization, sigmoid linear unit
SPPF	Spatial Pyramid Pooling Fast
BCE	Binary Cross-Entropy
CSPNet	Cross-stage partial network
RepNBottleneck	Re-parameterization Bottleneck without Identity Connection
RepConvN	Re-parameterization Convolution without Identity Connection
RepNCSP	Re-parameterization-net with Cross-Stage Partial
SiLU	Sigmoid linear unit

References

1. Fu, R.; Fan, H.; Zhu, Y.; Hui, B.; Zhang, Z.; Zhong, P.; Li, D.; Zhang, S.; Chen, G.; Wang, L. A dataset for infrared time-sensitive target detection and tracking for air ground application. *Chin. Sci. Data (Chin. Engl. Web Version)* **2022**, *7*, 206–221.
2. Mo, W. Aerial Image Target Detection Algorithm Based on Deep Learning. Master's Thesis, Harbin Institute of Technology, Harbin, China, 2020.
3. Jiao, J.; Zhou, Y.; Ye, Y.; Gao, C.; Gao, X. Research Progress on Object Detection from UAV Perspectives. *J. Image Graph.* **2023**, *28*, 2563–2586.
4. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More diverse means better: Multimodal deep learning meets remote sensing imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4340–4354. [CrossRef]
5. Zhang, J.; Lei, J.; Xie, W.; Fang, Z.; Li, Y.; Du, Q. SuperYOLO: Super Resolution Assisted Object Detection in Multimodal Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5605415. [CrossRef]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
7. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multi-box detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
9. Shehzadi, T.; Hashmi, K.A.; Stricker, D.; Afzal, M.Z. Object Detection with Transformers: A Review. *arXiv* **2023**, arXiv:2306.04670.
10. Gu, A.; Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv* **2023**, arXiv:2312.00752.
11. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO (Version 8.0.0) [Computer Software]. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 26 February 2024).
12. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Sun, X.; Fu, K. SCRDet: Towards More Robust Detection for Small, Cluttered and Rotated Objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8232–8241.
13. Chen, F.; Zheng, Q.; Zhao, Y.; Song, H.; Shin, H. DW-YOLO: An efficient object detector for drones and self-driving vehicles. *Arab. J. Sci. Eng.* **2023**, *48*, 1427–1436. [CrossRef]
14. Jiang, C.; Ren, H.; Ye, X.; Zhu, J.; Zeng, H.; Nan, Y.; Sun, M.; Ren, X.; Huo, H. Object detection from UAV thermal infrared images and videos using YOLO models. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102912. [CrossRef]
15. Li, Y.; Fan, Q.; Huang, H.; Han, Z.; Gu, Q. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. *Drones* **2023**, *7*, 304. [CrossRef]
16. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
17. Wang, F.; Wang, H.; Qin, Z.; Tang, J. UAV target detection algorithm based on improved YOLOv8. *IEEE Access* **2023**, *11*, 116534–116544. [CrossRef]
18. Pan, W.; Wei, C.; Qian, C. Improved YOLOv8s Model for Small Object Detection from Perspective of Drones. *J. Comput. Eng. Appl.* **2024**, *60*, 142–150.
19. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with Transformers. In Proceedings of the 16th European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229. [CrossRef]
20. Yao, Y.; Ai, J.; Li, B.; Zhang, C. Efficient DETR: Improving End-to-End Object Detector with Dense Prior [EB/O]. 2021. Available online: http://arxiv.org/pdf/2104.01318.pdf (accessed on 19 January 2023).
21. Wang, T.; Yuan, L.; Chen, Y.; Feng, J.; Yan, S. PnPDETR: Towards efficient visual analysis with Transformers. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4641–4650. [CrossRef]
22. Roh, B.; Shin, J.; Shin, W.; Kim, S. Sparse DETR: Efficient End-to-End Object Detection with Learnable Sparsity [EB/OL]. Available online: http://arxiv.org/pdf/2111.14330.pdf (accessed on 19 January 2023).
23. Sharma, M.; Dhanaraj, M.; Karnam, S.; Chachlakis, D.G.; Ptucha, R.; Markopoulos, P.P.; Saber, E. YOLOrs: Object detection in multimodal remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 1497–1508. [CrossRef]
24. Yao, Y.; Cheng, G.; Wang, G.; Li, S.; Zhou, P.; Xie, X.; Han, J. On improving bounding box representations for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 1–11. [CrossRef]
25. Luo, J.; Yang, X.; Yu, Y.; Li, Q.; Yan, J.; Li, Y. PointOBB: Learning Oriented Object Detection via Single Point Supervision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 17–21 June 2024; pp. 16730–16740.
26. 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 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
27. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.

28. Razakarivony, S.; Jurie, F. Vehicle detection in aerial imagery: A small target detection benchmark. *J. Vis. Commun. Image Represent.* **2016**, *34*, 187–203. [CrossRef]
29. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
30. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 10781–10790.
31. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Proc. Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
33. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.

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