

Article

YOLOSO: An Improved YOLO-Based Algorithm for UAV to Detect Small Ground Targets

Bo Lang ^{1,*}, Huamin Yang ^{1,*}, Ruoning Xu ² and Hongzhi Li ²¹ School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China; 2023200180@mails.cust.edu.cn² School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China; 2022111278@stu.hit.edu.cn (R.X.); lihongzhi2014@hit.edu.cn (H.L.)

* Correspondence: yhm@cust.edu.cn

Highlights

What are the main findings?

- A YOLOSO small target enhanced detection algorithm is proposed by optimizing YOLOv11, which adds a P2 high-resolution feature layer, improves C3k2 and C2PSA modules (proposing C3k2SO and C2PSASO), and adopts an ED-CBAM attention module together with multi-scale structural optimization, effectively solving the problems of small target feature loss and poor scale adaptability in UAV ground detection.
- Tests conducted on the VisDrone2019-DET dataset demonstrate that the proposed YOLOSO (3.56M parameters, 37.3% mAP50) surpasses representative lightweight YOLO architectures (including YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n) regarding the values of recall, mAP50, and mAP50–95. Meanwhile, its medium-to-large variant YOLOSO-S (14.85M parameters, 45.3% mAP50) attains a more favorable trade-off between computational efficiency and detection accuracy. Experiments on the DOTAv1 dataset further confirm its generalization capability.

What are the implications of the main findings?

- The optimized network structure and core modules provide a new optimization idea for the improvement of lightweight target-detection models, which can be referenced for the improvement of target-detection algorithms in similar small target and complex background scenarios.
- Both the YOLOSO algorithm and its YOLOSO-S variant can satisfy the technical requirements for real-time sensing and detection in pilotless aircraft scenarios like intelligent security, offering dependable technical support for the real-world deployment of UAV-based ground target detection.



Academic Editor: Emmanouel T. Michailidis

Received: 20 April 2026

Revised: 4 June 2026

Accepted: 5 June 2026

Published: 25 June 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Abstract

In response to the challenges in UAV-oriented ground small-object localization and detection, including the easy loss of tiny target features, insufficient scale adaptability, severe interference from complex backgrounds, as well as high missed and false detection rates and the inadequate localization accuracy of the conventional YOLOv11n model in such scenarios, this paper takes YOLOv11n as the basic framework and performs systematic optimization from three aspects, network structure, core modules, and feature enhancement, proposing a lightweight small-object-enhanced detection algorithm named YOLOSO for UAV applications. By introducing a P2 high-resolution feature branch with a stride of 4, a four-scale detection structure consisting of P2–P3–P4–P5 is constructed, which reduces the minimum detection stride from 8 to 4 and alleviates the loss of detailed feature information

for ultra-tiny targets. A bidirectional “top-down + bottom-up” multi-scale feature fusion strategy is utilized to improve the complementation between deep semantic information and shallow detailed features, while the core modules C3k2SO and C2PSASO are optimized and redesigned, respectively; by adjusting the channel compression ratio (0.25 for shallow modules and 0.75 for deep modules in C3k2SO; 0.25 in C2PSASO), optimizing the convolution kernel configuration (combining 1×3 and 3×1 convolutions), increasing the number of attention heads (from 4 to 8), and introducing residual connections with a 1×1 convolutional branch, the refinement and focusing ability of small-object feature extraction are improved. Additionally, an Enhanced Dual-branch Convolutional Block Attention Module (ED-CBAM) is proposed to further suppress background interference. Experimental results on the VisDrone2019-DET dataset demonstrate that the proposed YOLOS0 contains 3.56M parameters and maintains a lightweight structure, attaining P, R, and mAP50 values of 47.2%, 36.8%, and 37.3% in the test set, which are 4.5 percentage points, 4.8 percentage points, and 3.7 percentage points higher than those of the baseline YOLOv11n (42.7%, 32.0% and 33.6%), respectively. Meanwhile, the medium-to-large version YOLOS0-S (14.85M parameters, 45.3% mAP50) reduces the number of parameters by 53.6% compared with the same-scale Rtdetr-L (32.0M) while achieving significantly better performance (37.8% mAP50). Experiments on the DOTAv1 dataset further confirm the generalization of YOLOS0, achieving 62.2% precision and 27.3% mAP50, outperforming all compared YOLO models. Evaluated on the DOTA-v1 dataset, YOLOS0 achieves a feasible FPS of 20.53. Although slightly slower than mainstream lightweight YOLO models, the substantial accuracy gains fully offset the minor inference speed loss, and such performance trade-off is acceptable for practical UAV deployment. Ablation experiments verify that structural optimization (2.8 percentage points mAP50 improvement, from 33.6% to 36.4%) and the proposed C2PSASO (0.7 percentage points mAP50 improvement to 34.3%) and C3k2SO (1.4 percentage points mAP50 improvement to 35.0%) modules all contribute positive performance gains with favorable complementarity. While retaining lightweight characteristics, the model effectively enhances the detection accuracy of small objects in unmanned aerial vehicle scenarios and can provide technical references for practical applications such as remote sensing monitoring and security patrolling.

Keywords: small target detection; YOLO; feature fusion; attention mechanism

1. Introduction

With the in-depth integration of drone technology and computer vision, UAV-oriented ground target detection has evolved into a core supporting technology in fields such as remote sensing monitoring, smart agriculture, security patrols, and emergency rescue [1–3]. Compared with traditional ground monitoring and satellite remote sensing, UAVs, with their advantages of flexibility, low cost, and high-resolution imaging, can quickly acquire ground observation data in complex scenarios and realize the real-time perception and accurate positioning of ground targets [4,5]. However, UAV-view ground target detection is confronted with numerous inherent challenges: an extremely high proportion of small targets (for instance, distant vehicles, pedestrians, and crop seedlings only account for 1–5% of image pixels); sparse feature information that is susceptible to interference from background clutter; significant variations in target scales, where both large close-range targets and small distant targets coexist in the same scene; and complex imaging environments that are greatly influenced by factors such as illumination changes, airflow vibration, and occlusion [6–8]. These problems cause traditional target-detection algorithms to suffer from

missed detection, high false detection rates, and insufficient positioning accuracy in UAV scenarios, making them difficult to meet practical application requirements.

You Only Look Once (YOLO) [9] has been extensively applied in UAV-based ground target-detection tasks, thanks to its end-to-end inference capability and the balanced performance between speed and accuracy [9–11]. As the latest iteration of the series, YOLOv11 further improves feature representation capability and inference efficiency by introducing the C3k2 feature-extraction module, C2PSA attention mechanism, and optimized detection head structure [12]. Nevertheless, the original design of YOLOv11 is primarily tailored for general scenarios, and it still exhibits obvious deficiencies in small-target detection for UAVs: first, the large stride of characteristic extraction layers (with a minimum stride of 8) results in the severe loss of detailed features of small targets during the downsampling process; second, the excessively high channel compression ratio (default $e = 0.5$) makes it challenging to retain the limited feature information of small targets; third, the inadequate refinement of the attention mechanism prevents it from effectively focusing on small-target regions and suppressing interference from complex backgrounds [13,14]. Therefore, it is of important theoretical value and engineering significance to carry out targeted improvements on YOLOv11 according to the characteristics of UAV small targets, so as to enhance its detection performance in low-pixel, strong-interference, and multi-scale scenarios.

As an authoritative benchmark dataset for UAV vision tasks, the VisDrone dataset is composed of 10,209 images that were collected under different altitudes and scene conditions, covering 10 common object categories including pedestrians, vehicles, and bicycles. Among them, small targets (pixel area $< 32 \times 32$) account for more than 40%, which realistically simulates the complex scenarios of UAV-based ground detection [15]. In addition to VisDrone, DOTAv1 is another mainstream large-scale benchmark dedicated to aerial object-detection tasks, which is widely adopted in UAV remote sensing detection research [16]. This dataset contains 2806 high-resolution aerial images collected from diverse geographic regions and shooting perspectives, annotating a total of 188,282 object instances across 15 categories such as airplanes, ships, storage tanks, and vehicles. Different from ordinary UAV datasets focusing on horizontal bounding box detection, DOTAv1 features abundant arbitrarily oriented objects and densely distributed tiny targets, which poses great challenges to multi-scale and rotated object detection algorithms and can comprehensively evaluate the robustness of detection models in complex aerial scenes. Conducting algorithm research and validation based on the above datasets can effectively ensure the practicality and generalization ability of the improved algorithm.

2. Related Work

2.1. Research Progress of UAV-Based Ground Target-Detection Technology

Due to the particularity of scenarios, UAV-based ground target detection has formed a research paradigm of “general detection algorithm adaptation + scenario-specific optimization”. Early research efforts mostly relied on traditional computer vision techniques, including background modeling [17], threshold segmentation [18], and manual feature extraction [19]. However, these methods depend on manually designed feature operators, exhibit extremely poor adaptability to small targets and complex backgrounds, and fail to meet the dynamic requirements of UAV scenarios. CNN-based target detection has turned into the dominant.

As a classic representative of two-phase target-detection approaches, Faster R-CNN first utilizes a Region Proposal Network (RPN) to generate candidate regions and then carries out target classification, as well as bounding box regression [20]. Xiao et al. proposed a multi-resolution detection approach based on a modified R-CNN architecture, which improved the accuracy of ship detection in SAR images with complex resolutions by

adjusting the input image size and optimizing region proposal strategies [21]. Ke et al. integrated deformable convolution into Faster R-CNN to enhance its adaptability to the geometric deformations of ships [22]. Jian et al. designed the SS R-CNN framework, which improved the detection performance of small ship targets by pre-training a feature representation network through a self-supervised learning method [23]. Nevertheless, two-stage algorithms possess high computational complexity and low inference speed, rendering them difficult to meet the real-time detection demands of UAV-based tasks [24].

One-stage target-detection algorithms omit the region proposal step and directly conduct end-to-end target localization and classification, providing a more favorable coordination between processing speed and result accuracy. Algorithms including the YOLO series [9,10,25,26], SSD [27], and RetinaNet have been extensively employed in UAV application scenarios [28]. Yu et al. incorporated the coordinate attention mechanism and bidirectional feature pyramid into YOLOv5, which improved the efficiency of feature fusion and the accuracy of target detection [29]. Miao et al. integrated wavelet decomposition with an improved SSD model to strengthen the detection capability of near-shore ships in complex SAR environments [30]. Yang et al. proposed the ImprovedFCOS algorithm, resolving problems such as small target detection difficulties and misclassification in SAR images through multi-scale feature attention and feature refinement reuse [31]. Due to its capability of efficient feature extraction, the YOLO series is widely recognized as the preferred framework for ground target detection. Its latest versions (e.g., YOLOv11) have further enhanced their adaptability to complex scenarios through architectural optimization [12].

In addition, targeting the particularity of UAV small target detection, researchers have explored multi-modal fusion strategies. Liu et al. [32] put forward the TarDAL model, which integrates infrared and visible light data via target-aware dual adversarial learning to enhance the robustness of target detection in complex scenarios; Shen et al. [33] designed the ICAFusion module, which guides feature fusion based on iterative cross attention to strengthen the semantic consistency of targets in multi-spectral images. These methods mitigate the issue of insufficient small-target features in single-modal data by fusing complementary information from different modalities, but they also encounter difficulties such as increased computational complexity and challenges in modal alignment.

2.2. Research on Small-Target-Detection-Enhancement Technologies

The core challenges in detecting small targets lie in the sparsity of feature information, low signal-to-noise ratio, and high susceptibility to background interference. Existing enhancement technologies mainly focus on three directions: feature enhancement, scale adaptability, and attention mechanisms.

Feature enhancement technologies preserve small target detail information by reconstructing the feature extraction process. Zhang et al. [34] proposed the SuperYOLO model, which introduces super-resolution reconstruction technology to assist target detection and enhance the detailed representation of small targets, though it increases training overhead and relies heavily on the quality of reconstruction results. A cosine similarity-guided module dedicated to feature decomposition and fusion was integrated into YOLOv8 by Guo and his colleagues, with lightweight cross-modal alignment realized through the decomposition of common-specific features [35]. Fei et al. put forward an attention-directed differential fusion mechanism based on YOLOv5, which strengthens both the modeling of complementary features and the guidance of regions with salient targets [36]. A dual-branch asymmetric attention backbone network and a feature fusion pyramid were designed by Wang et al., realizing the mutual enhancement of semantic information and detailed features between the backbone network and the fusion layer [37]. These methods have alleviated the problem of small-target feature loss to a certain extent by optimizing feature

extraction and fusion strategies, but their adaptability to extreme scale variations in UAV application scenarios still needs to be further improved.

Scale adaptability technologies focus on the effective utilization of multi-scale features. Traditional methods construct multi-scale feature maps through Feature Pyramid Networks (FPNs), but deep feature maps have low resolution, and small target features are easily diluted [38]. Liu et al. proposed the Adaptive Spatial Feature Fusion (ASFF) module, which fuses features of different scales by learning adaptive weights to strengthen the uniformity of multi-scale object detection [39]. Zhao et al. put forward MSFA-YOLO, integrating C2fSE and DenseASPP modules to strengthen the representation of multi-scale features [40]. DGSP-YOLO was designed by Zhu et al., which achieves a significant improvement in small-target-detection capability and noise resistance through the embedding of SPDConv, C2fMHSA, and DySample samplers [41]. These methods achieve a better balance between small and large target detection by optimizing the generation and fusion of multi-scale features. However, some of these designs suffer from excessively high computational complexity, which limits their practical application in real-time scenarios.

Attention mechanisms have emerged as a key technology for small target detection, as they can strengthen the feature representation of target regions and suppress interference from complex backgrounds. MAEE-Net was proposed by Li et al., which integrates the Multi-Attention Feature Fusion Module (MAFM) and Edge Feature Enhancement Module (EFEM) into the neck network to boost shallow-layer target features and inhibit background interference [42]. Luo and his collaborators designed SHIP-YOLO, introducing a random attention mechanism and Wise-IoU loss to tackle the detection issues posed by small targets and complex backgrounds [43]. Woo et al. proposed the Convolutional Block Attention Module (CBAM), which improves the discriminative power of features through the synergy of channel attention and spatial attention [44]. These methods are capable of guiding the network to focus on small target regions, but some attention modules suffer from issues such as parameter redundancy and low computational efficiency, making them difficult to be directly applied to UAV edge deployment.

2.3. Research on Improvements to the YOLO Series Algorithms

Owing to their compact structure and superior inference efficiency, the YOLO series has stood out as the predominant framework in UAV-borne target detection, and current improvements are primarily centered on four directions: backbone network optimization, feature fusion enhancement, detection head refinement, and loss function design.

Backbone network optimization is designed to enhance feature-extraction capability. Jocher and his colleagues proposed YOLOv5; the C3 module and Spatial Pyramid Pooling-Fast (SPPF) module enable this model to strike a balance between feature extraction ability and computational efficiency [45]. Ge et al. put forward YOLOX, employing a decoupled head and the SimOTA label assignment strategy to boost detection precision and convergence speed [46]. Wang et al. developed YOLOv7, introducing the E-ELAN architecture and trainable free plugins to further elevate real-time detection performance [10]. Tian et al. presented YOLOv12, strengthening the computational efficiency of attention mechanisms and feature aggregation capability through the Area Attention and R-ELAN modules [47]. These improvements enhance adaptability to complex scenarios by optimizing the feature-extraction process.

Feature fusion enhancement focuses on the effective integration of multi-scale features. PANet [48] enhances the semantic information of shallow features through a bottom-up feature fusion path; BiFPN introduces weighted feature fusion and cross-scale connections to improve the flexibility of feature fusion [49]; the ASFF achieves the refined fusion of multi-scale features through adaptive weight learning and has been widely applied to

improvements of the YOLO series [39]; Zhu et al. introduced the ACmix module into YOLOv8, integrating convolution and self-attention mechanisms to strengthen global context modeling capability [50].

Detection head design aims to enhance the precision of classification and regression. YOLOv8 adopts a disentangled detection head, decoupling the two tasks of classification and regression to alleviate cross-task interference [35]; YOLOv10 introduces a dynamic detection head, improving the detection consistency of multi-scale targets by adaptively adjusting the detection branch structure [10]; YOLOv11, which this paper is based on, adopts C3k2 and C2PSA modules to enhance the perception capability of structured features [12].

Loss function design is mainly devoted to refining bounding box regression and classification performance. Rezatofighi et al. proposed GIoU loss to alleviate the issue caused by non-overlapping candidate boxes [51]. Zheng et al. proposed DIoU loss, accelerating convergence by penalizing the center distance of bounding boxes [52]. CIoU loss further incorporates the scale ratio characteristics of bounding boxes to boost overall regression performance [53]. SIoU loss was proposed by Gevorgyan; angle penalty and shape-aware penalty are introduced to boost the convergence speed and positioning accuracy [54]. Zhang and his colleagues proposed Inner-SIoU loss, integrating the direction awareness of SIoU and the internal scaling mechanism of Inner-IoU to strengthen the robustness of bounding box regression in SAR images [55].

2.4. Summary of Research Status

While existing research has yielded substantial advancements in ground target detection, several limitations persist in the scenario of small-target detection, these issues can be summarized as follows: (1) excessively high channel compression ratios employed in feature-extraction modules tend to induce the loss of sparse and valuable detailed features inherent to small targets; (2) conventional multi-scale feature fusion schemes exhibit inadequate adaptability to drastic scale variations, thereby impeding the balanced detection performance between small and large targets; (3) standard attention mechanisms lack sufficient fine-grained refinement capability, rendering them ineffective in accurately concentrating on small-target regions while suppressing interference from complex backgrounds; (4) certain enhanced detection models are accompanied by elevated computational complexity, which hinders their practical deployment on UAV edge platforms for real-time detection tasks.

To tackle the aforementioned issues, this study employs YOLOv11n as the baseline framework and performs targeted optimization from three perspectives, network architecture, key functional modules, and feature representation enhancement, thereby developing an efficient detection model tailored for small-target detection in UAV scenarios.

3. Materials and Methods

3.1. Overall Algorithm Architecture

Based on the YOLOv11n framework, this study carries out systematic optimization from three dimensions—network structure, core modules, and feature enhancement—to solve the problems of feature loss, poor scale adaptability, and intense background interference existing in UAV small-target detection. Correspondingly, an improved algorithm named YOLOS0 is proposed. The overall architecture of the proposed YOLOS0 adheres to the classic paradigm of “backbone network—feature fusion—detection head”, and its specific structure is illustrated in Figure 1.

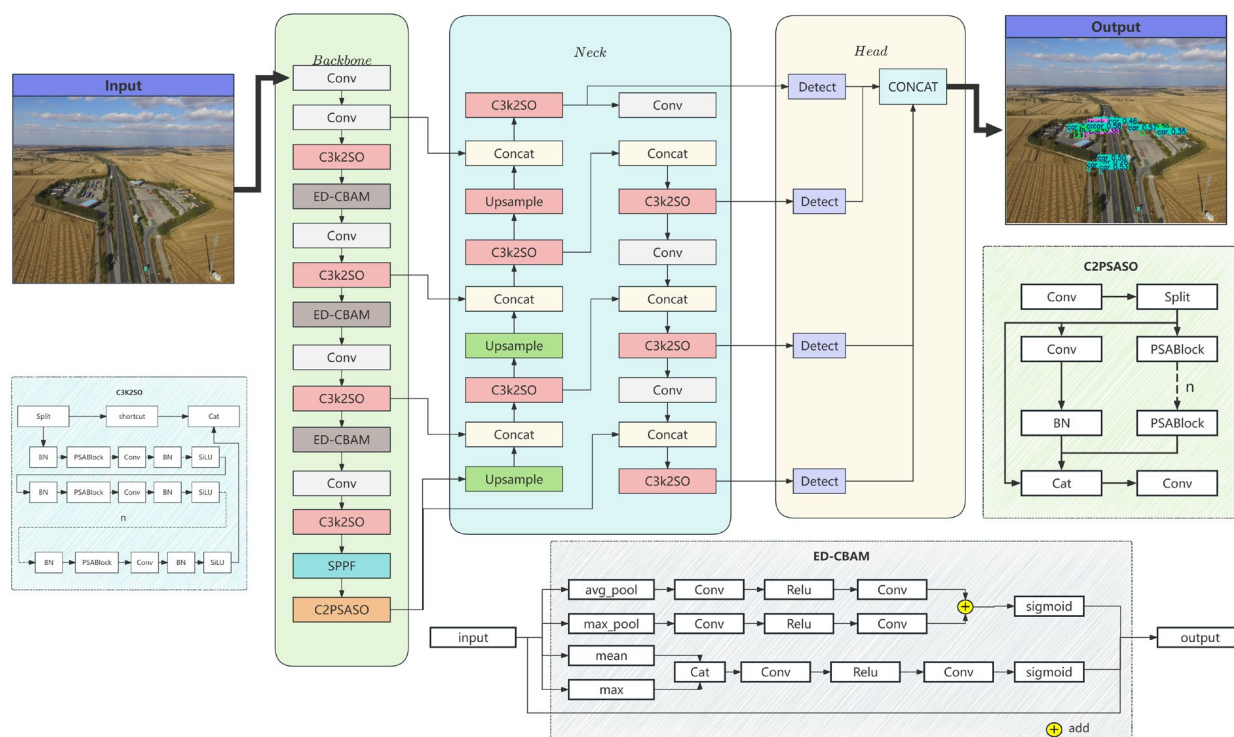


Figure 1. Concrete structural composition of YOLOS0.

3.1.1. Optimization Design for Network Structure

The detection branches of the original YOLOv11n model are P3/P4/P5, with a minimum stride of 8, which poses difficulties in capturing the details of tiny targets. To improve the performance for targets, a high-resolution P2 feature branch (with a stride of 4) is added in this paper, thereby constructing a four-scale detection framework consisting of “P2-P3-P4-P5”. The core merits of this structural optimization are as follows: (1) with a stride of only 4, the feature map resolution of the P2 branch is twice that of the P3 branch, enabling it to preserve more detailed information of small targets; (2) deep fusion of the P2-P3-P4-P5 branches is realized through multi-scale upsampling and feature concatenation, which enhances the feature correlation of targets across different scales; (3) each branch is equipped with a feature-extraction module tailored to small targets, which improves the pertinence of feature representation.

3.1.2. Principle of Feature Fusion

The feature fusion process follows a bidirectional fusion strategy of “top-down + bottom-up” and realizes the complementary enhancement of multi-scale features through Upsample and Concat operations. For any scale feature layer $F_i (i \in \{2, 3, 4, 5\})$, the mathematical expression of its fused feature F_i^{fuse} is as follows:

$$F_i^{fuse} = \text{Concat}(\text{Unsample}(F_{i+1}^{fuse}), F_i^{backbone}) \quad (1)$$

$$F_i^{refuse} = \text{C3k2SO}(F_i^{fuse}) \quad (2)$$

3.2. Optimization Design of Core Modules for Small Target Detection

To address the problems of severe feature loss and severe background interference for UAV small targets, this paper carries out targeted performance optimization on two core modules of YOLOv11, namely C3k2 and C2PSA.

3.2.1. Design of Optimized C3k2SO Module

The conventional C3k2 module has inherent limitations, including excessively high channel compression ratio, single-scale coarse convolution kernels, and deficient attention perception capability, which readily cause feature degradation and information attenuation of small targets. To solve the above issues, this study proposes the improved C3k2SO module.

The C3k2SO module is developed based on the C2f framework and retains its basic structure composed of multi-parallel branches and adaptive feature fusion. By reconfiguring the channel compression ratio, optimizing the convolution kernel combination, and upgrading the attention mechanism, the module achieves remarkable improvement in the feature-extraction capability for small targets. The detailed structure is illustrated in Figure 2.

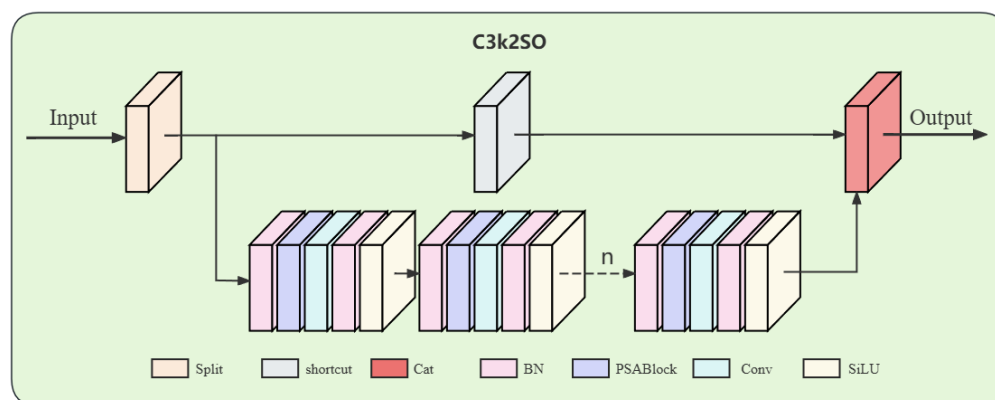


Figure 2. Structure of the C3k2SO module.

To mitigate the channel-wise feature loss of UAV small targets, this work adopts a hierarchical channel compression strategy. We adjust the expansion ratio to 0.25 for shallow network modules and 0.75 for deep network modules, while the original value is set to 0.5. The mathematical expression is as follows:

$$c = c_1 \times e \quad (3)$$

Herein, c_1 signifies the quantity of input channels, whereas c represents the channel dimension of the branch concealed layers. For shallow modules (e.g., P2/P3 branches), $e = 0.25$ maximizes the retention of detailed features of small targets; for deep modules (e.g., P4/P5 branches), $e = 0.75$ balances feature expression and computational efficiency.

We also adopt a strategy of optimized convolution kernel configuration. This strategy adjusts the convolution kernel of the Bottleneck module from 3×3 to a combination of 1×3 and 3×1 . The 1×1 convolutional operation is adopted for feature dimensionality reduction, while the 3×3 convolution is employed to mine local characteristic information. This approach not only lowers the computational cost but also reinforces the capability of capturing local textural details of small objects. The analytical expression of the convolution operation is given as follows:

$$F_{conv} = BN(Conv_{3 \times 3}(BN(Conv_{1 \times 1}(F_{in})))) \quad (4)$$

Herein, F_{in} signifies the input feature tensor of the module, $Conv_{k \times k}(\bullet)$ represents the convolutional computation, and $BN(\bullet)$ indicates the batch normalization procedure.

Within the attention branch, the quantity of attention heads in the PSABlock is adjusted from $\max(c/63, 1)$ to $\max(c/32, 2)$, and the attention channel compression ratio

is reduced from 0.5 to 0.25, so as to strengthen the feature focusing ability on small targets. The attention weight is calculated as follows:

$$\begin{aligned} \text{Attn}(Q, K, V) &= \text{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{head}}}}\right)V \\ d_{\text{head}} &= \frac{c \times \text{attn_ratio}}{\text{num_heads}} \end{aligned} \quad (5)$$

where Q, K, V represent the query, key, and value matrices respectively, d_{head} denotes the dimension of attention heads, $\text{attn_ratio} = 0.25$ is the attention channel compression ratio, and $\text{num_heads} = \max(c // 32, 2)$ is the number of attention heads.

Finally, we add a 1×1 convolutional layer at the end of the attention branch to achieve secondary refinement of features, with the mathematical expression as follows:

$$F_{\text{refine}} = \text{SiLU}(\text{BN}(\text{Conv}_{1 \times 1}(F_{\text{attn}}))) \quad (6)$$

where F_{attn} signifies the feature tensor output by the attention module, and $\text{SiLU}(\bullet)$ stands for the activation function.

3.2.2. Design of Optimized C2PSASO Module

The original C2PSA module suffers from excessive channel compression, insufficient attention head quantity, and the absence of a residual connection structure, which easily trigger gradient vanishing and feature loss of small targets during network forward propagation. A novel improved module named C2PSASO is proposed in this paper, and its complete network structure is demonstrated in Figure 3.

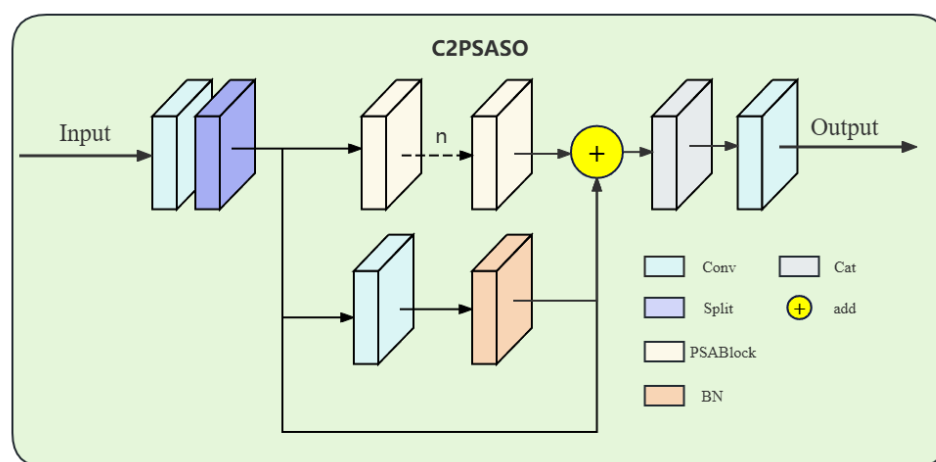


Figure 3. Structure of the C2PSASO module.

We first optimize the channel compression ratio by changing the expansion ratio from 0.5 to 0.25, so as to alleviate the loss of detailed features of small targets. The calculation formula for the number of hidden layer channels is as follows:

$$c = c_1 \times 0.25 \quad (7)$$

We adjusted the number of attention heads of PSABlock from $c // 64$ to $\max(2, c // 32)$ to enhance the capability of capturing fine-grained features. The calculation formula for the number of attention heads is as follows:

$$\text{num_heads} = \max\left(2, \left\lfloor \frac{c}{32} \right\rfloor\right) \quad (8)$$

We compressed the attention channels by adjusting the `attn_ratio` of the PSABlock from 0.5 to 0.25, retaining more attention channels and enhancing the feature representation of small targets:

$$c_{attn} = c \times 0.25 \quad (9)$$

Herein, c_{attn} signifies the effective quantity of channels within the attention mechanism.

We also added a residual connection with a 1×1 convolutional residual branch to strengthen gradient propagation and prevent small-target feature loss. The residual fusion formula is as follows:

$$\begin{aligned} F_{res} &= \text{Conv}_{1 \times 1}(F_b) \\ F_{out} &= \text{SiLU}(\text{BN}(F_m + F_{res})) \end{aligned} \quad (10)$$

Herein, F_b signifies the input feature tensor of the branch, F_m denotes the output feature of the attention module, F_{res} represents the residual feature, and F_{out} indicates the ultimate output of the module.

3.3. Design of Enhanced Dual-Branch Convolutional Block Attention Module (ED-CBAM)

The aforementioned C3k2SO and C2PSASO modules have been fundamentally optimized in terms of convolutional structures and internal attention hyperparameters. Nevertheless, UAV aerial scenarios are characterized by cluttered backgrounds and severe noise interference. A simple single attention mechanism cannot effectively distinguish small foreground targets from redundant background information, making the features of tiny targets easily obscured. To further address the deficiencies of the YOLOS algorithm in fine-grained feature focusing and background interference suppression, this paper performs dual improvements on the conventional Convolutional Block Attention Module (CBAM). Optimizations are implemented from two aspects, the channel branch structure and spatial feature extraction depth, and an Enhanced Dual-branch Convolutional Block Attention Module (ED-CBAM) is proposed.

This module adopts a lightweight design with a controllable increase in parameters and features plug-and-play capability. It can be deployed in the backbone network and feature fusion network. Combined with C3k2SO and C2PSASO modules, ED-CBAM jointly improves the network's ability to capture features of small UAV targets. The overall structure of ED-CBAM is illustrated in Figure 4.

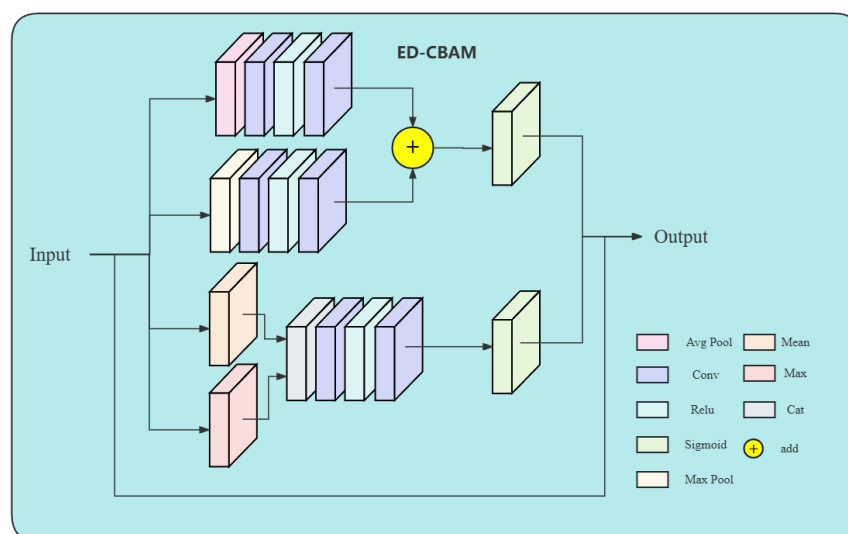


Figure 4. Structure of the ED-CBAM module.

As shown in Figure 4, ED-CBAM retains the serial two-stage optimization architecture of CBAM, namely channel attention followed by spatial attention, while targeted improvements are made to both sub-modules. For the channel attention branch, the traditional single-path pooling mapping structure is replaced with a parallel dual-branch structure. Adaptive average pooling and adaptive max pooling are adopted to collect global statistical information and critical extreme value information of feature channels, respectively. Each branch independently conducts feature compression and nonlinear mapping via two stacked 1×1 convolutional layers. The outputs of the two branches are fused and activated by the Sigmoid function to generate channel attention weights, which enhances the response of important feature channels and suppresses invalid ones.

In the spatial attention branch, the single 3×3 convolutional layer in the original CBAM is replaced by a multi-layer convolutional structure with increased depth. For the channel-weighted feature maps, mean features and maximum features are calculated separately and concatenated along the channel dimension. The fused features are then fed into a two-layer 3×3 convolutional network to explore spatial dependencies between pixels. The deep convolutional structure can extract the fine spatial texture features of small targets and accurately locate tiny targets. Meanwhile, the module integrates an adaptive channel threshold mechanism to dynamically adjust the channel compression ratio. This mechanism solves abnormal operation issues of feature layers with a small number of channels and enhances the general applicability of the module.

In terms of forward propagation, the input features are first processed by the dual-branch channel attention for channel dimension screening and then calibrated by the deep spatial attention in the spatial dimension. Finally, high-quality features optimized in two dimensions are outputted. Compared with the original CBAM, ED-CBAM integrates global and local features, as well as shallow and deep spatial information, and possesses stronger anti-interference performance. It can remarkably improve the utilization of small target features in complex UAV aerial scenarios.

4. Results

4.1. Experimental Environment

The adopted hardware configuration for neural network training in this study is described as follows: the graphical processing unit (GPU) utilizes a single virtual GPU equipped with 48 GB of video memory, yielding a total available video memory of 48 GB; the central processing unit (CPU) is a virtual 20-core Intel® Xeon® Platinum 8470Q processor. The comprehensive hardware resources furnish robust computational capability and sufficient memory support to ensure stable model training.

4.2. Dataset

In this experiment, the VisDrone2019-DET dataset [15] is employed. As a large-scale benchmark for object detection under drone-borne imagery, this dataset was developed by the AI-Eye Team of the Machine Learning and Data Mining Laboratory at Tianjin University. It aims to promote research and development in the field of automatic understanding of drone vision data and provides a comprehensive and rigorous evaluation platform for object-detection algorithms in drone scenarios. This dataset consists of 10,209 static images, together with 288 video segments and 261,908 video frames. All data are acquired by cameras equipped on various drone platforms, covering diverse scenarios across 14 cities in China, among which urban and rural regions serve as the two primary environmental settings. It involves scenes with varying target densities such as sparse and dense distributions, and the collection process is conducted under diverse weather and lighting

conditions. This can fully simulate the complex environmental constraints in real drone operations, demonstrating strong scene representativeness and practicality.

Regarding data annotation, the VisDrone2019-DET dataset employs meticulous manual annotation, with a total of over 2.6 million bounding boxes marked across 12 distinct object categories. Specifically, these categories include ignored regions (ID 0), pedestrians (ID 1), people (ID 2), bicycles (ID 3), cars (ID 4), vans (ID 5), trucks (ID 6), tricycles (ID 7), awning tricycles (ID 8), buses (ID 9), motorcycles (ID 10), and others (ID 11). Among them, there are 10 valid object categories (excluding ignored regions). Pedestrians and cars are the dominant categories, accounting for approximately 35% and 25%, respectively, which conforms to the object distribution characteristics of actual drone observation scenarios. Each annotation entry incorporates comprehensive attribute descriptions, such as the horizontal and vertical coordinates of the bounding box's top-left corner (x, y), as well as its corresponding width and height, together with an evaluation validity flag (1 indicates inclusion in evaluation, 0 denotes exclusion), class ID, target truncation level (ranging from 0 to 2, corresponding to no truncation, partial truncation, and complete truncation in sequence), and occlusion level (ranging from 0 to 2, representing no occlusion, partial occlusion, and heavy occlusion, respectively). These multi-dimensional annotation details provide strong support for the robustness verification and fine-grained performance analysis of algorithms and can meet the training and testing requirements of object-detection algorithms in complex scenarios.

The dataset is partitioned into three distinct subsets: training, validation, and test. Specifically, the training set comprises 6471 images, while the validation set contains 548 images. The test subset is further divided into a standard test set with 1610 images and an extended test set (including test-dev), amounting to 3190 images in total. The test-dev subset provides annotation information and can be used for publishing academic paper results, while the test-set-challenge subset is only for competitions without annotations provided. A key attribute of this dataset lies in its considerable variance in object scales: approximately 31.6% of targets are smaller than 32×32 pixels, while 70% are below 64×64 pixels, thus rendering it a representative dataset for dense small-object detection. Meanwhile, it presents real-scene challenges such as inconsistent image resolutions (common resolutions range from 1360×765 to 2000×1500), target occlusion, motion blur, and uneven illumination. These factors can effectively verify the adaptability and performance upper bound of object-detection algorithms from drone perspectives, and the dataset is widely applied in academic research and algorithm verification for drone-based object detection, small object detection, complex scene adaptation, and related fields. Figure 5 presents several representative annotated samples from the VisDrone2019-DET dataset. These examples vividly illustrate the visualization of bounding box annotations and category labels for diverse objects under different scenarios and intuitively reflect the annotation protocols and object distribution characteristics of the dataset.

To comprehensively evaluate the overall performance of the proposed algorithm and eliminate the contingency of experimental results on a single dataset, so as to enhance the generalization capability and universal applicability of the model in various aerial detection scenarios, this study additionally conducts comparative experiments on the DOTAv1 dataset [16]. As a canonical large-scale benchmark for high-resolution remote sensing aerial object detection, DOTAv1 contains 2806 aerial images collected from diverse geographical scenes and shooting perspectives, covering 15 fine-grained object categories such as airplanes, ships, storage tanks, and large-sized vehicles, with more than 188,000 annotated target instances in total. Different from VisDrone2019-DET that adopts horizontal bounding box annotation, DOTAv1 specializes in oriented object detection, where most targets are arbitrarily arranged with random rotation angles and densely distributed in wide-range

complex backgrounds. Such unique characteristics bring severe challenges for multi-scale feature learning and the high-precision localization of rotated tiny objects. Figure 6 displays typical annotated samples of the DOTAv1 dataset, which explicitly demonstrates the oriented bounding box annotation form and complex target distribution features of remote sensing aerial scenes.

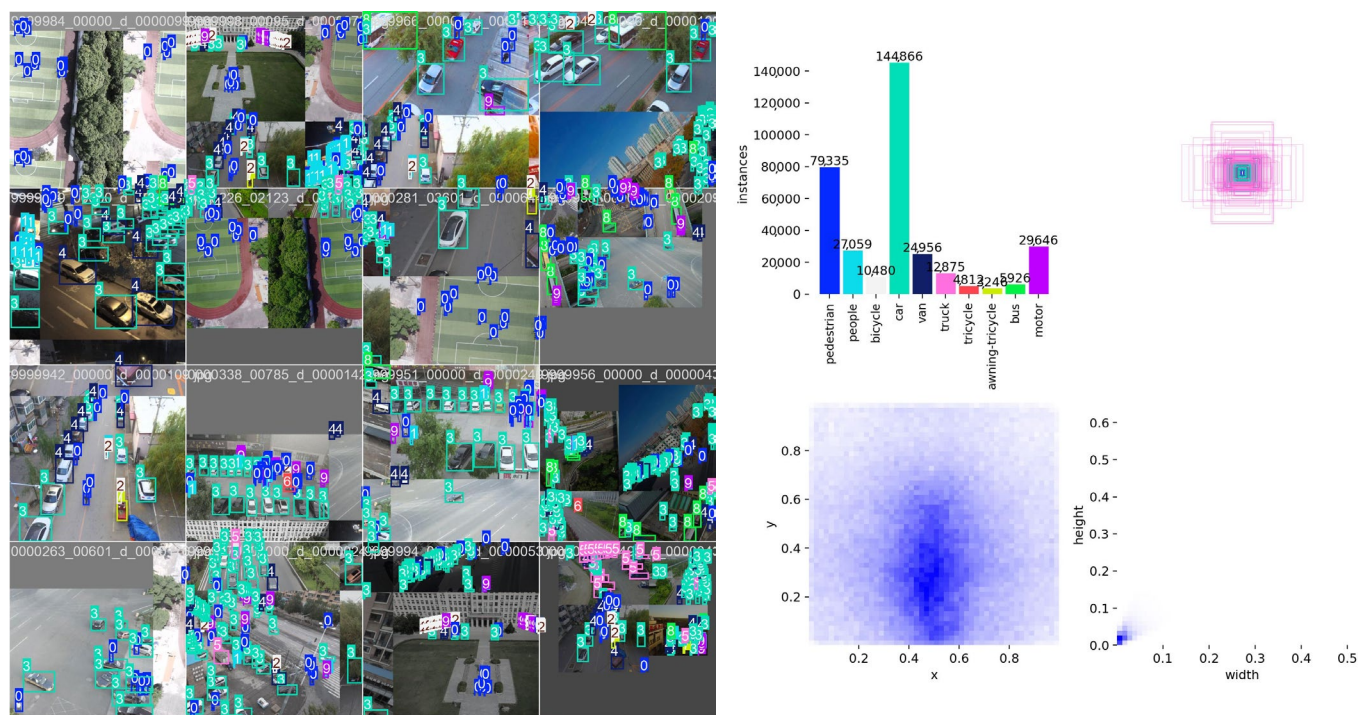


Figure 5. Examples of the VisDrone2019-DET dataset.

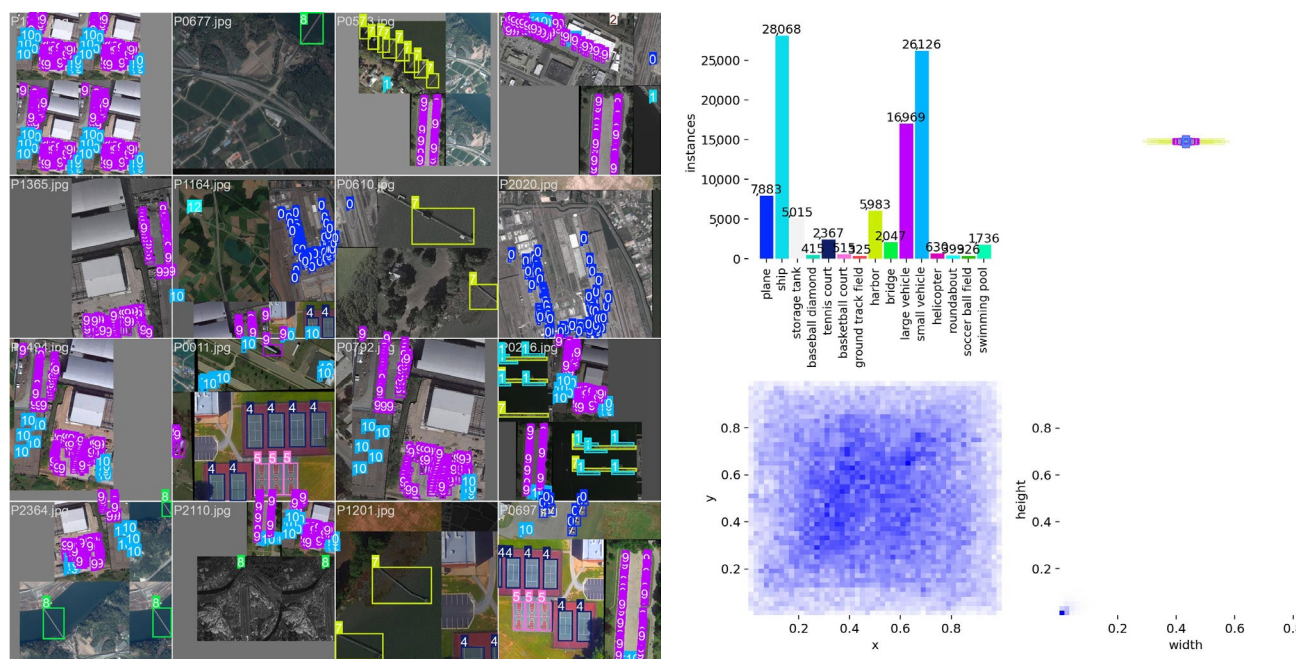


Figure 6. Examples of the DOTAv1 dataset.

4.3. Evaluation Metrics

To comprehensively assess the detection performance and engineering applicability of the proposed model, this experiment adopts six representative metrics widely used in object detection, namely precision (P), recall (R), mean average precision mAP@50, and mean average precision mAP@50–90. The above metrics are used for comparative analysis from four dimensions: detection accuracy, localization accuracy, and model lightweight degree. In addition, GFLOPs (Giga Floating Point Operations Per Second) is introduced as a crucial metric for evaluating the model's computational complexity. It represents the number of floating-point operations (in billions) required for a single forward pass of the model, directly reflecting the demand on computational resources and inference speed. Lower GFLOPs typically indicate higher computational efficiency, which is especially important for deployment on edge devices or real-time systems. The specific definitions, calculation methods, and evaluation significance of each metric are as follows. All metric computations are performed in strict accordance with the universal standards in object-detection research, so as to guarantee the reliability and comparability of the experimental results.

Precision serves as a key indicator for quantifying the accuracy of model detection outputs. It denotes the ratio of true positive samples to all instances predicted as positive by the model, which effectively characterizes the model's capability to mitigate false-positive detections (i.e., misidentifying background or non-target objects as valid targets). This is of great significance for practical application scenarios of YOLO models, such as drone-based small-object detection and remote sensing monitoring, as it helps avoid decision-making errors caused by false-positive outputs. The corresponding calculation formula is expressed as follows:

$$P = \frac{TP}{TP + FP} \quad (11)$$

In the formula, the True Positive (TP) denotes the quantity of samples that are both truly positive and correctly predicted as positive by the model, corresponding to the targets successfully detected. the False Positive (FP) refers to the quantity of samples incorrectly identified as positive despite being negative in reality, namely the background or non-target objects misjudged by the model. Generally, a higher precision value indicates more dependable detection outputs and fewer false-positive errors.

Recall is complementary to precision. It quantifies the model's ability to identify genuine positive samples, i.e., the ratio of actual positive samples successfully detected by the model, reflecting its capacity to reduce missed detections. In object-detection tasks, particularly for small and densely distributed objects, the recall value directly determines whether the model can fully locate all intended targets. Its calculation formula is as follows:

$$R = \frac{TP}{TP + FN} \quad (12)$$

In the formula, the False Negative (FN) denotes the quantity of samples that are truly positive yet misclassified as negative by the model, corresponding to targets that go undetected. A higher recall value signifies fewer omitted targets and stronger capability in identifying actual objects. Notably, precision and recall generally maintain a trade-off relationship: excessive pursuit of higher precision may result in reduced recall, and the converse also holds. Consequently, the model's overall detection capability ought to be evaluated synthetically by integrating these two metrics.

The mean average precision (mAP@50) is evaluated at an IoU threshold of 0.5, focusing on the model's recognition ability under relatively lenient localization criteria. In contrast,

mAP@50–95 refers to the mean average precision computed across a series of IoU thresholds ranging from 0.5 to 0.95 with a step interval of 0.05, that is

$$mAP_{50-95} = \frac{1}{10} \sum_{t=0}^9 mAP(IoU = 0.5 + 0.05 \times t) \quad (13)$$

This metric more rigorously reflects the model's comprehensive detection performance under different localization accuracy requirements and serves as a key indicator for evaluating detection quality.

Parameters (in units of M, meaning millions) serve as a key metric for evaluating the lightweight level of the model. They quantify the total amount of learnable parameters within the model, including weight and bias parameters in convolutional layers, fully connected layers, and other network structures. The scale of parameters is directly associated with the model's memory footprint, training complexity, and inference efficiency: fewer parameters indicate a more lightweight model, which demands less storage space, consumes less GPU memory during training, and achieves faster inference, thus being more applicable to deployment on resource-constrained platforms. Conversely, larger parameter counts usually strengthen the model's feature representation capability, but also lead to higher training difficulty, greater memory and storage consumption, and degraded inference speed. In this work, parameters are counted in millions (M) to enable an intuitive comparison of the lightweight characteristics among various YOLO architectures.

4.4. Comparative Experiments

4.4.1. Comparative Algorithm Setup

To objectively and fairly verify the detection performance and lightweight advantages of the proposed YOLOS model in this paper, a variety of YOLO-series models provided by Ultralytics [56] were selected as the core comparison benchmarks, including YOLOv8n, YOLOv9t, YOLOv10n, and the basic benchmark model YOLOv11n set in this study. In addition, on the VisDrone-DET2019 dataset, comparative experiments are also conducted against current state-of-the-art improved YOLO models (such as SuperYOLO [34] and LS-YOLO [57]). To avoid the distortion of comparison results caused by differences in model scale and ensure that all comparison models are at the same lightweight level, all the above YOLO-series models adopt the “n” (nano) version, with their parameters controlled within 3.5M, which is consistent with the lightweight design goal of the proposed YOLOS model. In addition, to further validate the model performance in medium and large-scale scenarios, the Rtdetr-L model [58] was introduced for comparison. Considering its large number of parameters, the S-version of YOLOS (YOLOS-S) was used for a fair comparison at the same parameter scale to eliminate performance interference caused by model size differences. Furthermore, to verify the generalization of the proposed model, experiments are also carried out on the DOTA-v1 dataset, where comparisons are made only against the official standard models.

To minimize the impact of experimental variables on the comparison outcomes, all models were trained using the same training dataset and a unified training protocol. The specific training configurations are as follows: we uniformly set the batch size to 16, set the initial learning rate to 0.01, and assign a weight decay coefficient of 0.0005 to ease overfitting. The stochastic gradient descent (SGD) optimizer is employed, with the total training epoch fixed at 200. Meanwhile, mainstream object-detection-training strategies, including Mosaic data augmentation, adaptive anchor computation, random cropping, and flipping, are applied equally across all models to guarantee training consistency.

In the testing stage, identical inference parameters are adopted for all models: the confidence threshold is set to 0.25, and the IoU threshold for non-maximum suppression

(NMS) is configured as 0.7. Such settings eliminate disturbances caused by inconsistent training and inference hyperparameters, thereby ensuring the reliability and accuracy of the quantitative comparison results.

The detection outputs of each model corresponding to the input image in Figure 7a are illustrated in Figure 7. It can be observed that the proposed YOLOS0 model achieves high-precision detection toward small vehicles in the scene. To further compare the performance of different models, key indicators including precision (P), recall (R), mAP@50, mAP@50–95, and parameters (M) are compared, so as to quantitatively assess the overall performance of the YOLOS0 model in UAV-borne ground small-object-detection tasks.

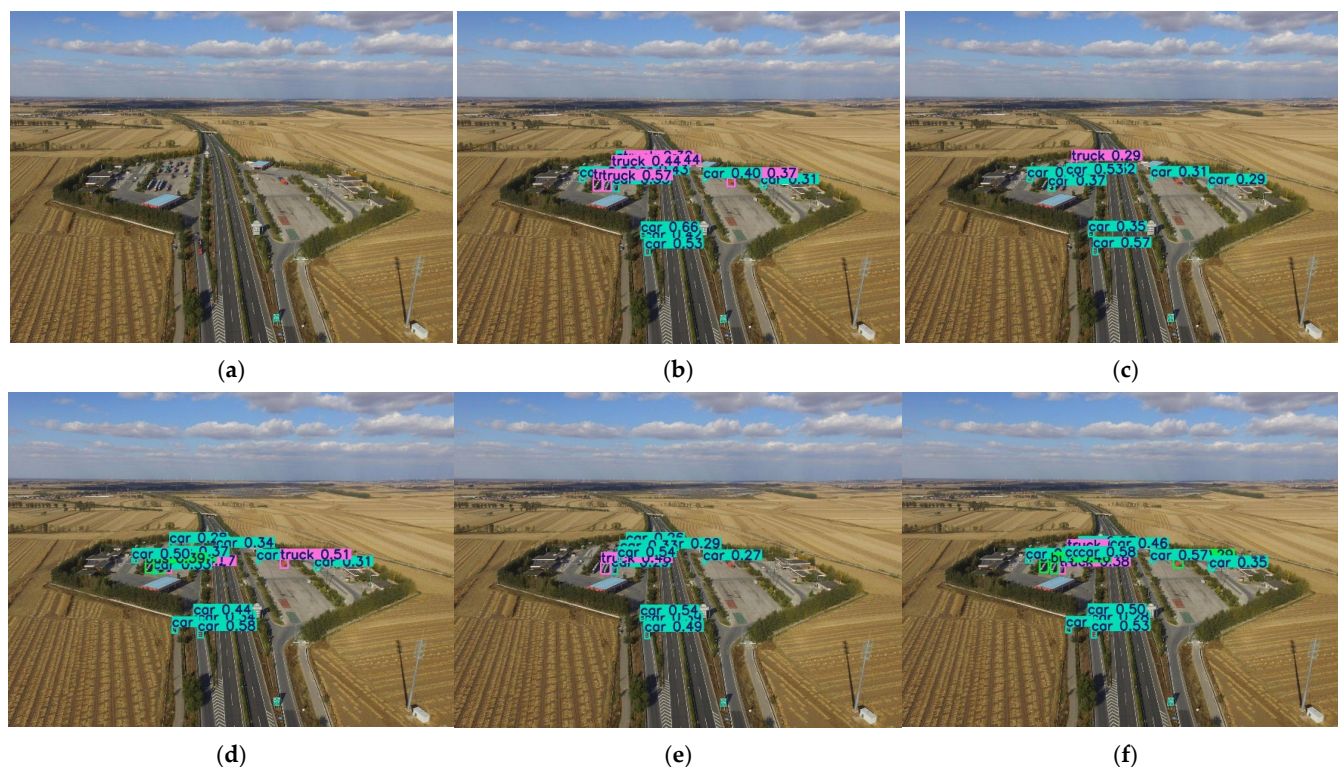


Figure 7. Detection results of YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n, and our proposed YOLOS0 model on a sample input image. (a–f): input image, detection result of YOLOv8n, detection result of YOLOv9t, detection result of YOLOv10n, detection result of YOLOv11n, and detection result of YOLOS0.

4.4.2. Comparison on VisDrone2019-DET

The performance comparison results on the VisDrone2019-DET test set are presented in Table 1. A dimension diagram is plotted based on these experimental results, as illustrated in Figure 8.

The comprehensive quantitative performance of all involved detection models on the VisDrone2019-DET dataset is summarized in Table 1. In this experiment, all comparison baselines are dominated by multiple mainstream lightweight YOLO-n series models, including YOLOv8n, YOLOv10n, and YOLOv11n. The comparison group also contains YOLOv9t, advanced improved detectors (SuperYOLO, LS-YOLO), the high-performance real-time detector Rtdetr-L, the classical two-stage algorithm Faster-RCNN, and our proposed YOLOS0 and its structurally enhanced variant YOLOS0-S. In terms of computational resource consumption reflected by parameters (M) and GFLOPs, the parameter and computational overhead of Faster-RCNN are excessively higher than all other competitors, which makes it inappropriate for horizontal comparison; thus, its corresponding M and GFLOPs data are not listed in the table. Conventional lightweight YOLO-n models maintain extremely low

computational costs, with parameters ranging from 2.59 to 3.01 and GFLOPs fluctuating between 6.5 and 8.8. Such lightweight characteristics enable these basic YOLO-n models to achieve fast inference speed for edge deployment. Nevertheless, enhanced models such as SuperYOLO and LS-YOLO increase network complexity to pursue better feature extraction capability, resulting in a sharp rise in the computational burden, where their GFLOPs reach 20.9 and 42.5, respectively. Different from the lightweight design of mainstream YOLO-n baselines, YOLOSOS-S is not a simplified lightweight model. Instead, it is a complicated and upgraded version derived from the lightweight YOLO-n architecture. By embedding additional feature-enhancement modules and multi-scale detection branches, YOLOSOS-S obtains a larger parameter size of 14.85 and a higher GFLOPs value of 66.5. Although its computational cost is inferior to basic YOLO-n models, it is still more lightweight than the heavyweight Rtdetr-L (103.4 GFLOPs).

Table 1. Quantitative metrics for comparative experiments on VisDrone2019-DET.

Model	M	GFLOPs	P (%)	R (%)	mAP50 (%)	mAP50-95 (%)
Faster-RCNN	-	-	43.3	31.8	31.2	17.0
YOLOv8n	3.01	8.8	42.9	32.9	32.6	19.0
YOLOv9t	2.70	7.9	43.1	32.4	32.3	18.6
YOLOv10n	2.70	8.4	42.4	31.6	31.3	18.4
YOLOv11n	2.59	6.5	42.7	32.0	33.6	19.3
SuperYOLO	7.70	20.9	42.5	31.2	31.4	18.3
LS-YOLO	22.60	42.5	41.4	31.8	31.6	17.1
YOLOSOS(OUR)	3.56	12.4	47.2	36.8	37.3	22.0
Rtdetr-L	32.00	103.4	49.4	38.3	37.8	22.3
YOLOSOS-S(OUR)	14.85	66.5	56.1	43.0	45.3	27.4



Figure 8. Multi-dimensional performance comparison of models on VisDrone2019-DET.

In terms of detection performance, six core evaluation metrics, including precision, recall, mAP50, and mAP50–95, are adopted to evaluate model capabilities for challenging aerial small-target detection. The original YOLO-n counterparts present balanced but mediocre detection performance. The precision and recall of mainstream YOLO-n models are concentrated at approximately 42.4–43.1% and 31.6–32.9%, while their mAP50 and mAP50–95 are limited below 33.6% and 19.3%. The modified detectors SuperYOLO and LS-YOLO fail to achieve effective performance improvement compared with the vanilla YOLO-n series, and partial indicators are even degraded. Our proposed basic YOLOS0 model achieves prominent performance gains on the basis of YOLO-n frameworks, reaching 47.2% precision, 36.8% recall, 37.3% mAP50, and 22.0% mAP50–95, which outperforms all lightweight YOLO-n baselines and other modified detectors. Benefiting from the complicated network structure and optimized detection strategies tailored for UAV scenarios, YOLOS0-S achieves the optimal results across all metrics among all contenders. It achieves the highest precision of 56.1%, recall of 43.0%, mAP50 of 45.3% and mAP50–95 of 27.4%. To conclude, the basic YOLOS0 realizes an excellent trade-off between computational complexity and detection accuracy, which is suitable for general real-time aerial detection tasks. As a high-precision complicated variant, YOLOS0-S sacrifices partial inference efficiency to achieve state-of-the-art detection performance, proving great application value for high-precision UAV target-detection tasks.

4.4.3. Comparison with on DOTAv1

To further validate the effectiveness of the proposed model, we conducted additional experiments on the DOTAv1 dataset, comparing our YOLOS0 series against official YOLO models (YOLOv8n, YOLOv9t, YOLOv10n, YOLOv11n). Table 2 summarizes the comparative results in terms of M, GFLOPs, P, R, and detection accuracy (mAP50 and mAP50–95).

Table 2. Quantitative metrics for comparative experiments on DOTAv1.

Model	M	GFLOPs	P (%)	R (%)	mAP50 (%)	mAP50–95 (%)
YOLOv8n	3.01	8.8	56.3	24.6	24.8	13.7
YOLOv9t	2.70	7.9	58.3	22.2	23.2	12.5
YOLOv10n	2.71	8.4	54.7	20.1	20.1	11.2
YOLOv11n	2.59	6.5	61.2	23.3	23.5	12.6
YOLOS0(OUR)	3.56	12.4	62.2	26.3	27.3	14.9

As shown in Table 2, the proposed YOLOS0 model achieves the best overall detection performance on the DOTA-v1 dataset. In terms of detection precision, YOLOS0 reaches 62.2%, which is higher than all baseline models. Its recall rate is 26.3%, outperforming YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n by 1.7%, 4.1%, 6.2%, and 3.0%, respectively. For comprehensive detection accuracy, the mAP50 and mAP50–95 of our model are 27.3% and 14.9%, ranking first among all compared models.

In terms of model scale and computational overhead, YOLOS0 has 3.56 M parameters and 12.4 GFLOPs, which is slightly larger than the lightweight YOLO series models. The increase in parameters and computation brings a significant improvement in detection accuracy, proving that the optimized structure of YOLOS0 is effective for aerial target-detection tasks on the DOTA-v1 dataset. Although the model has a slight rise in computational complexity, it obtains obvious performance gains and balances detection accuracy and model applicability well.

4.4.4. Comparison of FPS of Different Models

To comprehensively evaluate the real-time detection performance and practical deployment capability of the proposed YOLOS0 model, we further test and compare the

Frames Per Second (FPS) of all comparative models on the DOTA-v1 dataset. Consistent with the above comparison experiments, we select the mainstream lightweight YOLO series models including YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n as the baseline models. All models are tested under the same experimental hardware environment and parameter configuration to ensure the fairness and credibility of the FPS comparison results. The real-time inference speed of each model is statistically analyzed, and the differences in deployment efficiency and detection latency between the YOLOS0 model and other baseline models are quantitatively discussed.

The detailed FPS, parameter quantity (M), computational complexity (GFLOPs), and detection accuracy (mAP50) of each model are listed in Table 3. It can be observed that the four baseline lightweight models maintain excellent real-time inference performance, with FPS values ranging from 26.53 to 35.8. Specifically, YOLOv10n achieves the highest FPS of 35.8, possessing the fastest inference speed among all comparison models. Compared with the baseline models, the proposed YOLOS0 model has a slightly reduced inference speed, with an FPS of 20.53. This slight drop in real-time performance is mainly attributed to the increased model parameters and computational overhead brought by the optimized structural design.

Table 3. Quantitative comparison of parameters, computational complexity, accuracy, and inference speed of different models on DOTA-v1.

Model	M	GFLOPs	mAP50 (%)	FPS
YOLOv8n	3.01	8.8	24.8	27.47
YOLOv9t	2.70	7.9	23.2	27.6
YOLOv10n	2.71	8.4	20.1	35.8
YOLOv11n	2.59	6.5	23.5	26.53
YOLOS0(OUR)	3.56	12.4	27.3	20.53

Nevertheless, the moderate reduction in FPS is completely acceptable for practical aerial target-detection deployment scenarios. Compared with all baseline models, YOLOS0 achieves a significant accuracy breakthrough, with its mAP50 reaching 27.3%, which is 2.5%, 4.1%, 7.2%, and 3.8% higher than that of YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n, respectively. The substantial improvement in aerial detection accuracy far compensates for the minor loss of inference speed. In actual UAV remote sensing and aerial monitoring deployment tasks, high-precision target detection is the core demand, and the FPS of 20.53 can fully meet the real-time working requirements of conventional aerial detection scenarios. Therefore, the YOLOS0 model realizes an effective trade-off between detection accuracy and inference efficiency, and the slight efficiency drop is reasonable and deployable in practical applications.

4.5. Ablation Experiments

To clarify the specific contributions of the proposed optimization strategies, including the C2PSASO module, the C3k2SO module, the ED-CBAM module, and the overall structural modifications of the model, to detection performance and to further validate the rationality of the optimized design of the YOLOS0 model, a set of controlled ablation experiments were performed based on the baseline YOLOv11n network. The experimental design focused on single optimization strategies and combined optimization strategies. By comparing the number of parameters, GFLOPs, precision (P), recall (R), mAP50, and mAP50–95 across different experimental schemes, the effectiveness of each optimized module was quantified. The experimental design is shown in Table 4.

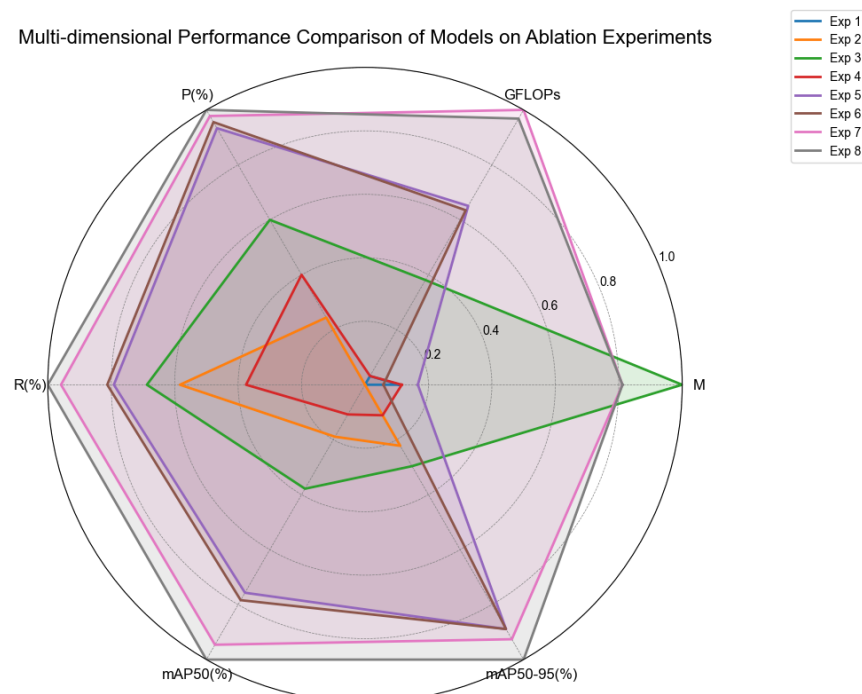
Table 4. Ablation settings.

Experiment No.	Experimental Setup
1	YOLOv11n
2	Based on YOLOv11n, only the C2PSA module is modified to C2PSASO
3	Based on YOLOv11n, only the C3k2 module is modified to C3k2SO
4	Based on YOLOv11n, only the ED-CBAM module is added
5	Only the model structure is modified
6	On the basis of Experiment 5, the C2PSA module is modified to C2PSASO
7	On the basis of Experiment 6, the C3k2 module is modified to C3k2SO
8	On the basis of Experiment 7, the ED-CBAM module is added

The experimental performance of each ablation group on the VisDrone2019-DET test set is presented in Table 5. The five-dimensional graph generated from these results is displayed in Figure 9.

Table 5. Quantitative metrics of ablation experiments based on ablation experiments.

Experiment No.	M	GFLOPs	P (%)	R (%)	mAP50 (%)	mAP50–95 (%)
1	2.59	6.5	42.7	32.0	33.6	19.3
2	2.44	6.3	43.8	34.8	34.3	19.9
3	3.82	8.7	45.4	35.3	35.0	20.1
4	2.60	6.5	44.5	33.8	34.0	19.6
5	2.67	10.4	46.9	35.8	36.4	21.7
6	2.52	10.3	47.0	35.9	36.5	21.7
7	3.56	12.6	47.1	36.6	37.1	21.8
8	3.56	12.4	47.2	36.8	37.3	22.0

**Figure 9.** Multi-dimensional performance comparison of models based on ablation experiments.

As shown in Experiment 2, replacing the original C2PSA with the proposed C2PSASO module slightly reduces model parameters (from 2.59 M to 2.44 M) and GFLOPs (from 6.5 to 6.3). Meanwhile, all detection metrics are improved: precision rises from 42.7% to 43.8%, recall increases from 32.0% to 34.8%, mAP50 goes up from 33.6% to 34.3%,

and mAP50–95 climbs from 19.3% to 19.9%. This demonstrates that the C2PSASO module can streamline model computation while enhancing feature-extraction capability.

Experiment 3 adopts the C3k2SO module individually. It brings a notable performance gain: precision reaches 45.4%, recall 35.3%, mAP50 35.0%, and mAP50–95 20.1%. Nevertheless, this module introduces more parameters (3.82 M) and higher computational overhead (8.7 GFLOPs), indicating its strong feature representation ability at the cost of moderate increased complexity.

By introducing the ED-CBAM attention module in Experiment 4, the model achieves better overall performance than the baseline YOLOv11n. The total parameters reach 2.6 M, which is only slightly higher than the baseline value of 2.59 M. It can be seen that using ED-CBAM alone will not lead to excessive expansion of the model size.

Experiment 5 only optimizes the overall model structure. Compared with the baseline, it achieves a substantial performance leap: precision, recall, mAP50, and mAP50–95 are promoted to 46.9%, 35.8%, 36.4% and 21.7% respectively. Although GFLOPs rises to 10.4, the parameter volume only increases slightly to 2.67 M, proving that structural optimization is a highly efficient way to boost detection accuracy with limited extra computation.

On the basis of structural modification, Experiment 6 further integrates the C2PSASO module. The parameters and GFLOPs decline marginally, and detection indicators see slight growth, which verifies the good compatibility between structural redesign and the C2PSASO module.

Experiment 7 combines structural optimization, C2PSASO, and C3k2SO modules. The detection performance is further elevated, and the overall parameters and computation are well controlled compared with the single use of C3k2SO. It proves that the joint application of multiple modules can balance model complexity and detection accuracy.

Experiment 8 is the complete YOLOS model integrating all proposed strategies. It obtains the optimal overall performance across all groups, with precision of 47.2%, recall of 36.8%, mAP50 of 37.3% and mAP50–95 of 22.0%. The parameters and GFLOPs remain at a reasonable level.

In summary, each designed module and structural improvement contributes positively to detection performance. The combination of all optimization strategies achieves mutual complementation, enabling the YOLOS model to obtain superior detection results while maintaining acceptable model scale and computational cost.

4.6. Experimental Summary

This experiment focuses on verifying the performance of the proposed YOLOS object-detection model. All experimental trials were carried out in a hardware environment configured with a single vGPU featuring 48 GB of video memory and a 20-core virtual Intel® Xeon® Platinum 8470Q processor. Taking the VisDrone2019-DET drone small-object detection dataset as the benchmark, precision, recall, mAP50, mAP50–95, GFLOPs, and the number of model parameters were selected as core evaluation metrics. The effectiveness of the proposed model was verified through comparative experiments and ablation experiments. The dataset comprises 10,209 static images and 261,908 video frames, with over 2.6 million annotated bounding boxes covering 10 valid object categories. Among these objects, 31.6% are smaller than 32×32 pixels, making it a typical small-object dense dataset that can effectively simulate real UAV operating scenarios.

In comparative experiments, the proposed YOLOS model is compared with representative lightweight YOLO architectures including YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n, all of which have fewer than 3.5 M parameters, together with SuperYOLO, LS-YOLO, the medium-to-large-scale model RT-DETR-L, and Faster-RCNN. All comparison models are trained and inferred under consistent experimental settings. The results indi-

cate that the YOLOSO model contains 3.56 million parameters and achieves 12.4 GFLOPs, slightly exceeding those of other lightweight counterparts, yet presents remarkable superiority in detection performance. Specifically, it achieves a precision of 47.2%, a recall of 36.8%, and mAP50 of 37.3%, corresponding to 4.5 percentage points, 4.8 percentage points, and 3.7 percentage points increments relative to YOLOv11n, respectively. Furthermore, on the DOTAv1 dataset, YOLOSO achieves 62.2% precision and 27.3% mAP50, outperforming all compared YOLO models. In terms of real-time inference performance, although YOLOSO obtains a relatively lower FPS of 20.53 compared with the baseline lightweight YOLO models (27.47 for YOLOv8n, 27.6 for YOLOv9t, 35.8 for YOLOv10n, and 26.53 for YOLOv11n), due to the increased parameter scale and computational complexity, the moderate FPS reduction is completely acceptable for practical aerial deployment scenarios. The significant accuracy improvement substantially compensates for the slight inference speed loss, and the FPS of 20.53 fully meets the real-time working requirements of conventional UAV remote sensing and aerial monitoring tasks, achieving a well-balanced trade-off between detection accuracy and deployment efficiency. The YOLOSO-S variant, which comprises 14.85 million parameters and 66.5 GFLOPs, achieves the optimal results across all metrics, with 56.1% precision, 43.0% recall, 45.3% mAP50, and 27.4% mAP50–95, thereby realizing a favorable trade-off between lightweight characteristics and detection accuracy. Ablation studies further verify the efficacy of each individually improved module. Replacing C2PSA with the C2PSASO module alone reduces the number of parameters to 2.44 million and GFLOPs to 6.3, while improving mAP50 by 0.7 percentage points. Introducing the C3k2SO module alone improves mAP50 by 1.4 percentage points (to 35.0%), and overall structural optimization improves mAP50 by 2.8 percentage points (to 36.4%). The YOLOSO model, which combines all three improvements, achieves the best overall performance, with mAP50 increased by 3.7 percentage points (to 37.3%) compared with the baseline YOLOv11n.

Extensive experimental results adequately validate the rationality and superior performance of the proposed YOLOSO model. Relative to mainstream detection frameworks, YOLOSO demonstrates more favorable detection accuracy and flexible deployment potential in UAV-borne small-object detection scenarios. Benefiting from the C2PSASO module, the C3k2SO module, the ED-CBAM module, and global structural optimization, the model achieves a desirable balance between lightweight architectural design and high-precision inference. This work can serve as an effective reference for the advancement and optimization of lightweight object-detection models.

5. Discussion

Experimental results show that the proposed YOLOSO model outperforms YOLOv8n, YOLOv9t, YOLOv10n, SuperYOLO, LS-YOLO, and the baseline YOLOv11n by a large margin in core metrics including precision, recall, and mAP50. In terms of real-time inference performance, although YOLOSO obtains a relatively lower FPS of 20.53 compared with the baseline lightweight YOLO models (27.47 for YOLOv8n, 27.6 for YOLOv9t, 35.8 for YOLOv10n, and 26.53 for YOLOv11n), due to the increased parameter scale and computational complexity, the moderate FPS reduction is completely acceptable for practical aerial deployment scenarios. The significant accuracy improvement substantially compensates for the slight inference speed loss, and the FPS of 20.53 fully meets the real-time working requirements of conventional UAV remote sensing and aerial monitoring tasks, achieving a well-balanced trade-off between detection accuracy and deployment efficiency. The primary causes for such performance improvement are summarized as follows. First, the newly added P2 high-resolution feature branch reduces the minimum detection stride from 8 to 4, effectively alleviating the loss of detailed features of small objects in the downsampling pro-

cess. This is highly consistent with the conclusion in existing studies that “high-resolution feature layers are crucial for improving small-object detection performance”. Furthermore, this paper constructs a four-scale detection framework of “P2-P3-P4-P5”, which strengthens detection consistency for multi-scale objects and compensates for the shortcoming that a single high-resolution branch can hardly maintain satisfactory performance for large objects. Second, the collaborative optimization of the two core modules, C3k2SO and C2PSASO, reduces feature loss of small objects while enhancing fine-grained feature extraction and attention focusing by adjusting channel compression ratios (0.25 for shallow modules and 0.75 for deep modules in C3k2SO; 0.25 in C2PSASO), optimizing convolution kernel configurations (combining 1×3 and 3×1 convolutions), and refining attention head numbers (from 4 to 8 in C3k2SO; from 4 to 8 in C2PSASO). Compared with existing methods that only optimize a single module, the proposed strategy is more systematic and comprehensive and achieves a more favorable balance between feature extraction efficiency and accuracy.

From the ablation experiments, the overall structural optimization contributes the most to performance improvement, with mAP50 increased by 2.8 percentage points higher than the baseline model (from 33.6% to 36.4%). This shows that the rationality of the network structure is a key factor determining small-object-detection performance. The C3k2SO module provides more significant accuracy gains, while the C2PSASO module shows obvious advantages in lightweight design. Their synergy enables YOLOS0 to strike a balance between lightweight architecture and high precision, which is of great importance for on-board deployment on UAVs. UAV platforms are typically constrained by their limited computing power and memory resources and cannot support excessively large models. Although the parameter count of YOLOS0 is around 3.56M, slightly higher than other lightweight YOLO models, it achieves remarkable performance improvement. Meanwhile, the medium-to-large version YOLOS0-S has 14.85M parameters, which is 53.6% fewer than Rtdetr-L (32.0M), further verifying the rationality of the proposed lightweight optimization strategy. In comparison with related works, the improvement strategies in this paper are more targeted. Most existing studies focus on single-dimensional optimization, while this work conducts systematic improvements from network structure and core modules, which better matches the characteristics of UAV-based small objects: sparse feature representation, significant scale variations, and strong background clutter. It thus alleviates the prevalent problems of high miss detection and false detection in such scenarios.

Nevertheless, this study also has certain limitations. First, experiments are only validated on the single VisDrone2019-DET dataset and DOTAv1 datasets. Although it covers various complex scenarios, it cannot fully represent all UAV application environments. Detection performance under harsh conditions such as heavy rain, dense fog, and high-altitude areas has not been verified, leaving room for improving model generalization. Second, despite its lightweight design, YOLOS0 still requires further acceleration for real-time on-board inference, especially when processing high-resolution images. Third, the detection performance for heavily occluded and severely truncated small objects remains unsatisfactory, as such targets have extremely sparse feature information and are difficult to recognize effectively. Future work will focus on optimizing feature enhancement schemes for these extreme cases.

We also acknowledge that a more detailed analysis by object size bucket (e.g., tiny, small, medium) and by occlusion/truncation level would further strengthen the evaluation of our method. The VisDrone dataset provides occlusion and truncation flags that enable such a fine-grained assessment. While this analysis is beyond the scope of the current manuscript, we plan to conduct it in future work to provide a more comprehensive characterization of YOLOS0's performance under varying object scales and occlusion conditions.

Based on the above discussion, the results of this study have clear theoretical value and engineering application prospects. Theoretically, the network design combining high-resolution branches and multi-scale fusion, as well as the collaborative optimization of C3k2SO and C2PSASO modules, provide new ideas and methods for improving lightweight object detectors in small-object scenarios, enriching the technical system of UAV-based small-object detection. For engineering applications, YOLOS0 achieves a favorable trade-off between lightweight design and high precision, making it suitable for UAV remote sensing monitoring, security patrol, smart agriculture, and other practical scenarios. For instance, it can accurately detect small pests and diseases in farmland to support precision agriculture and effectively identify distant pedestrians and vehicles in security tasks to improve patrol efficiency and safety. Future research will address the limitations of this work by further refining the model structure, expanding validation datasets, enhancing robustness in extreme scenarios, and improving inference efficiency, so as to promote the practical deployment of UAV-based small-object-detection technology.

6. Conclusions

Aiming at the core problems in UAV-based ground small object detection, such as feature loss of small objects, poor scale adaptability, and strong background interference, this paper takes YOLOv11n as the basic framework and systematically optimizes it from three dimensions: network structure, core modules, feature enhancement. This paper proposes a small object-enhanced detection algorithm (YOLOS0) suitable for UAV-based detection scenarios.

The optimized network structure effectively mitigates the phenomenon of feature loss in small target objects. By adding a P2 high-resolution feature branch with a stride of 4, a four-scale detection system of “P2-P3-P4-P5” is constructed, reducing the minimum detection stride from 8 to 4, which significantly improves the ability to capture details of tiny objects. Meanwhile, a bidirectional feature fusion strategy of “top-down + bottom-up” is adopted to realize the deep interaction of multi-scale features, enhancing the network’s adaptability to scale variations and providing sufficient feature support for small object detection. Experiments show that structural optimization alone improves mAP50 by 2.8 percentage points as compared with the original baseline model.

The collaborative optimization of the two core modules, C3k2SO and C2PSASO, significantly improves the refinement and effectiveness of feature extraction. The C3k2SO module reduces small object feature loss and enhances the ability to capture local textures of small objects by adjusting the channel compression ratio (0.25 for shallow modules and 0.75 for deep modules), optimizing convolution kernel configuration (combination of 1×3 and 3×1 convolutions), and improving the attention mechanism (attention heads increased from 4 to 8, channel compression ratio reduced from 0.5 to 0.25, with an additional 1×1 convolutional layer). The C2PSASO module avoids the gradient disappearance of small object features and strengthens feature focusing ability by reducing the channel compression ratio (from 0.5 to 0.25), increasing the number of attention heads (from 4 to 8), and adding residual connections with a 1×1 convolutional branch. Ablation experiments verify that replacing with the C2PSASO module alone improves mAP50 by 0.7 percentage points, and introducing the C3k2SO module alone improves mAP50 by 1.4 percentage points. Their synergistic effect further boosts the detection capability of the optimized model.

Experimental results demonstrate that the YOLOS0 model has 3.56M parameters, still within the range of lightweight models. Its recall and mAP50 reach 36.8% and 37.3%, respectively, which corresponds to 4.8 percentage points and 3.7 percentage points improvements compared with the baseline YOLOv11n (32.0% recall and 33.6% mAP50), and signifi-

cantly outperforming mainstream lightweight models such as YOLOv8n, YOLOv9t, and YOLOv10n. In terms of real-time inference performance on aerial detection tasks, YOLOS achieves an FPS of 20.53 on the DOTA-v1 dataset. Although this value is slightly lower than that of the lightweight baseline models, including YOLOv8n (27.47 FPS), YOLOv9t (27.6 FPS), YOLOv10n (35.8 FPS), and YOLOv11n (26.53 FPS), the moderate reduction in inference speed is entirely acceptable for practical UAV deployment. The substantial improvement in detection accuracy effectively compensates for the minor loss of real-time performance, and the achieved frame rate can fully meet the basic real-time operation requirements of conventional UAV remote sensing and aerial monitoring tasks, realizing a reasonable balance between high-precision detection and practical deployment efficiency. The medium-to-large version YOLOS-S reduces parameters by 53.6% compared with Rtdetr-L (14.85M vs. 32.0M), while all performance metrics are significantly improved (mAP50 of 45.3% vs. 37.8%), verifying the superiority of the model at different scales.

Author Contributions: B.L.: Methodology, Software, Writing—original draft, H.Y.: Funding acquisition, Project administration (Corresponding Author), R.X.: Software, Formal analysis, H.L.: Validation, Formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study are publicly available. The VisDrone2019-DET dataset is available at <https://github.com/VisDrone/VisDrone-Dataset> (accessed on 20 April 2026). The DOTAv1 dataset is available at <https://captain-whu.github.io/DOTA/index.html> (accessed on 20 April 2026).

Acknowledgments: The authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions that greatly improved the quality of this manuscript. During the preparation of this manuscript, the authors used ChatGPT-4o (OpenAI) for English translation and grammatical proofreading. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

References

1. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
2. Feng, Q.; Yang, J.; Zhu, D.; Liu, J.; Guo, H.; Bayartungalag, B.; Li, B. Integrating multitemporal Sentinel-1/2 data for coastal land cover classification using a multibranch convolutional neural network: A case of the Yellow River Delta. *Remote Sens.* **2019**, *11*, 1006. [CrossRef]
3. Cheng, S.; Zhu, Y.; Wu, S. Deep learning based efficient ship detection from drone-captured images for maritime surveillance. *Ocean Eng.* **2023**, *285*, 115440. [CrossRef]
4. Zhao, H.; Zhang, H.; Zhao, Y. Yolov7-sea: Object detection of maritime uav images based on improved yolov7. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 233–238.
5. Prasad, D.K.; Prasath, C.K.; Rajan, D.; Rachmawati, L.; Rajabally, E.; Quek, C. Object Detection in a Maritime Environment: Performance Evaluation of Background Subtraction Methods. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 1787–1802. [CrossRef]
6. Han, W.; Chen, J.; Wang, L.; Feng, R.; Li, F.; Wu, L.; Tian, T.; Yan, J. Methods for small, weak object detection in optical high-resolution remote sensing images: A survey of advances and challenges. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 8–34. [CrossRef]
7. Muzammul, M.; Li, X. Comprehensive Review of Deep Learning-Based Tiny Object Detection: Challenges, Strategies, and Future Directions. *Knowl. Inf. Syst.* **2025**, *67*, 3825–3913. [CrossRef]
8. Liu, J.; Zhang, J.; Ni, Y.; Chi, W.; Qi, Z. Small-object detection in remote sensing images with super-resolution perception. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 15721–15734. [CrossRef]
9. 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.

10. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011. [\[CrossRef\]](#)
11. Wang, C.Y.; Yeh, I.H.; Mark Liao, H.Y. Yolov9: Learning what you want to learn using programmable gradient information. In *Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer: Berlin, Germany, 2024; pp. 1–21.
12. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725.
13. He, L.H.; Zhou, Y.Z.; Liu, L.; Cao, W.; Ma, J.H. Research on object detection and recognition in remote sensing images based on YOLOv11. *Sci. Rep.* **2025**, *15*, 14032. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Xiao, R.; Wang, H.; Wang, L.; Yuan, H. C3Ghost and C3k2: Performance study of feature extraction module for small target detection in YOLOv11 remote sensing images. In *Proceedings of the Second International Conference on Big Data, Computational Intelligence, and Applications (BDCIA 2024)*; Agaian, S.S., Ed.; SPIE: Bellingham, WA, USA, 2024; p. 139.
15. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. VisDrone-MOT2019: A benchmark for multi-object tracking in drone videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019*; pp. 1013–1022.
16. Ding, J.; Xue, N.; Xia, G.-S.; Bai, X.; Yang, W.; Yang, M.Y.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; et al. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7352–7368. [\[CrossRef\]](#)
17. Schou, J.; Skriver, H.; Nielsen, A.A.; Conradsen, K. CFAR edge detector for polarimetric SAR images. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 20–32. [\[CrossRef\]](#)
18. Zhu, C.; Zhou, H.; Wang, R.; Guo, J. A Novel Hierarchical Method of Ship Detection from Spaceborne Optical Image Based on Shape and Texture Features. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 3446–3456. [\[CrossRef\]](#)
19. Tello, M.; Lopez-Martinez, C.; Mallorqui, J.J. A Novel Algorithm for Ship Detection in SAR Imagery Based on the Wavelet Transform. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 201–205. [\[CrossRef\]](#)
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [\[PubMed\]](#)
21. Xiao, Q.; Cheng, Y.; Xiao, M.; Zhang, J.; Shi, H.; Niu, L.; Ge, C.; Lang, H. Improved region convolutional neural network for ship detection in multiresolution synthetic aperture radar images. *Concurr. Comput. Pract. Exp.* **2020**, *25*, e5820.
22. Ke, X.; Zhang, X.; Zhang, T.; Shi, J.; Wei, S. SAR ship detection based on an improved faster R-CNN using deformable convolution. In *Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021*; pp. 3565–3568.
23. Jian, L.; Pu, Z.; Zhu, L.; Yao, T.; Liang, X. SS R-CNN: Self-Supervised Learning Improving Mask R-CNN for Ship Detection in Remote Sensing Images. *Remote Sens.* **2022**, *14*, 4383. [\[CrossRef\]](#)
24. Ali, M.L.; Zhang, Z. The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. *Computers* **2024**, *13*, 336. [\[CrossRef\]](#)
25. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; Kwon, Y.; Michael, K.; Fang, J.; Yifu, Z.; Wong, C.; Montes, D.; et al. *ultralytics/yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*, Zenodo: Geneva, Switzerland, 2022. Available online: <https://ui.adsabs.harvard.edu/abs/2022zndo...7347926J/exportcitation> (accessed on 20 April 2026).
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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023*; pp. 7464–7475.
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
28. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *42*, 318–327. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Yu, C.; Shin, Y. SAR ship detection based on improved YOLOv5 and BiFPN. *ICT Express* **2023**, *10*, 28–33. [\[CrossRef\]](#)
30. Miao, T.; Zeng, H.; Wang, H.; Yang, W. Inshore ship detection in SAR images via an improved SSD model with wavelet decomposition. In *Proceedings of the 2021 7th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR), Bali, Indonesia, 1–3 November 2021*; pp. 1–5.
31. Yang, S.; An, W.; Li, S.; Wei, G.; Zou, B. An Improved FCOS Method for Ship Detection in SAR Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8910–8927. [\[CrossRef\]](#)
32. Liu, J.; Fan, X.; Huang, Z.; Wu, G.; Liu, R.; Zhong, W.; Luo, Z. Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022*; pp. 5802–5811.
33. Shen, J.; Chen, Y.; Liu, Y.; Zuo, X.; Fan, H.; Yang, W. ICAFusion: Iterative cross-attention guided feature fusion for multispectral object detection. *Pattern Recognit.* **2024**, *145*, 109913. [\[CrossRef\]](#)

34. 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*, 1–15. [\[CrossRef\]](#)
35. Guo, H.; Sun, C.; Zhang, J.; Zhang, W.; Zhang, N. Mmyfnet: Multi-modality yolo fusion network for object detection in remote sensing images. *Remote Sens.* **2024**, *16*, 4451. [\[CrossRef\]](#)
36. Fei, X.; Guo, M.; Li, Y.; Yu, R.; Sun, L. ACDF-YOLO: Attentive and cross-differential fusion network for multimodal remote sensing object detection. *Remote Sens.* **2024**, *16*, 3532. [\[CrossRef\]](#)
37. Wang, J.; Su, N.; Zhao, C.; Yan, Y.; Feng, S. Multi-modal object detection method based on dual-branch asymmetric attention backbone and feature fusion pyramid network. *Remote Sens.* **2024**, *16*, 3904. [\[CrossRef\]](#)
38. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
39. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
40. Zhao, L.; Ning, F.; Xi, Y.; Liang, G.; He, Z.; Zhang, Y. MSFA-YOLO: A Multi-Scale SAR Ship Detection Algorithm Based on Fused Attention. *IEEE Access* **2024**, *12*, 24554–24568. [\[CrossRef\]](#)
41. Zhu, L.; Chen, J.; Chen, J.; Yang, H. DGSP-YOLO: A Novel High-Precision Synthetic Aperture Radar (SAR) Ship Detection Model. *IEEE Access* **2024**, *12*, 167919–167933. [\[CrossRef\]](#)
42. Li, Z.; Ma, H.; Guo, Z. MAEE-Net: SAR ship target detection network based on multi-input attention and edge feature enhancement. *Digit. Signal Process.* **2024**, *156*, 104810.
43. Luo, Y.; Li, M.; Wen, G.; Tan, Y.; Shi, C. SHIP-YOLO: A Lightweight Synthetic Aperture Radar Ship Detection Model Based on YOLOv8n Algorithm. *IEEE Access* **2024**, *12*, 37030–37041. [\[CrossRef\]](#)
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
45. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R.; et al. ultralytics/yolov5. v3.0. Available online: <https://github.com/ultralytics/yolov5/releases/tag/v3.0> (accessed on 20 February 2025).
46. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
47. Tian, Y.; Ye, Q.; Doermann, D. Yolov12: Attention-centric real-time object detectors. *arXiv* **2025**, arXiv:2502.12524.
48. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
49. 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 (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
50. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the Integration of Self-Attention and Convolution. *arXiv* **2021**, arXiv:2111.14556.
51. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *arXiv* **2019**, arXiv:1902.09630.
52. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7 February 2020; Association for the Advancement of Artificial Intelligence (AAAI): Palo Alto, CA, USA, 2020; Volume 34, pp. 12993–13000.
53. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. *IEEE Trans. Cybern.* **2021**, *52*, 8574–8586. [\[CrossRef\]](#)
54. Gevorgyan, Z. Siou Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**, arXiv:2205.12740.
55. Zhang, H.; Xu, C.; Zhang, S. Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box. *arXiv* **2023**, arXiv:2311.02877.
56. Ultralytics. Ultralytics YOLO Docs. Available online: <https://docs.ultralytics.com/models/> (accessed on 20 December 2025).
57. Zhang, W.; Liu, Z.; Zhou, S.; Qi, W.; Wu, X.; Zhang, T.; Han, L. Ls-Yolo: A Novel Model for Detecting Multiscale Landslides with Remote Sensing Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 4952–4965. [\[CrossRef\]](#)
58. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974.

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.