

Article

YOLO-Ssboat: Super-Small Ship Detection Network for Large-Scale Aerial and Remote Sensing Scenes

Yiliang Zeng ^{1,2}, Xiuhong Wang ^{1,2}, Jinlin Zou ^{3,4,*} and Hongtao Wu ⁵

¹ Beijing Engineering Research Center of Industrial Spectrum Imaging, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China; ylzeng@ustb.edu.cn (Y.Z.); m202321675@xs.ustb.edu.cn (X.W.)

² Shunde Innovation School, University of Science and Technology Beijing, Foshan 528399, China

³ Equipment Management and UAV Engineering School, Air Force Engineering University, Xi'an 710043, China

⁴ National Key Laboratory of Unmanned Aerial Vehicle Technology, Xi'an 710051, China

⁵ Shanxi Intelligent Transportation Institute Co., Ltd., Taiyuan 030032, China; wht_ustb_doc2014@163.com

* Correspondence: kingkingyufei@163.com

Abstract: Enhancing the detection capabilities of marine vessels is crucial for maritime security and intelligence acquisition. However, accurately identifying small ships in complex oceanic environments remains a significant challenge, as these targets are frequently obscured by ocean waves and other disturbances, compromising recognition accuracy and stability. To address this issue, we propose YOLO-ssboat, a novel small-target ship recognition algorithm based on the YOLOv8 framework. YOLO-ssboat integrates the C2f_DCNv3 module to extract fine-grained features of small vessels while mitigating background interference and preserving critical target details. Additionally, it employs a high-resolution feature layer and incorporates a Multi-Scale Weighted Pyramid Network (MSWPN) to enhance feature diversity. The algorithm further leverages an improved multi-attention detection head, Dyhead_v3, to refine the representation of small-target features. To tackle the challenge of wake waves from moving ships obscuring small targets, we introduce a gradient flow mechanism that improves detection efficiency under dynamic conditions. The Tail Wave Detection Method synergistically integrates gradient computation with target detection techniques. Furthermore, adversarial training enhances the network's robustness and ensures greater stability. Experimental evaluations on the Ship_detection and Vessel datasets demonstrate that YOLO-ssboat outperforms state-of-the-art detection algorithms in both accuracy and stability. Notably, the gradient flow mechanism enriches target feature extraction for moving vessels, thereby improving detection accuracy in wake-disturbed scenarios, while adversarial training further fortifies model resilience. These advancements offer significant implications for the long-range monitoring and detection of maritime vessels, contributing to enhanced situational awareness in expansive oceanic environments.

Keywords: small target detection; deformable convolution; feature fusion; adversarial training



Academic Editor: Pedro Melo-Pinto

Received: 8 April 2025

Revised: 18 May 2025

Accepted: 1 June 2025

Published: 4 June 2025

Citation: Zeng, Y.; Wang, X.; Zou, J.; Wu, H. YOLO-Ssboat: Super-Small Ship Detection Network for Large-Scale Aerial and Remote Sensing Scenes. *Remote Sens.* **2025**, *17*, 1948. <https://doi.org/10.3390/rs17111948>

Copyright: © 2025 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) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Maritime vessel target detection plays a critical role in marine navigation and maritime traffic management. With advancements in artificial intelligence [1] and unmanned aerial vehicle (UAV) technologies [2], UAVs equipped with multispectral sensors can efficiently capture oceanic scenes for vessel monitoring. Target detection in UAV remote sensing images significantly contributes to intelligent decision-making [3,4] and has been widely applied across multiple domains [5–7].

Traditional vessel detection methods primarily employed sliding window techniques [8] combined with handcrafted features [9–11] or simplified target representations [12]. While these approaches demonstrated certain effectiveness, their performance remained limited in complex environments. The rapid development of convolutional neural networks (CNNs) has revolutionized target detection methodologies, which can be categorized into two-stage and single-stage paradigms. In two-stage detectors, Girshick et al. [13,14] proposed R-CNN and Fast R-CNN frameworks that first generate region proposals before extracting features through CNNs, overcoming traditional limitations and enabling target localization via fully connected layers. Ren et al. [15] further enhanced training efficiency by introducing the Region Proposal Network (RPN) in Faster R-CNN, replacing selective search. Single-stage detectors achieve simultaneous classification and bounding box regression through single-step prediction. Redmon et al. [16] pioneered the YOLO series, renowned for its real-time performance and accuracy. Subsequent iterations, including YOLOv5 by the Ultralytics team [17], YOLOv7 by Wang et al. [18], and recent versions YOLOv9 [19], YOLOv10 [20], and YOLO11 [21] continue to advance detection speed and precision. Beyond the YOLO series, Liu et al. [22] developed SSD using a VGG-16 backbone with multi-scale prediction via progressively decreasing convolutional layers. Zhao et al. [23] proposed RT-DETR, optimizing target query selection and reducing computational redundancy for enhanced speed-accuracy balance.

Recent research has specifically addressed maritime detection challenges through algorithm adaptations. For optical remote sensing imagery, Yi et al. [24] improved Faster-RCNN accuracy at the expense of detection speed. Xu et al. [25] tackled feature misalignment and small-target loss in SAR images by integrating path-aggregated feature pyramids and CIoU loss. Li [26] developed MMAF-NET, leveraging multi-scale encoding and attention fusion for improved detection under complex marine conditions. Wen et al. [27] optimized anchor box dimensions and introduced hybrid loss functions with non-maximum suppression to address localization uncertainties. Kang et al. [28] enhanced small-target SAR detection through intermediate layer fusion for region proposal generation. Li et al. [29] employed four-scale feature fusion and attention mechanisms to handle vessel scale variation and channel information imbalance. Yu et al. [30] augmented YOLOv5 with bidirectional feature pyramids for superior small-target detection. Chen et al. [31] proposed SAS-FPN, combining attention mechanisms and spatial pyramids to preserve critical vessel features across scales. For rotational invariance in SAR detection, Li et al. [32] introduced a multidimensional domain network utilizing polar Fourier transforms to extract rotation-invariant features. Chen et al. [33] addressed dense vessel detection through attention mechanisms, Soft-NMS, and GIoU loss to mitigate missed detections in overlapping scenarios.

Based on the above, deep learning-based ship detection significantly improves the detection efficiency of ships on the sea surface. However, existing studies mostly focus on stable and simple backgrounds, while the detection of ships affected by wake interference has not received sufficient attention. The utilization of UAV imagery or video for intelligent ship detection has emerged as an increasingly prevalent trend. However, as illustrated in Figure 1, the fixed flight altitude and variable shooting angles of UAVs result in alterations to the shape and size of targets in aerial imagery. UAV-based small target ship detection still faces the following challenges:

1. The long distance, wide angle, and small proportion of the target ships in large scenes captured by UAVs make it difficult for deep networks to extract features, thus increasing the difficulty of target detection.
2. Aerial images of small target ships exhibit uncertain angles, diverse shapes, and complex marine environmental backgrounds. The varying shapes and backgrounds of ships affect the stability of detection algorithms.

3. The wake generated by small target ships in motion often occupies an area larger than the ship itself. The wake typically covers the target, thus increasing the difficulty of detection. To address these issues, this paper proposes an innovative method for small target ship detection in marine aerial imagery, YOLO-ssboat.

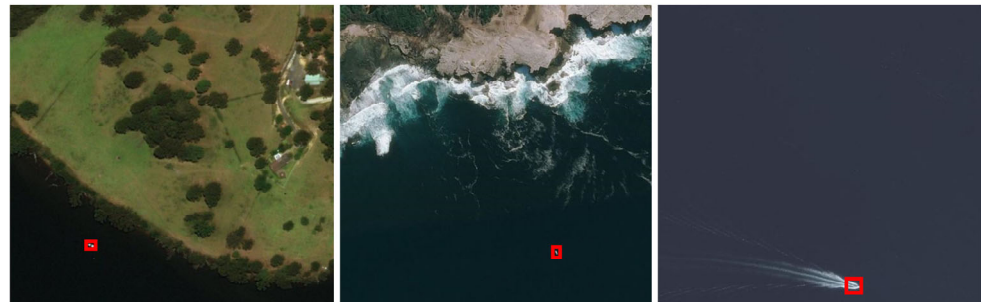


Figure 1. Aerial Perspective for Detecting Small Maritime Vessels Using UAVs. The red box indicates the ground truth.

The main contributions of this paper are as follows:

1. An innovative feature extraction module, C2f_DCNv3, is proposed. C2f_DCNv3 introduces offsets to shift the convolution operations towards the target, reducing the focus on the background and enhancing the attention to ship targets. The Neck feature fusion structure is redesigned by incorporating small-scale feature layers, utilizing MSWPN to fuse all features within the same scale, and applying weighted fusion across different scales. Furthermore, the attention-based tri-level detection head, Dyhead_v3, is improved to dynamically adjust the output of the detection head according to the target size while upgrading the three-level detection head to a four-level detection head.
2. To address the issue of ship wake covering the ship target, we propose a ship wake gradient detection method. We have created a custom wake detection dataset, transforming the problem of ship detection into that of detecting the wake. Using gradient flow methods, we locate small target ships, and the wake gradient detection method demonstrates excellent performance in detecting ships affected by wakes.
3. This paper introduces adversarial training into the domain of ship target detection. To address the problem of adversarial samples compromising deep network performance, we explore the impact of disturbances with varying intensities on the vulnerability of deep neural networks. Additionally, we investigate the role of different proportions of adversarial samples in enhancing network robustness.

The structure of this paper is as follows: Section 2 reviews the development of deformable convolutions, feature fusion results, and attention mechanisms in the field of object detection. Section 3 provides an overview of UAV imagery and remote sensing ship small target detection models, offering a detailed description of the model structure and relevant module principles. Section 4 describes the experimental environment and parameter configurations, presenting ablation experiments, comparative evaluations with other models, and visualization experiments aimed at assessing the effectiveness of the proposed method. Section 5 investigates in-depth how the proposed model addresses the challenges in small-target ship detection. Finally, Section 6 summarizes the entire paper and discusses future research directions.

2. Related Work

Object detection technology has been the subject of extensive investigation, culminating in remarkable progress in accuracy and computational efficiency. This section

reviews the evolution of cutting-edge methodologies, with an emphasis on three pivotal components: deformable convolution for feature extraction, feature fusion, and attention mechanisms.

2.1. Deformable Convolution

The primary function of convolutional neural networks (CNNs) is to extract features from images. In 2D convolution, feature extraction typically involves two main steps: first, sampling the input image x over a predefined grid $\mathcal{R}$; second, applying weighted summation to the sampled values using a set of learnable weights w . The design of the sampling grid $\mathcal{R}$ largely determines the receptive field and dilation rate. For example, a standard 3×3 convolution kernel with a dilation rate of 1 is defined as follows:

$$\mathcal{R} = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\} \quad (1)$$

This grid spans a local neighborhood of 3×3 pixels, enabling the kernel to capture spatial correlations within this region. The dilation rate explicitly controls the spacing between the sampled points, where a rate of 1 corresponds to contiguous sampling (no gaps), while higher rates induce sparser sampling patterns to enlarge the receptive field without increasing the parameter count.

For each position p_0 on the output feature map y , we have:

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n) \quad (2)$$

In convolutional neural networks (CNNs), the fixed sampling positions p_n within the grid $\mathcal{R}$ inherently limit the ability to model geometric transformations, a common drawback in CNN-based object detection frameworks. To address this limitation, Dai et al. [34] proposed deformable convolution, which enhances geometric transformation modeling by introducing learnable offsets to the convolution operation. In deformable convolution, the regular grid $\mathcal{R}$ is adaptively expanded through learnable offsets $\{\Delta p_n \mid n = 1, \dots, N\}$ denotes the cardinality of the grid. This modification transforms the standard convolution operation Equation (2) into its deformable counterpart Equation (3):

$$y(p_0) = \sum_{p_n \in \mathcal{R}} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \quad (3)$$

Compared with conventional convolutions, deformable convolutions introduce learnable offsets that adapt to the geometric variations of objects, enabling more accurate feature extraction aligned with object shapes. However, these offsets may extend beyond the target region, potentially degrading the quality of extracted features. To address this limitation, Zhu et al. [35] proposed DCNv2 based on DCNv1, which enhances the modeling capacity of deformable convolutions to better focus on relevant object regions. The improved deformable convolution can be formally expressed as Equation (4):

$$y(p) = \sum_{k=1}^K w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k \quad (4)$$

Here, Δp_k and Δm_k denote the learnable offset and modulation scalar at the k -th location, respectively. DCNv2 is typically employed as an extension of standard convolution, where it benefits from loading pre-trained weights from conventional convolutions and fine-tuning them for improved performance. However, when training large-scale vision foundation models from scratch, DCNv2 still incurs substantial computational costs. To address this issue, Wang et al. [36] proposed DCNv3, which reduces computational overhead

by sharing weights across convolutional neurons. Furthermore, DCNv3 introduces spatial clustering, dividing the training process into G groups, with each group assigned its own sampling offset p_{gk} and modulation scale m_{gk} . The DCNv3 operation can be formulated as Equation (5):

$$y(p_0) = \sum_{g=1}^G \sum_{k=1}^K w_g m_{gk} x_g(p_0 + p_k + \Delta p_{gk}) \quad (5)$$

where G represents the total number of aggregation groups. For the g -th group, $w_g \in \mathbb{R}^{C \times C'}$ represents the position-independent projection weights of that group, where $C' = C/G$ is the group dimension. $m_{gk} \in \mathbb{R}$ represents the modulation scalar of the k -th sampling point in the g -th group, normalized across the K -dimension using the softmax function. $x_g \in \mathbb{R}^{C' \times H \times W}$ represents the sliced input feature map. Δp_{gk} is the offset corresponding to the grid sampling position p_k in the g -th group.

2.2. Feature Fusion

Traditional single-scale feature extraction methods often fail to meet the requirements of object detection in complex scenarios. Multi-scale feature fusion techniques have been introduced in object detection architectures to address this issue. Lin et al. [37] proposed a multi-scale feature fusion network structure known as FPN (Feature Pyramid Network). The core idea of FPN is to construct a feature pyramid through top-down and bottom-up feature propagation, allowing the network to obtain rich feature information at different scales. Liu et al. [38] introduced a path aggregation network (PANet), which enhances the entire feature hierarchy through bottom-up path augmentation, utilizing precise localization information from the lower layers. This reduces the information path between the lower and upper layers, connecting feature grids across all levels so that useful information from each feature level is directly propagated to the subsequent proposal subnetworks. Ghiasi et al. [39] proposed NAS-FPN, which leverages neural architecture search (NAS) in a scalable search space covering all cross-scale connections to discover a new feature pyramid structure. NAS-FPN is composed of top-down and bottom-up connections, enabling cross-range feature fusion. Tan et al. [40] introduced a novel feature fusion method called the repeated weighted bi-directional feature pyramid network (BiFPN).

2.3. Attention

In recent years, researchers have improved object detectors by focusing on scale perception, spatial perception, and task awareness. The authors of [41] improved this by introducing balanced sampling and a balanced feature pyramid. Wang et al. [42] proposed a pyramid convolution that extracts both scale and spatial features using an improved 3-D convolution. It is well known that convolutional neural networks (CNNs) are limited in learning spatial transformations present in images. Yu et al. [43] introduced dilated convolutions to aggregate contextual information from exponentially expanding receptive fields. Duan et al. [44] formulated object detection as a representation of keypoints to simplify the learning process. Qiu et al. [45] improved performance by detecting each object as a triplet rather than a pair of keypoints, thereby reducing mispredictions. Zou et al. [46] proposed extracting boundary features from the extreme points of each boundary to enhance point features, achieving state-of-the-art performance. Finally, Dai et al. [47] introduced a dynamic detection head that combines scale attention, spatial attention, and task attention, significantly improving object detection efficiency.

2.4. Adversarial Training

Adversarial examples refer to synthetic samples that introduce subtle perturbations to the original samples, leading to incorrect classification by the model. Adversarial attacks

involve the use of these adversarial examples to compromise the accuracy of classification models. Since the concept of adversarial examples was introduced by Szegedy et al. [48], an increasing number of researchers have proposed methods for generating adversarial examples. For instance, FGSM (Fast Gradient Sign Method) [49] generates adversarial examples by adding small perturbations in the direction of the largest gradient change in a neural network model, leading to misclassification. PGD (Projected Gradient Descent) [50], by leveraging gradient information, iteratively approaches the decision boundary of the model, making the adversarial examples closer to the boundary and thus more deceptive. Deepfool [51] computes the minimum distance between the decision boundary of the original sample and the adversarial sample to determine the perturbation size. Auto Attack [52] combines multiple state-of-the-art adversarial attack methods and automates the process, providing a powerful and reliable tool for evaluating the adversarial robustness of models.

3. Proposed Model

3.1. The Proposed Method

The proposed YOLO-ssboat framework is illustrated in Figure 2. Specifically, the C2f_DCNv3 module is introduced in the backbone to extract discriminative features of small-scale vessel targets. In the feature fusion stage, the MSWPN module is employed to integrate multi-scale features, thereby enriching the representation of small vessel targets. Finally, the DyHead_v3 detection head is used to enhance the feature representation of small vessels. Additionally, for small targets partially obscured by tail waves, YOLO-ssboat first detects the tail wave, and then the Tail Wave Detection Method is applied to accurately determine the position of the vessel target.

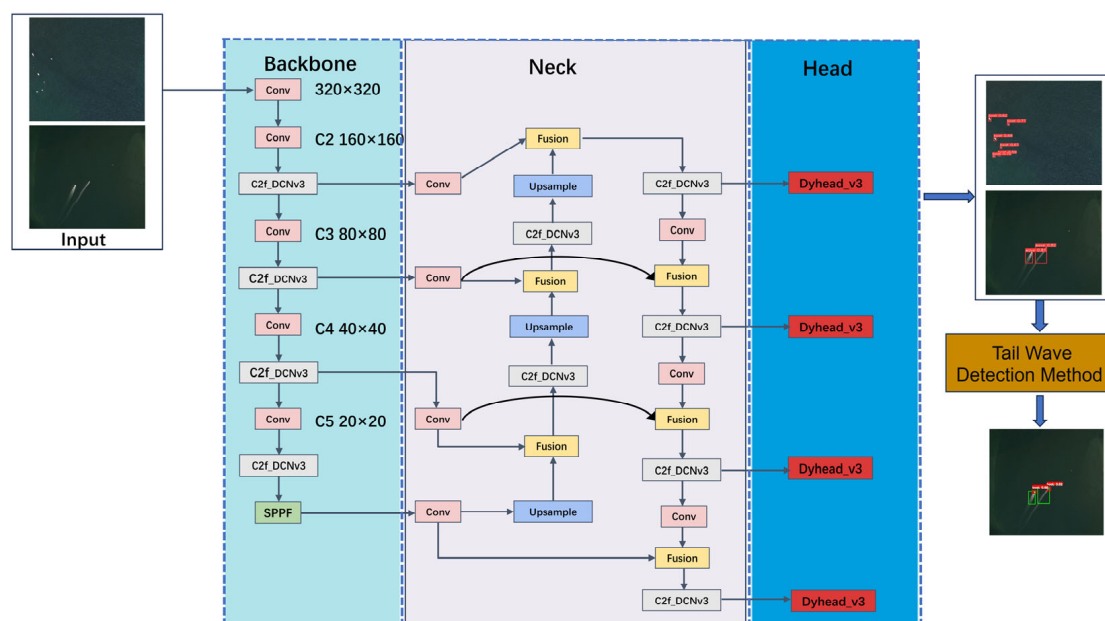


Figure 2. Network model of YOLO-ssboat algorithm. For conventional vessel detection, YOLO-ssboat can accurately detect small vessel targets. In cases where vessels are partially occluded by tail waves, the model first detects the tail wave and then applies the Tail Wave Detection Method to locate the vessel target. (The green boxes indicate the detected ship wakes).

3.2. Proposed C2f_DCNv3

The C2f_DCNv3 structure proposed in this paper is shown in Figure 3. C2f_DCNv3 first processes the input image using a 1×1 convolution (Conv1), doubling the number of channels in the output feature map to enhance the model's feature representation capability.

The input feature map is then split into two parts by the Split module. One part enters the Bottleneck module, while the other part is directly involved in subsequent concatenation. After splitting, the feature map is processed layer by layer through multiple Bottleneck modules, where deformable convolution (DCNv3) is used to extract deeper features. The Bottleneck uses shortcut connections to enhance gradient propagation and information flow. Finally, a convolution (Conv2) is used to compress the number of channels in the concatenated feature map to the desired output channels. One key issue in small target ship detection in UAV aerial images of the ocean is how to adapt to geometric variations in object scale, posture, and deformation, and how to distinguish ship targets from the marine background, such as waves.

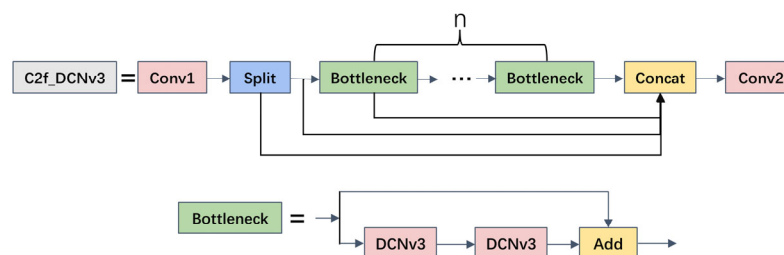


Figure 3. C2f_DCNv3 Architecture Diagram.

Figure 4a illustrates the rigid feature mapping of standard convolution. The blue vertical lines represent the fixed receptive fields of conventional convolutional kernels. The upper-layer red dots indicate the predefined anchor points of the standard convolution kernels, while the lower-layer red dots represent the expanded receptive field regions. The geometric centers of the receptive fields in standard convolution strictly align with the grid coordinates. Figure 4b shows a regular square grid (referred to as the base grid), where the uniformly distributed black dots denote the fixed sampling locations of standard convolution (e.g., a 3×3 grid). The red dots represent the deformed sampling positions obtained by applying learnable offsets to the original black dots. For instance, if the coordinate of a black dot is $(0, 0)$ and the offset is $\Delta p = (0.2, -0.3)$, then the corresponding sampling location for deformable convolution becomes $(0.2, -0.3)$. These offsets Δp are generated by the network through learning. Figure 4c illustrates the dynamic adaptability of deformable convolution. The upper-layer red dots represent the predefined anchor points of the deformable convolution kernel, while the lower-layer red dots reflect the receptive field locations that adaptively shift according to the target.

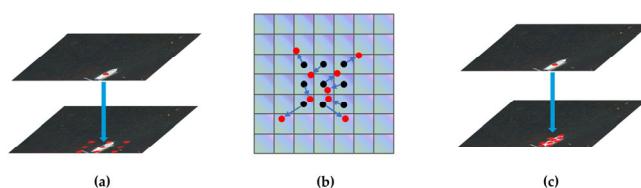


Figure 4. Comparison chart of Regular Convolution Feature Maps and Deformable Convolution. (a) Rigid feature mapping of conventional convolution. (b) Visual comparison of sampling strategies between standard convolution and deformable convolution. (c) Dynamic adaptation of deformable convolution. (The blue arrows map the sampling area of the convolutional kernel).

To more intuitively demonstrate the effectiveness of deformable convolution in enhancing the feature extraction of small ship targets, we visualized the feature maps of the 2nd, 4th, and 6th layers in the backbone of YOLO-ssboat during dense small ship detection, corresponding to layers C2, C3, and C4 in Figure 2. Figure 5a shows the feature extraction heatmaps using standard convolution, while Figure 5b presents the results obtained with C2f_DCNv3. In the shallow layers with higher feature resolution (i.e., the 2nd and 4th

layers), the standard convolution heatmaps reveal large blue areas in the background, such as ocean waves. In contrast, the use of C2f_DCNv3 significantly reduces the blue background regions and enhances the contrast between the targets and the background. In the deeper layers with lower resolution (e.g., the 6th layer), standard convolution tends to yield dispersed feature responses, whereas deformable convolution maintains concentrated yellow response regions due to its adaptive receptive field, indicating stronger capability in capturing long-range features of deformable targets. The more concentrated distribution of yellow areas in the deformable convolution heatmaps reflects improved target localization accuracy. Notably, in the complex scenes represented by the 6th layer, the target edges and local features still exhibit high-contrast responses. This characteristic enables the network to maintain robust feature extraction performance even when ship targets undergo deformation or occlusion.

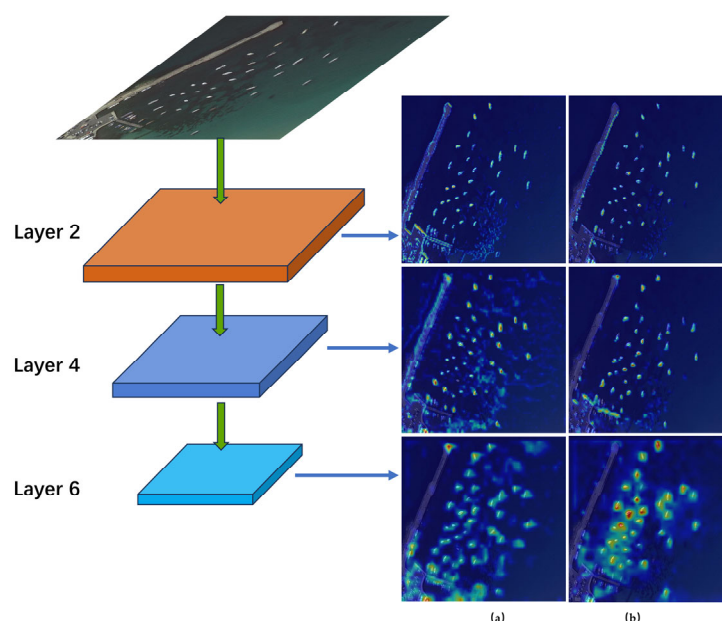


Figure 5. Heatmap visualization of different feature extraction modules: (a) Feature layer heatmap of standard convolution. (b) Feature extraction heatmap of deformable convolution. (The green arrows represent the transmission process of feature layers at different scales).

3.3. Proposed MSWPN

Small target ships in the ocean occupy very few pixels, and as they pass through multiple convolution layers, their features may be progressively lost. As illustrated in the heatmap in Figure 6a, the resolution decreases from 160 at layer C2 to 20 at layer C5. As the number of convolution layers increases, the features corresponding to small target ships are gradually diminished. In deeper layers, the activation of small target ships becomes less prominent, while background noise, such as ocean waves, is more likely to be activated. From the heatmap, it is evident that the C2 layer exhibits stronger activation for small target ships compared to other layers. To enhance the detection of small target ships, the C2 detection layer is integrated into the feature fusion network. The fusion of feature maps at different scales is crucial for small-target ship detection. The proposed Multi-Scale Weighted Fusion Network (MSWPN), as depicted in Figure 6, addresses this by incorporating the C2 feature layer and simultaneously connecting inputs and outputs at the same scale. For instance, P4(1), P4(2), and P4(3) are feature maps with a resolution of 40, and these are fused together to aggregate more features without significantly increasing computational costs. In the typical approach of fusing feature maps from different sources, these are usually resized to a uniform resolution before being added together. However, the

contribution of each feature map to the final output is not necessarily uniform. As shown in the heatmap in Figure 6a, lower-level feature maps clearly provide more significant contributions for small target detection. Hence, the conventional approach may not be optimal. To address this limitation, we assign a weight to each feature map, and the output is given by:

$$\text{Output} = \frac{w_1 P_1 + w_2 P_2 + w_3 P_3}{w_1 + w_2 + w_3 + \varepsilon} \quad (6)$$

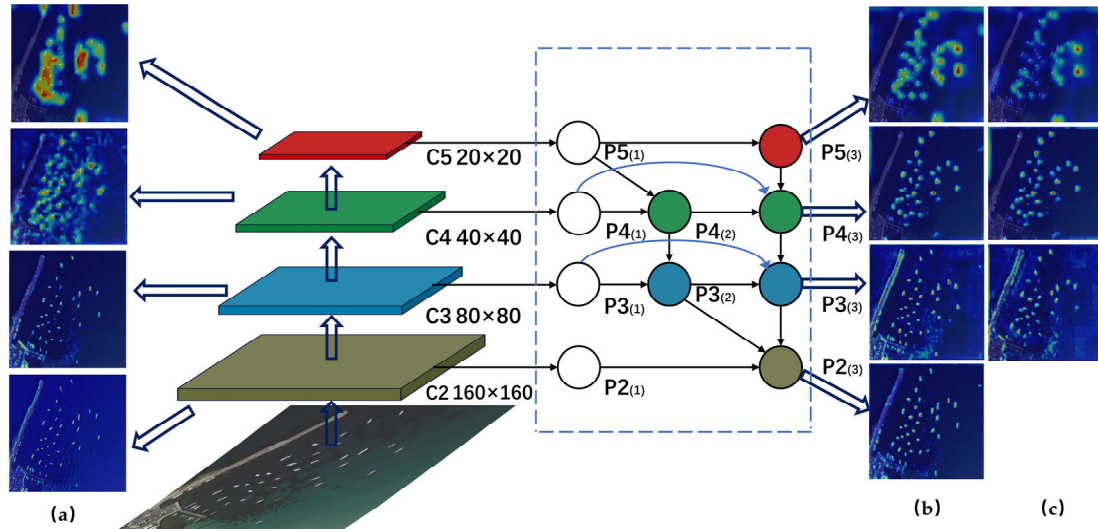


Figure 6. Multi-Scale Weighted Fusion Structure. (a) Visualizations of multi-scale feature maps extracted from the backbone network; (b) Feature maps after multi-scale fusion using the proposed MSWPN module; (c) Feature maps obtained from the YOLOv8 feature fusion process. (The different colors in the figure represent feature layers at different scales, and the arrows represent feature transfers).

In the equation, P_i represents the feature map, and w_i denotes the weight assigned to each feature map, with $\varepsilon = 0.0001$ used to maintain numerical stability. This formulation ensures that each normalized weight lies between 0 and 1. For instance, in Figure 6, node $P_4(2)$ has two inputs, $P_5(1)$ and $P_4(1)$, to which weights w_1 and w_2 are assigned, respectively. The features of this node can be expressed by the following equation:

$$P_4(2) = \text{Conv} \left(\frac{w_1 \cdot P_4(1) + w_2 \cdot \text{Resize}(P_5(1))}{w_1 + w_2 + \varepsilon} \right) \quad (7)$$

The heatmap visualization of features fused at different scales using MSWPN is shown in Figure 6b. After the multi-scale weighted feature fusion, the activation of the small targets is significantly enhanced, while the activation in the background, particularly the ocean waves, is notably reduced.

In MSWPN, the weights w_i are learnable parameters that are continuously adjusted during the training process, and their values may vary across different datasets. These weights are initialized with small random values and constrained to be non-negative through the ReLU activation function, ensuring $w_i \geq 0$ to prevent negative weights from disrupting the rationality of feature fusion. During training, the weights are optimized via gradient descent to adaptively balance the contributions of features at different resolutions. A constant ε is fixed at 0.0001 to avoid division by zero when all weights approach zero and to stabilize gradient computation, thereby preventing numerical explosion during back-propagation. To ensure reproducibility and consistency with the original implementation, it is essential to pay attention to weight initialization using small random values (e.g., from

$N(0, 0.01)$), enforce non-negativity, and strictly use $\varepsilon = 0.0001$ in the denominator without omission or arbitrary modification.

3.4. Proposed Dyhead_DCNv3

Traditional detection heads, when processing objects of varying sizes, are typically designed for specific target dimensions at each scale. Predictions for each position are generated independently. Additionally, the original YOLOv8 detection head lacks dynamic learning capabilities, thereby imposing significant limitations on the detection of multi-scale objects, particularly small targets. The variation in object scale is intricately tied to features at different hierarchical levels, and enhancing features at these levels, denoted as $\mathcal{F}$, significantly improves scale perception, particularly for small vessel detection. The geometric transformations of vessels, with varying shapes, are intricately associated with spatial positional features across different levels. Enhancing the representational learning of $\mathcal{F}$ at diverse spatial locations contributes significantly to improved spatial awareness in object detection. Moreover, distinct object representations and associated tasks are frequently linked to features across various channels. Enhancing the representational learning of $\mathcal{F}$ across different channels significantly benefits task-specific object detection.

The improved Dyhead_v3 proposed in this paper is shown in Figure 7. It unifies scale-aware attention $\pi_L(\cdot)$, spatial-aware attention $\pi_S(\cdot)$, and task-aware attention $\pi_C(\cdot)$, thereby better integrating contextual information. By utilizing DCNv3, spatial-aware attention enhances the model's ability to perceive the position of small target vessels.

$$W(\mathcal{F}) = \pi(\mathcal{F}) \cdot \mathcal{F} \quad (8)$$

where $\pi(\cdot)$ represents the attention function, and L, S, C correspond to the scale, spatial, and task dimensions, respectively. In Dyhead_v3, the attention function is decomposed into three consecutive attention mechanisms, each focusing on a single dimension.

$$W(\mathcal{F}) = \pi_C(\pi_S(\pi_L(\mathcal{F}) \cdot \mathcal{F}) \cdot \mathcal{F}) \cdot \mathcal{F} \quad (9)$$

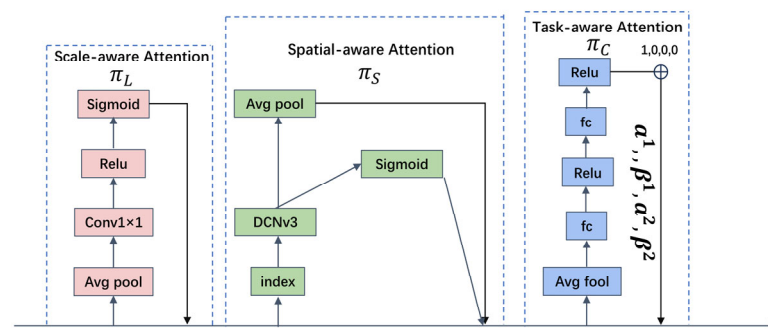


Figure 7. Dyhead_v3 Structure Diagram. The Dyhead_v3 architecture consists of three key attention modules: Scale-aware Attention ($\pi_L(\cdot)$), Spatial-aware Attention ($\pi_S(\cdot)$), and Task-aware Attention ($\pi_C(\cdot)$). (The arrows represent the transfer process of the eigenvectors).

In the equation, $\pi_L(\cdot)$, $\pi_S(\cdot)$, $\pi_C(\cdot)$ are three distinct attention functions, each operating on the L, S, C dimensions, respectively.

1. Scale Attention

First, scale-aware attention is introduced to dynamically fuse features based on the semantic importance of different scales.

$$\pi_L(\mathcal{F}) \cdot \mathcal{F} = \sigma \left(f \left(\frac{1}{SC} \sum_{S,C} \mathcal{F} \right) \right) \cdot \mathcal{F} \quad (10)$$

In the equation, $f(\cdot)$ is a linear function approximated by a 1×1 convolutional layer, and $\sigma(x) = \max(0, \min(1, (x + 1)/2))$ is a sigmoid function.

2. Spatial Attention

The spatial module is decomposed into two steps. First, deformable convolution DCNv3 is employed to induce sparsity in the attention-learning process. Then, features are aggregated across layers at the same spatial location.

$$\pi_S(\mathcal{F}) \cdot \mathcal{F} = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K w_{l,k} \cdot \mathcal{F}(l; p_k + \Delta p_k; c) \cdot \Delta m_k \quad (11)$$

Here, k denotes the number of sparse sampling positions, $p_k + \Delta p_k$ represents the position shifted by the self-learned spatial offset Δp_k to focus on a discriminative region, and Δm_k is the self-learned importance scalar for position p_k . Both are learned from the median-level input features of $\mathcal{F}$.

3. Task Attention

To achieve joint learning and generalize diverse object representations, task-aware attention is deployed in the final stage. It dynamically toggles feature channels ON and OFF to support different tasks:

$$\pi_C(\mathcal{F}) \cdot \mathcal{F} = \max\left(\alpha^1(\mathcal{F}) \cdot \mathcal{F}_c + \beta^1(\mathcal{F}), \alpha^2(\mathcal{F}) \cdot \mathcal{F}_c + \beta^2(\mathcal{F})\right) \quad (12)$$

The channels of the $\mathcal{F}_c$ feature layer, $[\alpha^1, \alpha^2, \beta^1, \beta^2]^T = \theta(\cdot)$, serve as hyperfunctions for learning and controlling activation thresholds. First, global average pooling is performed over the $L \times S$ dimensions to reduce dimensionality. This is followed by two fully connected layers and a normalization layer. Finally, a shifted sigmoid function is applied to normalize the output to the range $[-1, 1]$.

3.5. Tail Wave Detection Method

Beyond challenges such as the diminutive size of ship targets and the inherent complexity of the marine environment, the wake produced by a ship's motion further complicates detection efforts. Typically, the wake generated by small vessels at sea substantially exceeds the physical dimensions of the vessel itself. When a vessel is obscured by its wake, target detection networks—perturbed by wave interference—struggle to extract salient ship features, culminating in detection failure. To tackle this challenge, we introduce a novel gradient flow algorithm, as illustrated in Figure 8. The proposed algorithm initially leverages a deep neural network to detect the ship's wake, subsequently transforming the detected wake into a grayscale image that retains solely intensity information. Subsequently, the Sobel operator is employed to compute image gradients along the x and y axes, as delineated in Equations (13) and (14), where $I(x + 1, y)$ denotes the pixel intensity at coordinate $(x + 1, y)$. The computed gradient quantifies both the direction and magnitude of intensity variations within the image. Following the gradient computation along both axes, the gradient magnitudes at each pixel are derived through summation, as articulated in Equation (11). The gradient flow elucidates the trajectory of the most pronounced intensity transitions in the image, with the wake distinctly accentuated at the vessel's location. By harnessing this gradient flow, the proposed approach enables precise localization of the vessel, even under occlusion by its wake, thereby addressing the shortcomings of conventional target detection techniques in maritime scenarios.

$$G_x \approx I(x + 1, y) - I(x - 1, y) \quad (13)$$

$$G_y \approx I(x, y + 1) - I(x, y - 1) \quad (14)$$

$$M = \sqrt{G_x^2 + G_y^2} \quad (15)$$

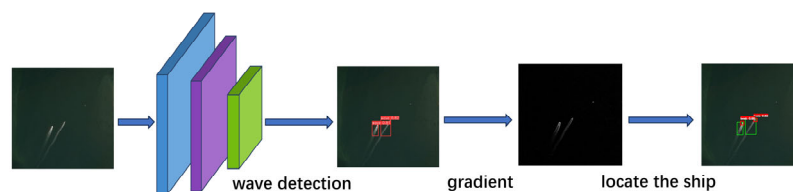


Figure 8. Schematic diagram of Tail Wave Detection Method. Using the detected wake from object detection, the ship target is localized through gradient-based computation. (The green box shows the detected wake wave).

3.6. Adversarial Training

Adversarial Training (AT) is an efficient method for defending against adversarial examples in deep learning, playing a crucial role in ensuring the robustness and security of models. Common adversarial methods can be broadly categorized into two types. The first is standard adversarial training (SAT), which generates adversarial examples at each step by adding perturbations using methods like PGD:

$$x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x L(f(x), y)) \quad (16)$$

where x is the natural sample, y is the label, ϵ is the hyperparameter controlling the magnitude of the perturbation, and $\nabla_x L(f(x))$ represents the gradient of the loss function with respect to the input. Given the constraint of adversarial perturbation, the goal is to find the adversarial perturbation that maximizes the model's loss. The model parameters θ are then adjusted to minimize this loss, thus improving the model's robustness. The objective function for adversarial training is:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in \mathcal{S}} L(f(x + \delta; \theta), y) \right] \quad (17)$$

4. Experiment

4.1. Experiment Dataset

4.1.1. Ship_Detection Dataset

The ship_detection [53] dataset is a publicly available ship detection dataset released on the Roboflow platform on 14 August 2023. It contains 6917 training images and 1281 testing images, all with a resolution of 640×640 pixels. As shown in Table 1, small ships (with target pixel area $S < 32^2$) account for approximately 60.8% of the dataset, medium-sized ships for 30.5%, and large ships for about 8.7%. The size distribution is illustrated in Figure 9. The dataset primarily includes civilian vessels such as fishing boats, small yachts, and speedboats, with a smaller portion of images containing large cargo ships. It covers various weather conditions, including clear, cloudy, and light fog, with all images captured during the daytime. The sea state in most images ranges from calm to slight waves.

Table 1. The dataset target size statistics. S denotes the size of the ship targets in the dataset.

	Ship_Detection	Vessel
Image Size	640×640	416×416
$S < 32^2$	15,362 (60.8%)	14,674 (82%)
$32^2 < S < 96^2$	7715 (30.5%)	3137 (17.5%)
$S > 96^2$	2189 (8.7%)	75 (0.5%)

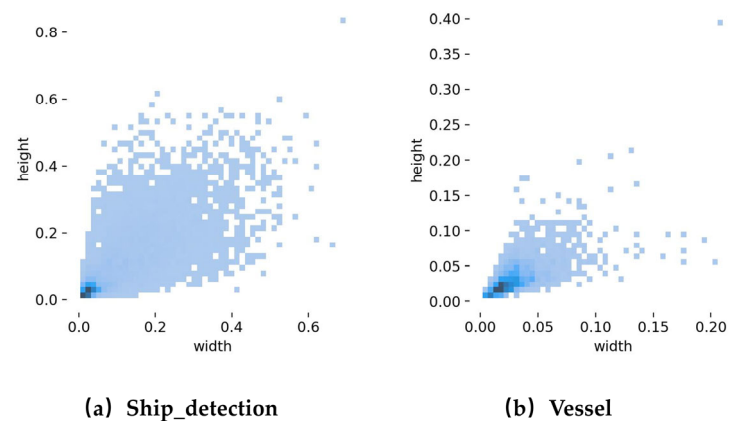


Figure 9. Dataset Size Distribution Chart. (The darker the blue, the more data there is).

4.1.2. Vessel Dataset

The Vessel dataset [54] is a publicly available ship detection dataset released on the Roboflow platform on 6 December 2021. It consists of 2882 training images and 826 testing images, primarily featuring small fishing boats and patrol vessels, with a small number of medium-sized cargo ships. Table 1, small vessels account for 82% of the dataset, medium-sized vessels for 17.5%, and large vessels for only 0.5%, indicating a strong dominance of small targets. The Vessel dataset includes scenes such as nearshore waters and port docking areas. Some images exhibit lighting variations and background interference (e.g., dock infrastructure), which present additional challenges for detection.

Both the ship_detection and Vessel datasets focus on UAV- or satellite-based remote sensing perspectives, with targets primarily consisting of small-sized vessels. These datasets are well-suited for low-altitude small ship target detection tasks.

In the wake gradient flow detection, images containing wake waves were selected from the ship_detection and Vessel datasets and annotated using Labelme tools. A total of 1484 wake images were obtained, forming the wake detection dataset.

4.2. Experimental Setup and Evaluation Metrics

4.2.1. Experimental Setup

This study was carried out on a Linux-based operating system. The software and hardware configurations utilized in this experiment are summarized in Table 2.

Table 2. Experimental Software and Hardware Configuration.

Computer Configurations		Version
GPU	NVIDIA Geforce RTX 4090 (NVIDIA, Santa Clara, CA, USA)	
Cuda		11.8
Python		3.8
Pytorch		2.0.0
Torchvision		2.0.1

Table 2 outlines the experimental parameter settings: batchsize ensures stable gradient computation and enhances training efficiency, while workers specifies the number of worker threads for data loading. Setting batchsize and workers to 16, based on the available computational resources, significantly accelerates data loading and preprocessing. epoch denotes the number of complete iterations over the training dataset. Insufficient epochs result in an undertrained model, whereas excessive epochs may lead to model overfitting. lr_ratio controls the step size for parameter updates. A high learning rate may cause oscillation and prevent convergence, while a low learning rate may slow down optimization

or halt it prematurely before convergence is achieved. In this study, epoch is set to 300, and lr_ratio is configured to 0.001. Parameters not specified in Table 3 are assigned their default values as defined by the algorithm. To ensure the reliability of the experiments, all model comparisons and ablation studies in this paper were conducted based on the hardware and software environment detailed in Table 2, as well as the parameter settings listed in Table 3.

Table 3. Experimental Parameter Settings.

Parameter	Setting
Batch size	16
epochs	300
workers	16
lr_ratio	0.001

4.2.2. Evaluation Metrics

To ensure a fair evaluation of the model's detection performance, Precision, Recall, and mAP were used as evaluation metrics. These metrics were derived from the following parameters: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP refers to the number of ships correctly detected, TN denotes the correctly identified background, FP represents the number of false detections, and FN indicates the number of False Negative:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (18)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (19)$$

$$\text{mAP} = \int_0^1 P(R) dR \quad (20)$$

To provide a more intuitive evaluation of the model's detection performance on small target ships, we also introduced the AP_S and AR_S metrics from the COCO index. APs and ARs are commonly used evaluation metrics in the COCO dataset [55]. Specifically, APs, AP_M, and AP_L refer to the average precision for small ($S < 32^2$), medium ($32^2 < S < 96^2$), and large ($S > 96^2$) objects, respectively. Similarly, AR_S, AP_M, and AP_L represent the recall rates for small, medium, and large objects, respectively. Additionally, the overall performance of the algorithm was evaluated by considering both the parameter count and computational complexity (GFlops) in order to determine its suitability for deployment on UAVs.

4.3. Ablation Experiment

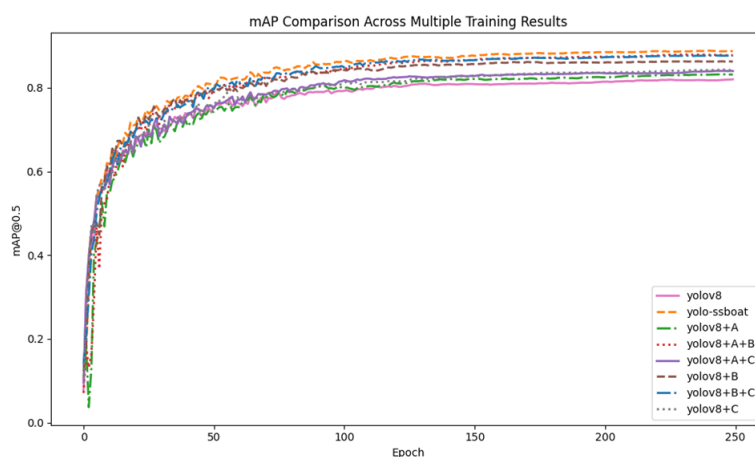
In this section, ablation experiments are performed using two datasets, ship_detection and Vessel, to evaluate the individual contribution of each module in the algorithm, using consistent training parameters in the ablation experiments to ensure the validity of the results. Using the results of the YOLOv8 network as a benchmark, eight different combinations of ablation experiments were used.

The results of the ablation experiments under the ship_detection dataset are shown in Table 4, and the mAP@50 (When the Intersection over Union (IoU) threshold is set to 0.5, the calculated mAP) curve is shown in Figure 10a. After using the C2f_DCNv3 feature extraction module, Precision (P) decreases slightly, and Recall (R) and mAP@50 both increase by 1.3% because the enhancement of feature extraction by deformable convolution can make it easier for the model to capture small and weak feature targets, thereby reducing missed detections and increasing the recall rate and mAP@50, but it may also lead to the interference features in the background being misdetected as targets, resulting in an increase in false positives, thereby slightly reducing the accuracy. The use of multi-scale weighted

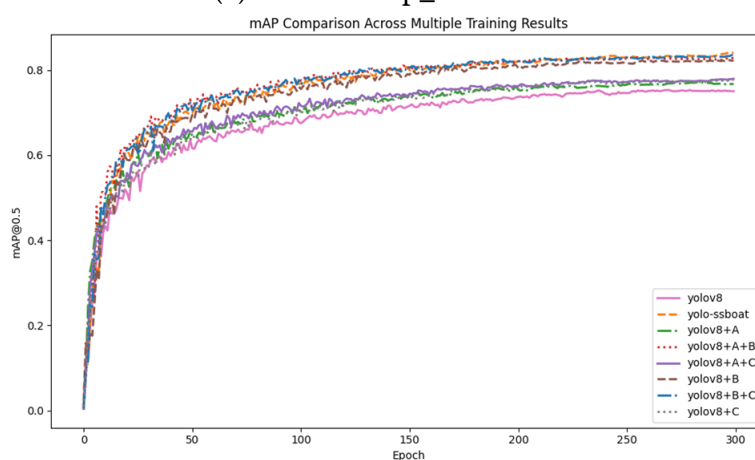
feature fusion network increases the accuracy by 2.6, the recall rate by 3.3%, and the mAP@50 by 4.4% due to the addition of small-scale feature layers and the weighting of different scale feature layers, and when using the dynamic detection head Dyhead_v3, because the detection head can be dynamically adjusted according to the size of the target ship, the accuracy increases by 2.4%, the recall rate increases by 2.5%, and the mAP@50 increases by 2.4%. Compared with the baseline model, the accuracy of YOLO-ssboat increased by 4.3%, the recall rate increased by 6.2%, the mAP@50 increased by 6.8%, and the accuracy of small targets increased by 10.4%, which proved that YOLO-ssboat has a good detection ability for small targets.

Table 4. Ablation Experiment on the ship_detection Dataset.

C2f_DCNv3	MSWPN	Dyhead_v3	P	R	mAP@50	APs	ARs
×	×	×	0.859	0.75	0.819	0.64	0.745
✓	×	×	0.849	0.763	0.832	0.65	0.751
×	✓	×	0.885	0.783	0.863	0.713	0.815
×	×	✓	0.883	0.775	0.843	0.672	0.756
✓	×	✓	0.886	0.776	0.844	0.67	0.758
×	✓	✓	0.886	0.807	0.876	0.726	0.812
✓	✓	×	0.89	0.794	0.878	0.729	0.822
✓	✓	✓	0.902	0.812	0.887	0.744	0.825



(a) Dataset ship_detection.



(b) Dataset Vessel.

Figure 10. Ablation Experiment mAP Curve Chart (Among them, A is the C2f_DCNv3 module, B is the MSWPN structure, and C is the Dyhead_v3 detection head).

The ablation results under the Vessel dataset are shown in Table 5. The Ablation Experiment on the Vessel Dataset and the comparison curve of the ablation mAP@50 are shown in Figure 10b. After using the C2f_DCNv3 feature extraction module, P increased by 1.9%, R increased by 2.4%, mAP@50 increased by 1.8%, and MSWPN increased by 7%, which greatly improved the problem of missed detection, and the dynamic detection head Dyhead_v3 increased P by 2%, R by 2.4%, and mAP@50 by 2.5% on the Vessel dataset. Compared with the baseline model, YOLO-ssboat increased P by 4.8%, R by 9.8, and mAP@50 by 8.9%. The Vessel dataset accounts for a large proportion of small targets, so the overall mAP@50 is lower than that of the ship_detection dataset, but the Vessel dataset improves the recognition accuracy of ships more than that of the ship_detection dataset.

Table 5. Ablation Experiment on the Vessel Dataset.

C2f_DCNv3	MSWPN	Dyhead_v3	P	R	mAP@50	APs	ARs
×	×	×	0.789	0.681	0.752	0.337	0.475
✓	×	×	0.808	0.705	0.77	0.355	0.482
×	✓	×	0.810	0.751	0.821	0.348	0.494
×	×	✓	0.809	0.705	0.777	0.361	0.477
✓	×	✓	0.829	0.708	0.779	0.359	0.479
×	✓	✓	0.826	0.773	0.835	0.373	0.490
✓	✓	×	0.803	0.771	0.829	0.357	0.493
✓	✓	✓	0.837	0.779	0.841	0.360	0.496

4.4. Comparison with Other Models

The experimental results are presented in Table 6, comparing YOLO-ssboat with several state-of-the-art object detection technologies, including SSD [21], Faster R-CNN [14], the YOLO series, and RT-DETR [22]. The results demonstrate that YOLO-ssboat outperforms all other methods on both the ship_detection and Vessel datasets, achieving the highest precision (P), recall (R), and mAP@50 indicators while requiring the fewest parameters and the second-lowest computational cost. Compared to YOLOv7 [17], which ranks second, YOLO-ssboat achieves a 2.8% improvement in mAP@50 on the ship_detection dataset, utilizing only 7% of the parameters and 24% of the computations of YOLOv7. Similarly, on the Vessel dataset, YOLO-ssboat achieves a 4.5% increase in accuracy, a 4% improvement in recall, and a 6.3% enhancement in mAP@50 compared to YOLOv9 [18], the second-best performer. Notably, YOLO-ssboat requires only 4.3% of the parameters and 9.47% of the computational cost of YOLOv9. These results highlight the suitability of YOLO-ssboat for deployment on UAV platforms to detect ships, given its superior performance and resource efficiency compared to other object detection algorithms.

Table 6. Comparison with Other Advanced Object Detection Algorithms.

Model	Ship_Detection Dataset			Vessel Dataset			Parameters (M)	GFlops
	P	R	mAP@50	P	R	mAP@50		
SSD	0.621	0.617	0.658	0.701	0.548	0.612	58.3	249.0
Faster-Rcnn	0.678	0.632	0.654	0.673	0.529	0.589	-	-
YOLOv5	0.871	0.809	0.85	0.782	0.688	0.753	7.06	16.5
YOLOv7	0.893	0.802	0.859	0.774	0.689	0.725	36.4	103.2
YOLOv9	0.876	0.783	0.844	0.792	0.739	0.778	60.7	264.9
YOLOv11	0.861	0.794	0.814	0.752	0.703	0.751	2.58	6.3
RT-DETR	0.783	0.725	0.781	0.676	0.632	0.651	32.8	108
OURs	0.902	0.812	0.887	0.837	0.779	0.841	2.56	25.1

To visually demonstrate the detection performance of YOLO-ssboat, three images with different backgrounds were selected for a comparative test, with the Intersection over Union (IoU) threshold and confidence level (Conf) set to 0.5. Figure 11 illustrates the detection of densely packed ships docked at a port. Detecting small, densely distributed ship targets not only evaluates the accuracy of small target detection but also tests the comprehensiveness of target identification. Faster R-CNN, SSD, and other YOLO variants exhibit missed detections of small ship targets (green circles in Figure 11 indicate missed detections). Benefiting from the dynamic detection capabilities of the Dyhead_v3 module, YOLO-ssboat successfully identifies nearly all ships in this scenario. Figure 12 shows the cases involving the detection of ships against a complex ocean background. In this context, small targets constitute only a minor proportion, and their features are challenging to extract. As shown in Figure 12, there are six small ship targets in total. YOLOv5 [16], YOLOv7, and YOLOv9 detect only five targets, while other detection algorithms also fail to detect all targets. However, thanks to the target-oriented design of the C2f_DCNv3 feature extraction module introduced in this study, YOLO-ssboat successfully detects all six ships. Figure 13 demonstrates target detection against a coastal background. Coastal detection is often disrupted by waves and rocks. In Figure 13, there are three small ship targets. The best-performing YOLOv5 and YOLOv9 detect only two of these targets, YOLOv11 and RT-DETR detect one target, while YOLOv7, SSD, and Faster R-CNN fail to detect any. Leveraging a small target detection layer and the weighted fusion of multi-scale features, YOLO-ssboat effectively enriches small target feature representation, enabling it to successfully detect all small ship targets in this scenario.

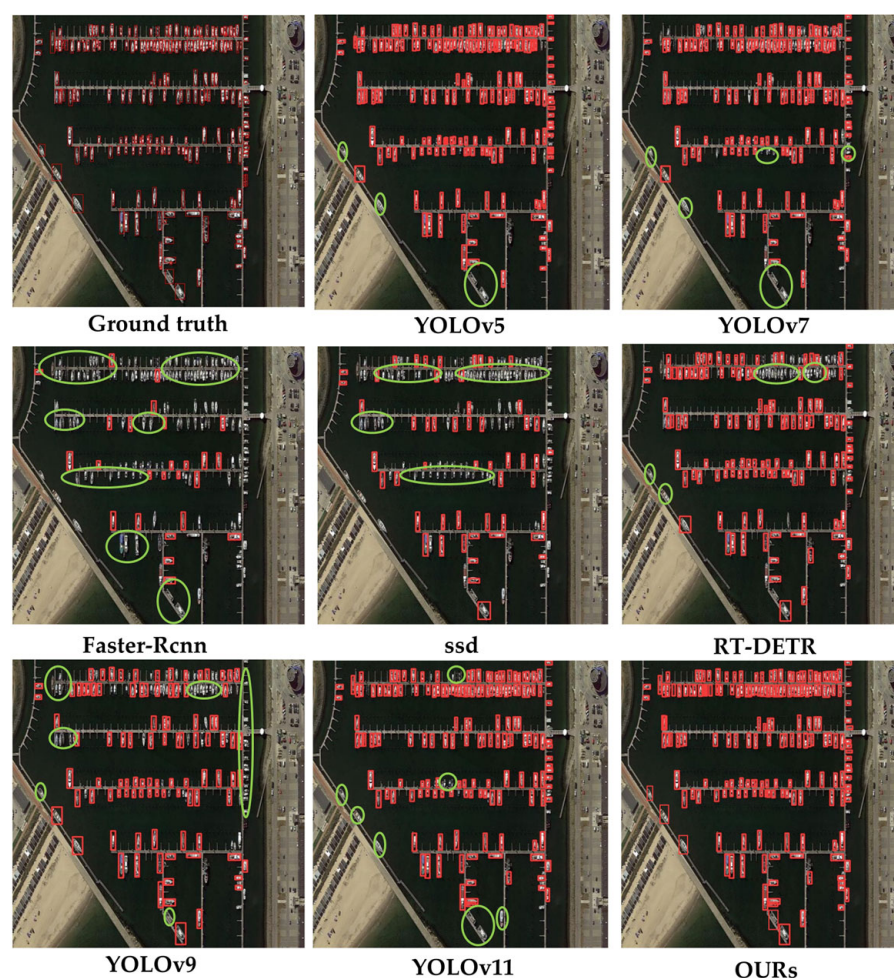


Figure 11. Dense Ship Detection Diagram.

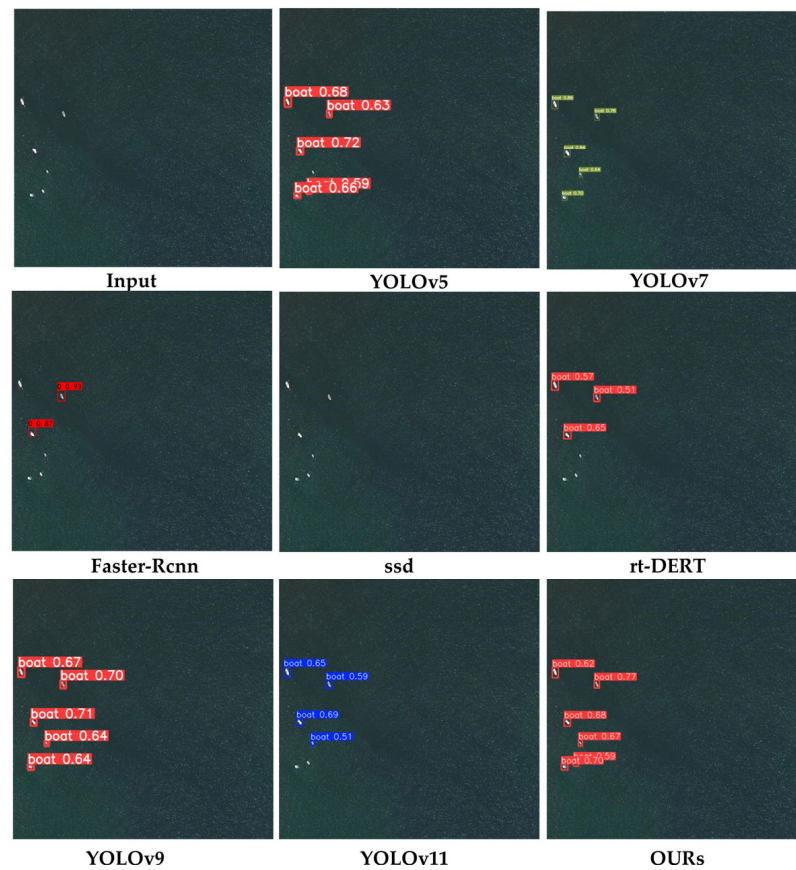


Figure 12. Small Target Ship Detection in Ocean Background.

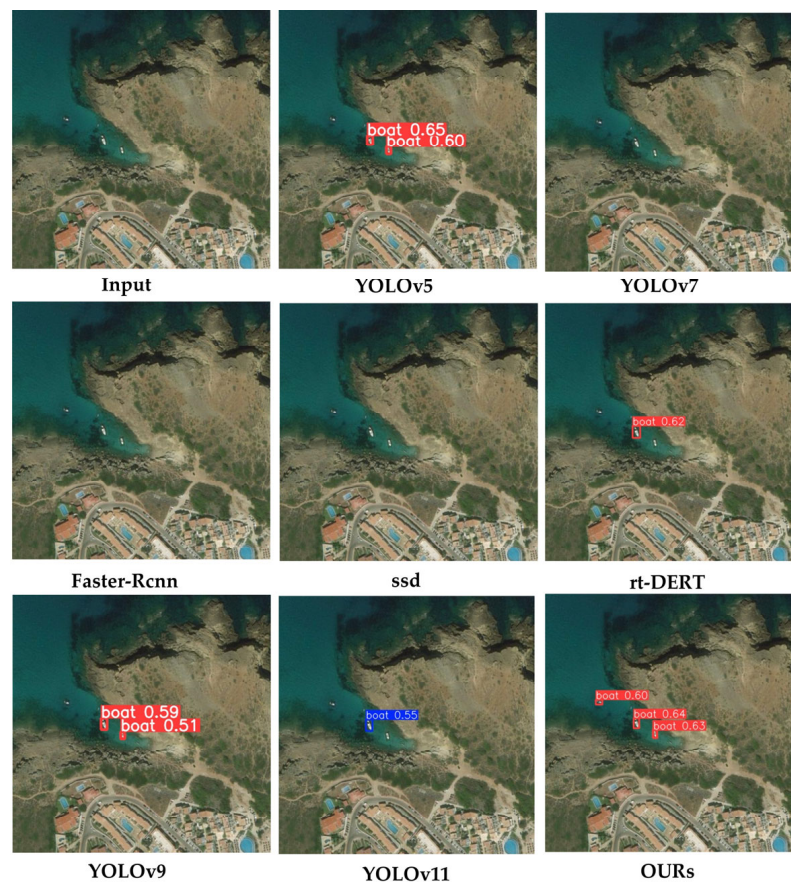


Figure 13. Ship Detection in Complex Coastal Environments.

4.5. Comparative Experiment on Improving Modules

4.5.1. Comparison of Different Feature Fusion Methods

To further evaluate the effectiveness of the deformable convolution feature extraction module in ship small-object detection tasks, we integrated the C2f_DCNv3 module into the YOLOv8 framework and compared its performance with several commonly used improved feature extraction modules. C2f-DSCConv [56] and C2f-RFAConv [57]. C2f-DSCConv incorporates Depthwise Separable Convolution, which significantly reduces parameter complexity through the combination of depthwise convolution and 1×1 pointwise convolution. This design allows the model to focus more on the local features of small targets rather than redundant background information. C2f-RFAConv is a feature extraction module that embeds the Receptive Field Attention (RFA) mechanism. It enhances the model's ability to concentrate on small target regions by reducing the influence of irrelevant background areas. The validation set from the ship_detection dataset was selected as the experimental dataset to assess the performance differences of various feature extraction modules in detecting small ships in complex backgrounds. The experimental results, as shown in Table 7, illustrate the performance of different modules across various metrics. The comparative analysis demonstrates the superiority of C2f_DCNv3 in extracting features for ship small-object detection.

Table 7. Comparison of Feature Extraction Modules.

Model	P	R	mAP@50	mAP@50-95
Yolov8	0.859	0.75	0.819	0.411
C2f-DSCConv	0.847	0.742	0.809	0.404
C2f-RFAConv	0.854	0.735	0.812	0.406
C2f_DCNv3	0.849	0.763	0.832	0.414

4.5.2. Comparison of Different Feature Fusion Methods

To validate the effectiveness of our proposed fusion structure, we conducted detailed comparative experiments on the ship_detection dataset. Keeping other modules unchanged, we compared the performance of different fusion structures. The results are shown in Table 8. We systematically compared the performance of the SDI [58] and HSFPN [59] feature fusion structures. SDI (Scale-aware Deep Integration) enhances the representation of small targets by dynamically integrating multi-level features through deep fusion across multiple scales. HSFPN, an improved version of the traditional Feature Pyramid Network (FPN), introduces a hierarchical scale-aware mechanism to optimize feature flow within the pyramid, thereby enhancing multi-scale detection capabilities. The results indicate that, compared to other advanced modules, our proposed feature fusion structure demonstrates a significant advantage in detecting small ship targets.

Table 8. Comparison of Feature fusion Modules.

Model	P	R	mAP@50	mAP@50-95
Yolov8	0.859	0.75	0.819	0.411
SDI	0.834	0.746	0.804	0.358
HSFPN	0.818	0.739	0.798	0.397
(ours)	0.885	0.783	0.863	0.423

4.5.3. Comparison Experiment of Different Improved Detection Heads

To verify the effectiveness of the improved detection head, we modified only the detection head in the YOLOv8 network to evaluate the impact of Dyhead-DCNv3. The proposed

detection head is compared against RepHead [60] and ASFFHead [61]. RepHead is based on the structural re-parameterization technique, employing a multi-branch architecture during training to enhance feature extraction, which is then merged into a single-branch structure during inference to improve efficiency. ASFFHead leverages Adaptive Spatial Feature Fusion to dynamically adjust the weights of features from different pyramid levels, effectively addressing feature conflicts in multi-scale object detection. As shown in Table 9, experimental results on the ship_detection dataset demonstrate that Dyhead-DCNv3 outperforms traditional and other improved detection heads.

Table 9. Comparison of Detection Head Modules.

Model	P	R	mAP@50	mAP@50-95
Yolov8	0.859	0.75	0.819	0.411
RepHead	0.839	0.755	0.814	0.407
ASFFHead	0.852	0.739	0.813	0.409
Dyhead-DCNv3	0.883	0.775	0.843	0.418

4.6. Ship Detection Based on Wake Characteristics

Table 10 presents the quantitative analysis of the improvement brought by the Tail Wave Detection Method for small vessel target detection. By detecting tail waves to infer vessel locations, the method significantly enhances detection performance, achieving a 3.4% increase in precision, a 4.0% increase in recall, and a 4.1% improvement in mAP@50; TWDM is the Tail Wave Detection Method.

Table 10. Quantitative analysis of wake wave detection methods (TWDM is Tail Wave Detection Method).

Model	P	R	mAP@50	APs	ARs
YOLO-ssboat	0.902	0.812	0.887	0.744	0.825
YOLO-ssboat+TWDM	0.936	0.852	0.928	0.781	0.861

Two groups of such ships were selected for analysis to evaluate the performance of YOLO-ssboat in detecting ships obscured by wake waves. To visually illustrate the detection performance, the confidence threshold for ship detection based on wake wave features was set to 0.2, significantly lower than the typical threshold of 0.5 used in practical applications. Consequently, ship targets with confidence levels below 0.5 in the following results would be missed in real-world applications.

Figure 14 depicts four ships obscured by wake waves. The second-best performer, YOLOv5, detected only three ships due to the small size of the targets and the occlusion caused by the wake waves. SSD, Faster R-CNN, and other YOLO-based detection methods failed to detect all targets. In contrast, YOLO-ssboat successfully identified all small ship targets with confidence levels exceeding 0.5, ensuring no missed detections in practical scenarios. Figure 15 illustrates the detection of a single ship obscured by wake waves. Except for Faster R-CNN, the confidence levels of all other detection algorithms fell below 0.5. The detection confidence of YOLO-ssboat reached an impressive 0.96, significantly surpassing that of Faster R-CNN. These results demonstrate that YOLO-ssboat exhibits superior detection performance in identifying ships obscured by wake waves.

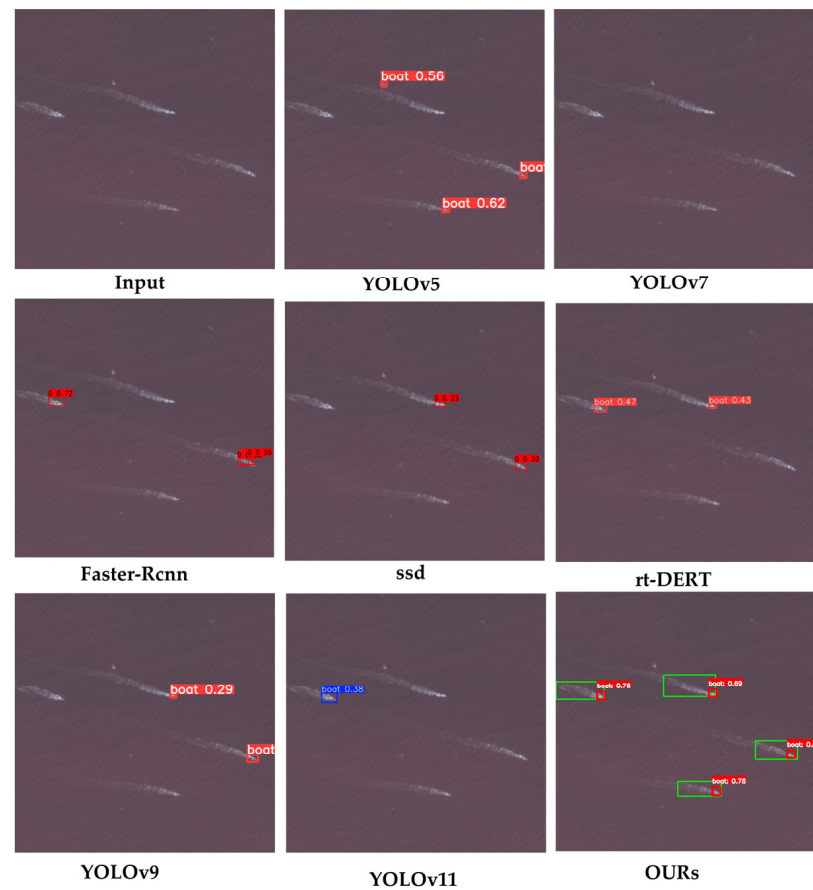


Figure 14. Detection of Multiple Ships with Wake Waves. (The green box shows the detected wake).

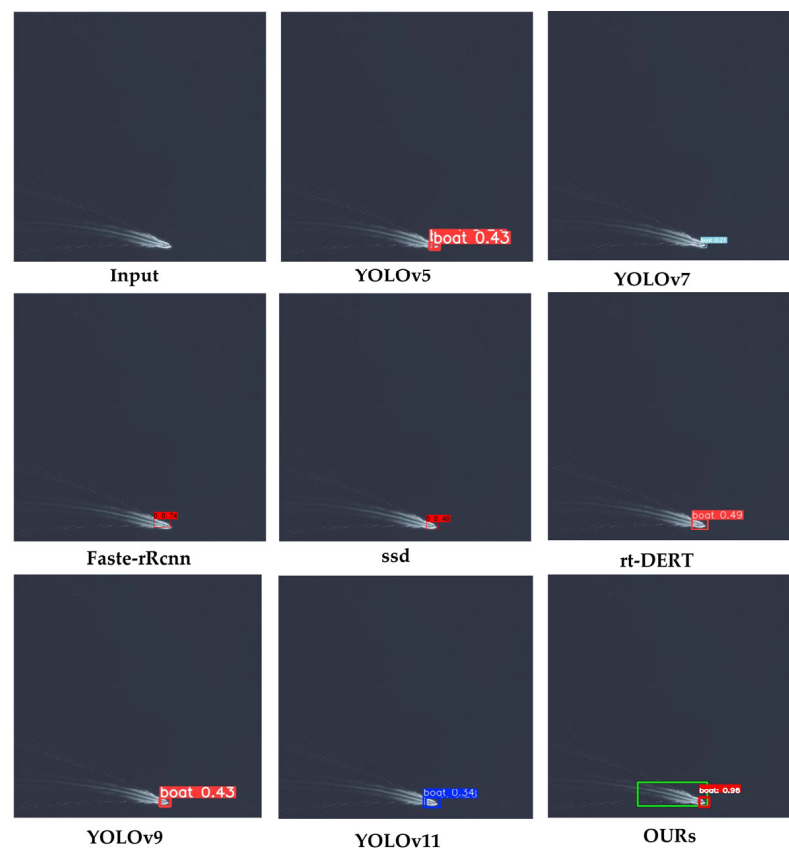


Figure 15. Detection of a Single Ship with Wake Wave. (The green box shows the detected wake).

4.7. Adversarial Training

To investigate the impact of adversarial samples on ship small-object detection networks, we used the PGD attack on the ship_detection dataset to generate adversarial samples with varying perturbation intensities. We then trained models using natural samples and adversarial samples of different intensities to examine their recognition performance on both types of data. As shown in Table 11, models trained solely on natural samples maintained a high recognition rate for natural inputs but suffered a significant drop in performance when subjected to adversarial attacks. In contrast, models trained on adversarial samples demonstrated improved robustness against attacks but exhibited weaker performance on natural samples.

Table 11. Adversarial Training with Different Perturbation Intensities.

Adversarial Training	Nature Samples			Adversarial Samples (Per = 0.01)		
	P	R	mAP@50	P	R	mAP@50
Perturbation = 0	0.859	0.75	0.819	0.801	0.58	0.665
Perturbation = 0.01	0.772	0.593	0.65	0.835	0.727	0.786
Perturbation = 0.03	0.745	0.492	0.557	0.835	0.701	0.771

To enhance the model's robustness on both natural and adversarial samples, we adopted a mixed training approach that incorporates both types of data. We further explored the impact of different adversarial-to-natural sample ratios on model performance. As indicated in Tables 12 and 13, increasing the proportion of adversarial samples in the training set improves the model's robustness against adversarial attacks but at the cost of reduced detection efficiency on natural samples. Our experimental results demonstrate that a balanced 50% mixing ratio yields optimal comprehensive detection performance, achieving the best trade-off between adversarial robustness and natural sample recognition accuracy.

Table 12. Adversarial Training with Different Ratios (PGD = 0.01).

Adversarial Samples Ratio	Natural Samples			Adversarial Samples		
	P	R	mAP@50	P	R	mAP@50
(20%)	0.825	0.762	0.809	0.834	0.681	0.76
(35%)	0.83	0.726	0.787	0.822	0.696	0.766
(50%)	0.827	0.727	0.787	0.834	0.695	0.772
(65%)	0.815	0.724	0.781	0.812	0.724	0.776
(80%)	0.806	0.717	0.773	0.845	0.699	0.774

Table 13. Adversarial Training with Different Ratios (PGD = 0.03).

Adversarial Samples Ratio	Natural Samples			Adversarial Samples		
	P	R	mAP@50	P	R	mAP@50
(20%)	0.84	0.736	0.801	0.824	0.645	0.732
(35%)	0.829	0.735	0.795	0.82	0.688	0.757
(50%)	0.806	0.696	0.792	0.809	0.696	0.765
(65%)	0.839	0.717	0.786	0.832	0.683	0.759
(80%)	0.832	0.719	0.783	0.814	0.703	0.769

This finding suggests that carefully calibrated hybrid training can effectively reconcile the inherent tension between adversarial defense capability and natural sample processing efficiency in deep learning-based detection systems.

As illustrated in Figure 16, a network model trained on natural samples demonstrates robust detection performance when identifying such samples. The heatmap visualization

reveals that the model exhibits a high degree of attention to ship targets. However, as shown in Figure 17, when the samples are subjected to adversarial attacks, the presence of adversarial perturbations alters the model's attention regions, leading to a failure in detecting ship targets. After undergoing adversarial training, the model acquires defensive capabilities against adversarial samples, enabling its attention regions to realign with the ship targets, as depicted in Figure 18. Compared to models trained exclusively on natural samples, adversarially trained models exhibit superior robustness and stability, achieving higher accuracy in detecting perturbed samples.

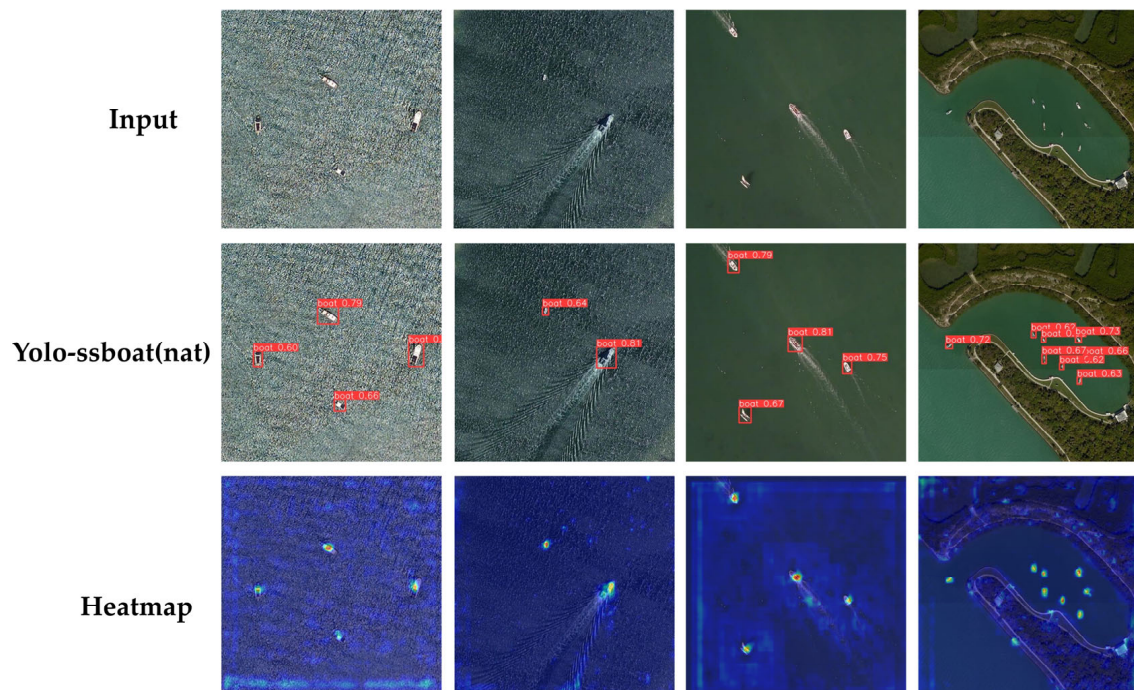


Figure 16. Natural sample-trained model detects natural samples.

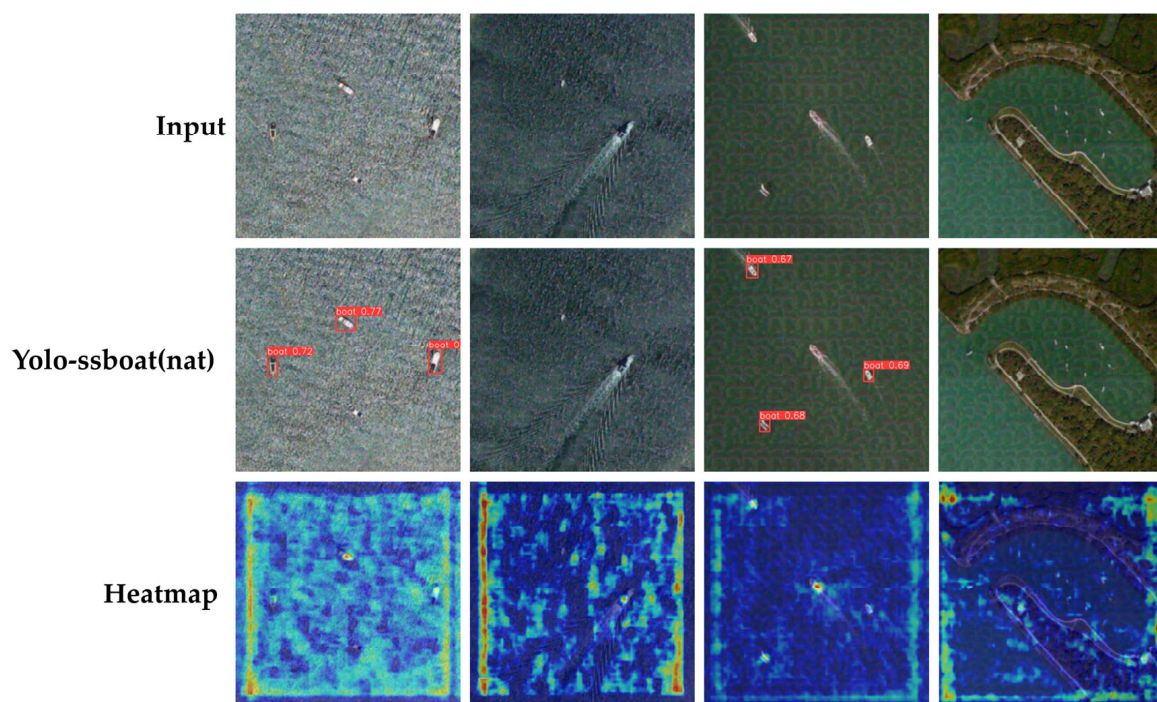


Figure 17. Natural sample-trained model detects adversarial samples.

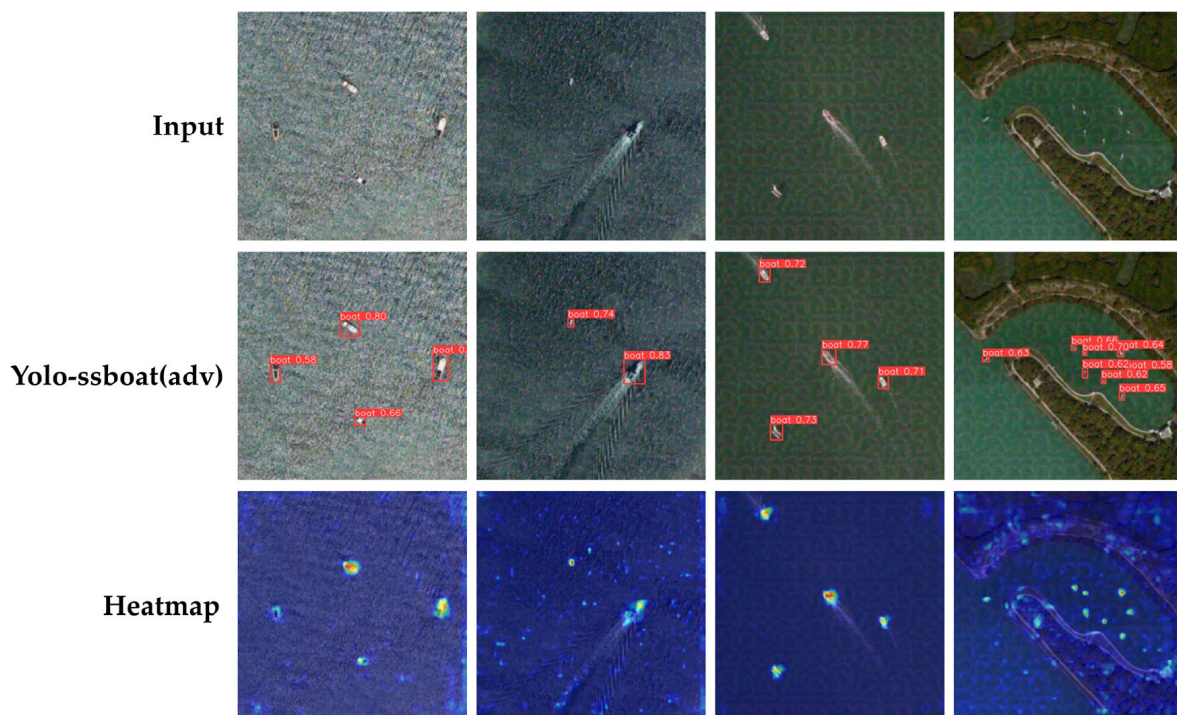


Figure 18. An adversarial sample-trained model detects adversarial samples.

5. Discussion

Detection of small ship targets in complex marine environments faces numerous challenges. For example, the small target size makes feature extraction difficult; variations in the angles of remote sensing imagery and the diverse morphologies of ships compromise the algorithm's stability; and ship-generated wakes can obscure targets, thereby increasing detection difficulty. To address these challenges, YOLO-ssboat employs a novel feature extraction module, C2f_DCNv3, to enhance the efficiency of target feature extraction, adopts the MSWPN multi-scale feature fusion structure to enrich small target features, and utilizes Dyhead_v3 to strengthen the feature representation of small ships. As illustrated in Figures 11–13, YOLO-ssboat demonstrates significant advantages over other detection methods in scenarios with complex backgrounds and densely populated ship targets.

To tackle the issue of target occlusion by wakes, this study proposes a gradient flow method that first detects the wakes generated by ship movement and then uses the gradient flow to localize the ship. As shown in Figures 14 and 15, the gradient flow method reliably detects ships that are missed by other detection methods.

Furthermore, to enhance the robustness of YOLO-ssboat, this study also investigates the impact of adversarial sample attacks and employs adversarial training to improve the model's visual defense against such perturbations. As shown in Table 11, adding adversarial samples to the training dataset can enhance the model's resistance to adversarial attacks, though it may reduce recognition efficiency for natural samples. As demonstrated in Tables 12 and 13, when the number of natural and adversarial samples in the training dataset is balanced, the model achieves optimal overall performance in detecting both types of samples.

6. Conclusions

In this study, we investigate the challenges associated with detecting small ship targets in remote sensing imagery, focusing on three key aspects: feature extraction, feature fusion, and feature representation. In addition, to address the susceptibility of small ship targets to

wake interference and adversarial attacks, we propose a novel Tail Wave Detection Method and conduct adversarial training to enhance the robustness of the algorithm, thereby achieving precise ship localization. The Tail Wave Detection Method represents a novel approach that integrates gradient computation with target detection techniques to achieve vessel localization through gradient variation analysis. This methodology constitutes the first reported instance of such a combined application in the field of ship target detection.

Experimental results on the ship_detection and Vessel datasets demonstrate that, compared with state-of-the-art object detection algorithms, YOLO-ssboat improves detection accuracy without increasing the number of parameters. Ablation studies indicate that the proposed deformable convolution feature extraction module, multi-scale weighted feature fusion module, and dynamic detection head play significant roles in enhancing detection accuracy. Moreover, comparative analyses with other object detection models reveal that the wake gradient method substantially improves the detection accuracy of small ship targets affected by wake interference. Incorporating adversarial samples into the training dataset bolsters the network's defense against adversarial perturbations, with optimal overall performance observed when adversarial samples account for 50% of the training data.

In summary, the proposed YOLO-ssboat model exhibits high practical value and is well-suited for the maritime surveillance of small ship targets. However, its computational cost is higher compared to YOLOv8. Future work will focus on optimizing the model using lightweight techniques for real-time deployment and extending its application to other remote sensing targets, such as those captured by unmanned aerial vehicles.

Author Contributions: Conceptualization, Y.Z. and X.W.; methodology, Y.Z. and X.W.; validation, Y.Z. and X.W.; investigation, X.W. and J.Z.; writing—original draft preparation, X.W.; writing—review and editing, Y.Z., X.W., J.Z. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the National Key Laboratory of Unmanned Aerial Vehicle Technology in NPU (WR2024132), the National Natural Science Foundation of China (61801018), the Youth Innovation Team of Shaanxi University, and the Fundamental Research Funds for the Central Universities (FRF-GF-20-13B).

Data Availability Statement: The Ship_detection dataset and Vessel dataset can be accessed at <https://github.com/asidh/dataset>. Accessed on 6 May 2025.

Conflicts of Interest: Author H.W. was employed by Shanxi Intelligent Transportation Institute Co. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [\[CrossRef\]](#)
2. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones* **2022**, *6*, 147. [\[CrossRef\]](#)
3. Kanellakis, C.; Nikolakopoulos, G. Survey on Computer Vision for UAVs: Current Developments and Trends. *J. Intell. Robot. Syst.* **2017**, *87*, 141–168. [\[CrossRef\]](#)
4. Karachristos, K.; Koukiou, G.; Anastassopoulos, V. PolSAR Cell Information Representation by a Pair of Elementary Scatterers. *Remote Sens.* **2022**, *14*, 695. [\[CrossRef\]](#)
5. Pan, D.; Gao, X.; Dai, W.; Fu, J.; Wang, Z.; Sun, X.; Wu, Y. SRT-Net: Scattering Region Topology Network for Oriented Ship Detection in Large-Scale SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5202318. [\[CrossRef\]](#)
6. Wen, G.; Cao, P.; Wang, H.; Chen, H.; Liu, X.; Xu, J.; Zaiane, O. MS-SSD: Multi-scale single shot detector for ship detection in remote sensing images. *Appl. Intell.* **2023**, *53*, 1586–1604. [\[CrossRef\]](#)

7. Fan, X.; Hu, Z.; Zhao, Y.; Chen, J.; Wei, T.; Huang, Z. A Small-Ship Object Detection Method for Satellite Remote Sensing Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 11886–11898. [\[CrossRef\]](#)
8. Stokkel, X.L.X. Detecting Humans from a Top-Down Perspective Using an Unmanned Aerial Vehicle. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 2015.
9. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [\[CrossRef\]](#)
10. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [\[CrossRef\]](#)
11. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, NY, USA, 20–26 June 2005; pp. 886–893.
12. Wang, Y.R.; Zhu, X.L.; Wu, B. Automatic Detection of Individual Oil Palm Trees from UAV Images Using HOG Features and an SVM Classifier. *Int. J. Remote Sens.* **2019**, *40*, 7356–7370. [\[CrossRef\]](#)
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
15. 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. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
17. Mokayed, H.; Nayeibastaneh, A.; De, K.; Sozos, S.; Hagner, O.; Backe, B. Nordic Vehicle Dataset (NVD): Performance of vehicle detectors using newly captured NVD from UAV in different snowy weather conditions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Vancouver, BC, Canada, 17–24 June 2023; pp. 5314–5322.
18. Wang, C.; Bochkovskiy, A.; Liao, H.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, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
19. Wang, C.-Y.; Yeh, I.H.; Liao, H.-Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616.
20. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *arXiv* **2024**, arXiv:2405.14458.
21. Khanam, R.; Hussain, M. YOLOv11: An Overview of the Key Architectural Enhancements. *arXiv* **2024**, arXiv:2410.17725.
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
23. 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 2024, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
24. Yi, D.L.; Shun, S.Z.; Wen, Q. A Lightweight Faster R-CNN for Ship Detection in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5.
25. Xu, C.A.; Su, H.; Gao, L.; Wu, J.F.; Yan, W.J.; Jian, T.; Wang, H.Y. Feature Aligned Ship Detection Based on Improved RPDet in SAR Images. *Displays* **2022**, *74*, 102191.
26. Li, Z.; Kong, D.; Liu, J.; Sun, X.; Du, Q.; Zhang, L. Multi-Scale Accurate Ship Detection Network Driven by Multi-Attention Fusion for Complex 183 Maritime Backgrounds. *IEEE Sens. J.* **2024**, *24*, 9208–9216. [\[CrossRef\]](#)
27. Wen, R.L.; Wei, Q.Y.; Xin, Q.C.; Lu, Y.X. An Enhanced CNN-Enabled Learning Method for Promoting Ship Detection in Maritime 185 Surveillance System. *Ocean Eng.* **2021**, *235*, 109435.
28. Kang, M.; Ji, K.; Leng, X.; Lin, Z. Contextual region-based convolutional neural network with multilayer fusion for sar ship detection. *Remote Sens.* **2017**, *9*, 860. [\[CrossRef\]](#)
29. Li, S.; Fu, X.; Dong, J. Improved ship detection algorithm based on yolox for sar outline enhancement image. *Remote Sens.* **2022**, *14*, 4070. [\[CrossRef\]](#)
30. Yu, C.; Shin, Y. Sar ship detection based on improved yolov5 and bifpn. *ICT Express* **2023**, *10*, 28–33. [\[CrossRef\]](#)
31. Chen, Z.; Liu, C.; Filaretov, V.F.; Yukhimets, D.A. Multi-scale ship detection algorithm based on yolov7 for complex scene sar images. *Remote Sens.* **2023**, *15*, 2071. [\[CrossRef\]](#)
32. Li, D.; Liang, Q.; Liu, H.; Liu, Q.; Liu, H.; Liao, G. A novel multidimensional domain deep learning network for sar ship detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5203213. [\[CrossRef\]](#)

33. Chen, C.; He, C.; Hu, C.; Pei, H.; Jiao, L. A deep neural network based on an attention mechanism for sar ship detection in multiscale and complex scenarios. *IEEE Access* **2019**, *7*, 104848–104863. [\[CrossRef\]](#)
34. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the IEEE International Conference on 200 Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 764–773.
35. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable ConvNets V2: More Deformable, Better Results. In Proceedings of the IEEE/CVF 202 Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
36. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 18–22 June 2023; pp. 14408–14419.
37. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
38. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
39. Ghiasi, G.; Lin, T.-Y.; Pang, R.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7036–7045.
40. 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.
41. Pang, J.; Chen, K.; Shi, J.; Feng, H.; Ouyang, W.; Lin, D. Libra R-CNN: Towards Balanced Learning for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 821–830.
42. Wang, X.; Zhang, S.; Yu, Z.; Feng, L.; Zhang, W. Scale-Equalizing Pyramid Convolution for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 13356–13365.
43. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
44. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6568–6577.
45. Qiu, H.; Ma, Y.; Li, Z.; Liu, S.; Sun, J. BorderDet: Border Feature for Dense Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 549–564.
46. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv:1905.05055. [\[CrossRef\]](#)
47. Dai, X.; Chen, Y.; Xiao, B. Dynamic Head: Unifying Object Detection Heads with Attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 19–25 June 2021; pp. 7373–7382.
48. Szegedy, C.; Zaremba, W.; Sutskever, I. Intriguing Properties of Neural Networks. *arXiv* **2013**, arXiv:1312.6199.
49. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. *arXiv* **2014**, arXiv:1412.6572.
50. Madry, A.; Makelov, A.; Schmidt, L. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2017**, arXiv:1706.06083, 234.
51. Moosavi, D.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
52. Croce, F.; Hein, M. Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 12–18 July 2020; pp. 2206–2216.
53. Velishala, S. Ship_Detection. Computer Vision Project, Roboflow, Version 1. Available online: https://universe.roboflow.com/sharath-velishala-pgpib/ship_detection-uirank (accessed on 24 August 2023).
54. Virag, I. Vessel_Detection_416_416. Computer Vision Project, Version 1. Available online: https://universe.roboflow.com/igor-virag-8tftp1/vessel_detection_416_416 (accessed on 6 September 2021).
55. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
56. Wang, S.; Xiang, J.; Chen, D. A Method for Detecting Tomato Maturity Based on Deep Learning. *Appl. Sci.* **2024**, *14*, 11111. [\[CrossRef\]](#)
57. Wei, H.; Zhao, L.; Li, R. RFACConv-CBM-ViT: Enhanced Vision Transformer for Metal Surface Defect Detection. *J. Supercomput.* **2025**, *81*, 1–38. [\[CrossRef\]](#)
58. Peng, Y.; Sonka, M.; Chen, D.Z. U-Net v2: Rethinking the Skip Connections of U-Net for Medical Image Segmentation. *arXiv* **2023**, arXiv:2311.17791.

59. Guo, J.; Xu, F.; Wang, S. Pavement Disease Object Detection for UAVs Based on Improved YOLOv8. In Proceedings of the Fourth International Conference on Computer Vision and Pattern Analysis (ICCPA 2024), Anshan, China, 17–19 May 2024; Volume 13256, pp. 136–143.
60. Ding, X.; Zhang, X.; Ma, N. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13733–13742.
61. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.

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.