Acadlore Transactions on AI and Machine Learning
https://www.acadlore.com/journals/ATAIML

LMS-YOLO: A StarNet-Enhanced Lightweight Framework for Robust Marine Object Detection in Complex Water Surface Environments

Yuhan Sun[1,2*], Xin Liu[1,2], Qingfa Zhang[1,2], Mingzhi Shao[1,2], Tengwen Zhang[1,2]

[1] Shandong Jiaotong University, 264210 Weihai, China
[2] Weihai Institute of Marine Information Science and Technology, 264200 Weihai, China

[*] Correspondence: Yuhan Sun (23226023@stu.sdjtu.edu.cn)

Received: 08-22-2025 **Revised:** 10-05-2025 **Accepted:** 10-12-2025

Citation: Y. H. Sun, X. Liu, Q. F. Zhang, M. Z. Shao, and T. W. Zhang, "LMS-YOLO: A starnet-enhanced lightweight framework for robust marine object detection in complex water surface environments," *Acadlore Trans. Mach. Learn.*, vol. 4, no. 4, pp. 247–262, 2025. https://doi.org/10.56578/ataiml040402.

Abstract: Accurate and efficient detection of small-scale targets on dynamic water surfaces remains a critical challenge in the deployment of unmanned surface vehicles (USVs) for maritime applications. Complex background interference—such as wave motion, sunlight reflections, and low contrast—often leads to missed or false detections, particularly when using conventional convolutional neural networks. To address these issues, this study introduces LMS-YOLO, a lightweight detection framework built upon the YOLOv8n architecture and optimized for real-time marine object recognition. The proposed network integrates three key components: (1) a C2f-SBS module incorporating StarNet-based Star Blocks, which streamlines multi-scale feature extraction while reducing parameter overhead; (2) a Shared Convolutional Lightweight Detection Head (SCLD), designed to enhance detection precision across scales using a unified convolutional strategy; and (3) a Mixed Local Channel Attention (MLCA) module, which reinforces context-aware representation under complex maritime conditions. Evaluated on the WSODD and FloW-Img datasets, LMS-YOLO achieves a 5.5% improvement in precision and a 2.3% gain in mAP@0.5 compared to YOLOv8n, while reducing parameter count and computational cost by 37.18% and 34.57%, respectively. The model operates at 128 FPS on standard hardware, demonstrating its practical viability for embedded deployment in marine perception systems. These results highlight the potential of LMS-YOLO as a deployable solution for high-speed, high-accuracy marine object detection in real-world environments.

Keywords: Marine object detection; YOLOv8; StarNet; Lightweight neural network; Shared convolution

1 Introduction

In recent years, the rapid integration of USVs into a wide range of maritime applications—including environmental monitoring, emergency rescue, oceanographic surveys, and waterborne logistics—has significantly raised the demand for accurate perception capabilities in complex aquatic environments. Among the core perception tasks, marine object detection plays a decisive role, directly impacting the autonomous navigation, situational awareness, and operational safety of USVs. However, detecting small-scale floating objects such as buoys, debris, and watercraft under dynamic maritime conditions remains a persistent challenge. Factors such as cluttered backgrounds, variable lighting, low contrast, and the inherently small size of many targets contribute to high false-positive and false-negative rates. These difficulties have led to a growing research focus on the development of deep learning–based object detection techniques tailored for marine scenarios.

Contemporary object detection methods based on deep convolutional neural networks are typically categorized into two paradigms: two-stage and one-stage detectors. Two-stage frameworks, such as Faster R-CNN, first generate region proposals and then perform classification and regression tasks [1]. While offering high detection accuracy, these models are often unsuitable for real-time deployment due to their heavy computational burden. In contrast, one-stage detectors perform classification and localization in a unified manner, typically using predefined anchor boxes, and are known for their speed and simplicity. The YOLO family has emerged as a representative of one-stage models, achieving impressive results across a variety of vision tasks [2].

From YOLOv3 [3] to the more recent YOLOv4 [4], YOLOv5 [5], YOLOv6 [6], and YOLOv7 [7], successive versions have brought notable improvements in detection accuracy, robustness, and computational efficiency.

Nevertheless, existing models continue to face challenges when applied to small object detection in maritime environments, where maintaining a balance between accuracy and model complexity is crucial. For example, Huang et al. [8] proposed an improved YOLOv4-based ship detection algorithm that replaces the original SPP structure with an RFB_s module to expand multi-scale receptive fields and introduces the CBAM attention mechanism to enhance feature representation. The method significantly improves detection accuracy for small ships and reduces background interference, though it still shows performance degradation in complex maritime scenes. Zhang et al. [9] developed SE-NMS-YOLOv5 by integrating SE attention to strengthen channel features, optimizing the NMS process to alleviate missed detections under occlusion,This approach improves detection precision and recall in multi-target maritime environments but remains limited by environmental variations such as reflection and light fluctuation. Jiang et al. [10] developed YOLOv7-Ship by incorporating CA-M and ODConv modules, achieving improvements in both accuracy and real-time performance at the cost of increased architectural complexity. Li et al. [11] proposed YOLO-WSD, which employed a C2F-E module and WIoU loss function to enhance feature representation but still faced challenges in multi-scale feature fusion. More recently, Wang and Zhao [12] proposed YOLOv8-MSS, incorporating an additional detection head and SENetV2 module to improve performance on small targets, though issues related to redundant computations and latency persisted.

Despite these advancements, several core limitations remain unresolved: insufficient lightweight design, reduced robustness in dynamic environments, and inadequate multi-scale feature fusion. These issues are particularly pressing in embedded deployment scenarios—such as those involving USVs and UAVs—where computational resources are constrained. Deploying large-scale detection networks under such conditions can cause processing delays, hinder real-time decision-making, and even reduce the operational lifespan of the onboard hardware due to sustained high-load computation. Furthermore, environmental challenges—such as continuous wave motion, strong backlighting, and surface reflections—compound the difficulty of detecting small floating targets reliably.

To address these persistent limitations, this paper presents LMS-YOLO, a lightweight and high-performance marine object detection framework built upon the YOLOv8n backbone. The proposed architecture introduces three core innovations that collectively enhance detection accuracy while preserving computational efficiency:

(1) C2f-SBS Module: A modified backbone structure that integrates Star Blocks from StarNet [13], reducing parameter count and memory footprint while reinforcing multi-scale feature extraction through a streamlined convolutional design.

(2) SCLD Head: A novel detection head architecture that shares convolutional parameters across multiple scales, significantly reducing redundancy without compromising localization or classification performance.

(3) MLCA Module: A lightweight attention mechanism [14] that synergistically combines local and global spatial-channel features to improve detection robustness under challenging maritime conditions, such as surface reflections, wave distortion, and fog.

By incorporating these modules, LMS-YOLO achieves an improved trade-off between accuracy and model compactness, making it suitable for real-time deployment in embedded marine perception systems. The effectiveness of the proposed framework is validated through extensive experiments on public benchmarks and real-world datasets.

2 YOLOv8 Algorithm

The YOLOv8 framework is structured into four key components: input processing, backbone network, neck, and output head, as illustrated in Figure 1. At the input stage, the model applies a combination of preprocessing techniques including Mosaic data augmentation, adaptive image resizing, and grayscale padding, aiming to improve generalization and maintain spatial consistency across variable input dimensions.

The backbone is responsible for hierarchical feature extraction and is constructed using a series of convolutional layers, C2f blocks, and a Spatial Pyramid Pooling—Fast (SPPF) module. These elements collectively enhance the network's ability to capture both local and global contextual information. The neck adopts a Path Aggregation Network (PANet) structure to facilitate the fusion of multi-scale feature maps via up-sampling and down-sampling operations. Feature representations from different levels are integrated to better capture objects of varying sizes.

In the output head, YOLOv8 employs a decoupled architecture that processes classification and regression tasks independently. This separation allows the network to optimize the two tasks more effectively. During training, the model assigns positive and negative samples based on Intersection-over-Union (IoU) thresholds, and calculates the overall loss using a composite formulation: binary cross-entropy (BCE) for classification and a combination of Distribution Focal Loss (DFL) and Complete Intersection-over-Union (CIoU) for bounding box regression [15, 16].

Compared with its predecessor YOLOv5, YOLOv8 incorporates several important architectural changes aimed at improving both efficiency and accuracy. In the backbone, YOLOv8 replaces the C3 modules found in YOLOv5 with C2f blocks, which reduce the number of intermediate channels and promote denser gradient propagation. This leads to a more compact network with enhanced convergence characteristics. In the neck, the up-sampling mechanism is redesigned by eliminating convolutional operations prior to up-sampling; instead, a down-sampling operation is first applied, followed by feature aggregation through the updated C2f modules.

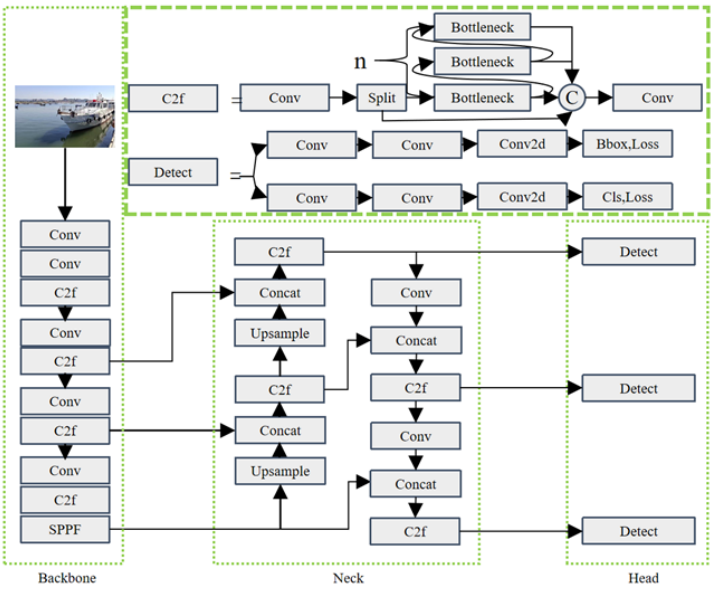


Figure 1. The structure of YOLOv8, highlighting its anchor-free head and improved C2f-based backbone

Perhaps most notably, the prediction head in YOLOv8 shifts away from the traditional anchor-based detection paradigm. The model adopts an anchor-free approach, directly predicting the center coordinates of objects along with their relative width and height. This simplification eliminates the need for predefined anchor boxes, streamlines the matching strategy during training, and contributes to improved inference speed and accuracy [17].

Overall, these enhancements enable YOLOv8 to achieve a favorable trade-off between detection precision and computational efficiency, making it a strong candidate for real-time object detection in both general-purpose and resource-constrained applications.

3 LMS-YOLO

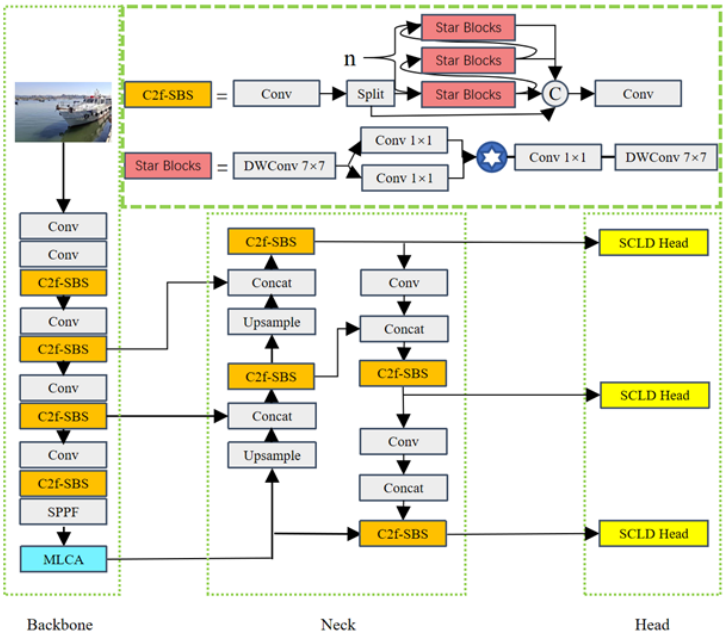


Figure 2. Schematic of LMS-YOLO integrating C2f-SBS, MLCA, and SCLD modules

This work introduces LMS-YOLO, an enhanced object detection framework built upon the YOLOv8n baseline, specifically optimized for detecting marine targets with improved accuracy and computational efficiency, as illustrated in Figure 2. To reduce model complexity while enhancing feature expressiveness, the standard C2f modules in both

the backbone and neck are replaced with a newly designed C2f-SBS module. This substitution simplifies the overall architecture, reduces the number of parameters and floating-point operations, and strengthens the model's capacity for multi-scale feature representation.

To further improve the network's ability to capture informative features in complex maritime environments, a MLCA module is embedded within the backbone. This lightweight attention mechanism enhances the extraction of context-aware features by integrating both local and global spatial-channel relationships.

In addition, the original detection head is replaced by a SCLD Head. This component allows feature maps from multiple detection scales to be processed using shared convolutional layers, effectively reducing parameter redundancy and computational load while preserving scale-specific representational quality.

3.1 C2f-SBS

In the YOLOv8 architecture, the C2f module employs multiple bottleneck layers to stabilize training by alleviating vanishing and exploding gradient problems, thereby supporting deeper network construction and improved convergence. Despite these advantages, the repeated stacking of bottlenecks imposes substantial computational overhead and lacks mechanisms for selectively enhancing multi-scale features—factors that hinder both efficiency and adaptability in complex detection tasks.

To overcome these limitations, this study introduces Star Blocks from the StarNet architecture to replace the dual-branch structure of the original C2f design, as illustrated in Figure 3. Developed by Microsoft in 2024, StarNet is a lightweight neural framework that utilizes a star-shaped computation strategy to effectively capture high-dimensional and nonlinear feature interactions. Importantly, it achieves this without increasing computational complexity, making it well suited for integration into lightweight detection networks.

StarNet employs a hierarchical design with depthwise separable convolutions and batch normalization (BN) to optimize feature extraction. Specifically, Eq. (1) illustrates that the input feature x is processed through a depthwise separable convolution to generate preliminary features F. This operation significantly reduces computational cost while maintaining effective feature representation. Eq. (2) further enhances the feature representation by applying convolution followed by BN f_C and an activation function ReLU. The final output y is combined with the preliminary feature x via a residual connection, ensuring continuity and stability in information propagation.

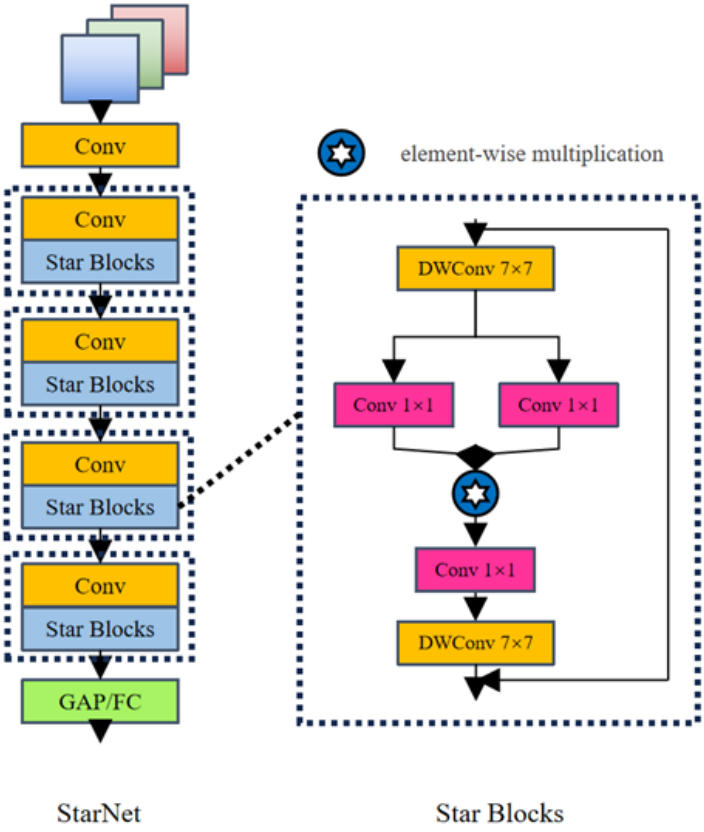


Figure 3. Structure of Star Blocks used in the C2f-SBS module for lightweight feature mapping

$$x = f_{DWC}(F) \tag{1}$$

$$y = x + f_C\left(\mathrm{ReLU}\left(f_C(x)\right) * f_C(x)\right) \tag{2}$$

In Eq. (1), F denotes the input feature map, which is a multidimensional array output from the previous network stage containing the current stage's extracted features. x represents the result of preliminary feature processing, encoding new features through initial nonlinear transformation and spatial-channel encoding, while f_DWC denotes the depthwise separable convolution function. In Eq. (2), y represents the final output feature map, x corresponds to the preliminary feature from Eq. (1), f_C denotes the convolution plus BN operation, ReLU is the rectified linear unit activation function, and * represents element-wise multiplication, where corresponding elements of two tensors with identical shapes are multiplied.

We propose the C2f-SBS module, derived from StarNet, as illustrated in Figure 4. Unlike traditional bottleneck designs that incur redundant computation due to dimension expansion and reduction, C2f-SBS replaces bottlenecks with Star Blocks, leveraging star-shaped operations to efficiently capture high-dimensional and nonlinear features in low-dimensional spaces while substantially reducing parameters. Importantly, this reduction in parameters does not compromise the extraction of marine object features; instead, it enhances the expressive power of the network.

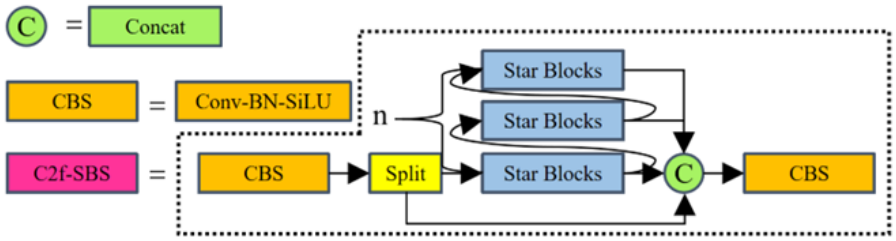


Figure 4. C2f-SBS module design used in LMS-YOLO for efficient multi-scale feature representation

The detailed structure of the C2f-SBS module is illustrated in Figure 4. First, the input data passes through a convolutional layer (Conv) to extract initial features, followed by a Split operation that divides the output features into two branches. In the main path, features are processed through multiple Star Blocks modules, performing layer-by-layer feature extraction and refinement. Simultaneously, the features in the secondary branch are concatenated (Concat) with the main path output at the end, enabling effective multi-scale feature fusion and enhancing the model's feature representation capability. Finally, the concatenated features are passed through a convolutional layer to generate the output feature map. Compared with the original neck convolutional structure, the C2f-SBS module significantly improves feature extraction efficiency while reducing computational complexity, demonstrating superior performance advantages.

3.2 MLCA

In complex and dynamic maritime environments, conventional object detection networks continue to encounter considerable challenges. Factors such as background clutter, wave-induced distortions, and lighting variations frequently contribute to missed detections or false positives—issues that are particularly pronounced when detecting small-scale targets.

To mitigate these effects, a lightweight MLCA module is integrated into the final stage of the backbone network. This module is designed to refine feature representation by simultaneously capturing channel-wise dependencies and spatial contextual information, while balancing both local and global feature interactions within the receptive field.

By reinforcing informative features and suppressing irrelevant background noise, the MLCA module significantly enhances the network's robustness in visually complex scenarios, leading to improved detection accuracy and reduced false detection rates—especially for small marine objects. The detailed structure of the MLCA module is illustrated in Figure 5.

The structure of the MLCA network is shown in Figure 5. The core of the MLCA module lies in fusing local and global features while integrating both channel and spatial information, thereby enhancing the model's attention to informative features. First, Eq. (3) extracts channel descriptors f_c from the feature map $F \in R^{C\times H\times W}$ using Global Average Pooling (GAP), where C denotes the number of channels, H and W represent the height and width of the feature map, respectively. Next, Eq. (4) employs a small feedforward network (comprising one or two fully

connected layers) to compute the channel attention weights M_c, which are mapped to the [0,1] range via the Sigmoid function to emphasize important channels and suppress irrelevant ones.

$$f_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_{cij} \tag{3}$$

$$M_c = \text{Sigmoid}\left(W_1 \cdot \text{ReLU}\left(W_0 f_c\right)\right) \tag{4}$$

Here, SigmoidandReLUdenote activation functions, W_0 and W_1 are trainable weight matrices.

The spatial attention mechanism processes the feature map via convolution (typically with a kernel) to generate a spatial attention map M_S, as shown in Eq. (5).

$$M_s = \text{Sigmoid}(\text{Conv}(F)) \tag{5}$$

Finally, Eq. (6) combines the channel attention M_c and spatial attention M_s with the original feature map F to obtain the enhanced feature map F'.

$$F' = F \times M_c \times M_s \tag{6}$$

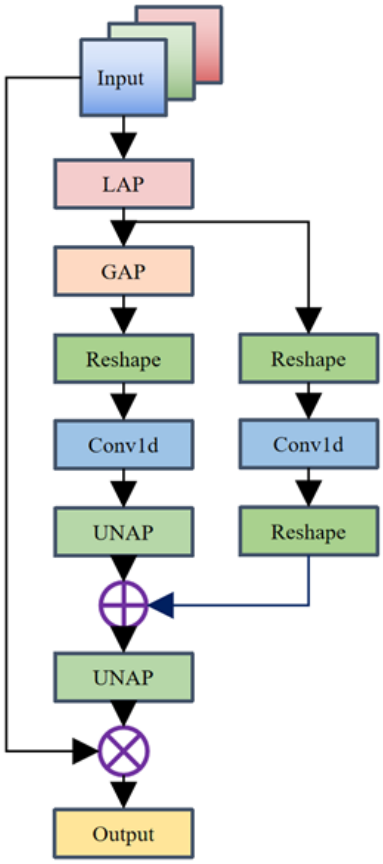


Figure 5. Architecture of the MLCA module for enhancing spatial and channel attention

By reinforcing informative features and suppressing noisy ones, the MLCA module enables the model to more accurately capture small target features in complex backgrounds, reducing both missed detections and false positives.

3.3 SCLD Head

To improve detection precision while maintaining a lightweight structure, YOLOv8 adopts a decoupled head, in which classification and bounding box regression are processed independently. Although this design enhances detection accuracy, it also introduces significant parameter redundancy, thereby limiting its applicability in computationally constrained environments.

To address this limitation, we propose the SCLD Head, which utilizes a unified convolutional structure shared across multiple detection scales. This approach significantly reduces the number of parameters and computation without sacrificing representational capacity. The architectural design of the SCLD module is illustrated in Figure 6.

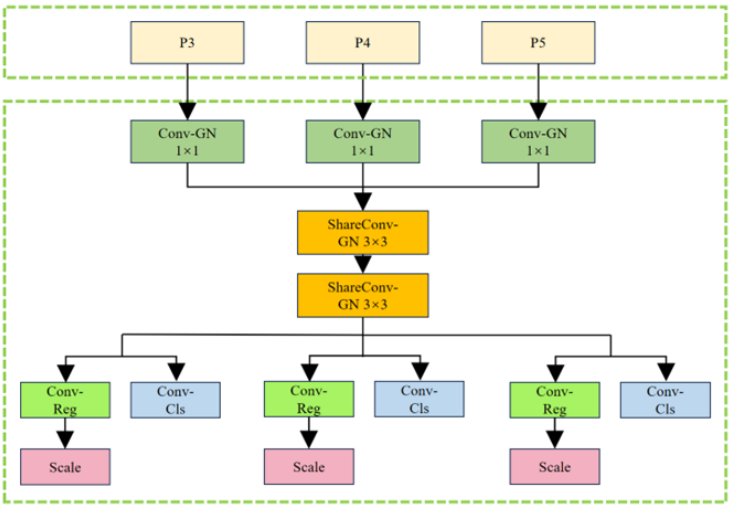


Figure 6. Architecture of the SCLD head

The SCLD head receives feature maps from three resolution stages—P3, P4, and P5—in the YOLOv8n network. To mitigate the potential drawback of shared kernels (i.e., reduced expressiveness at different scales), a scale-adaptive preprocessing step is introduced prior to the shared layers. Specifically, each scale undergoes a 1×1 convolution followed by Group Normalization (GN) [18], a process we refer to as differentiated channel compression. This ensures that each scale-specific feature map is appropriately normalized and dimensionally aligned before entering the shared convolution layers, thereby preserving essential scale-dependent characteristics.

Within the shared convolutional block, two 3×3 convolutional layers are applied sequentially. The first performs standard convolution to extract generic spatial features (e.g., edges and corners), while the second employs a dynamically adjustable dilation rate that adapts based on the input scale. This dual-convolution strategy enables multi-receptive field fusion, balancing the advantages of parameter sharing with the necessity for scale-specific spatial resolution.

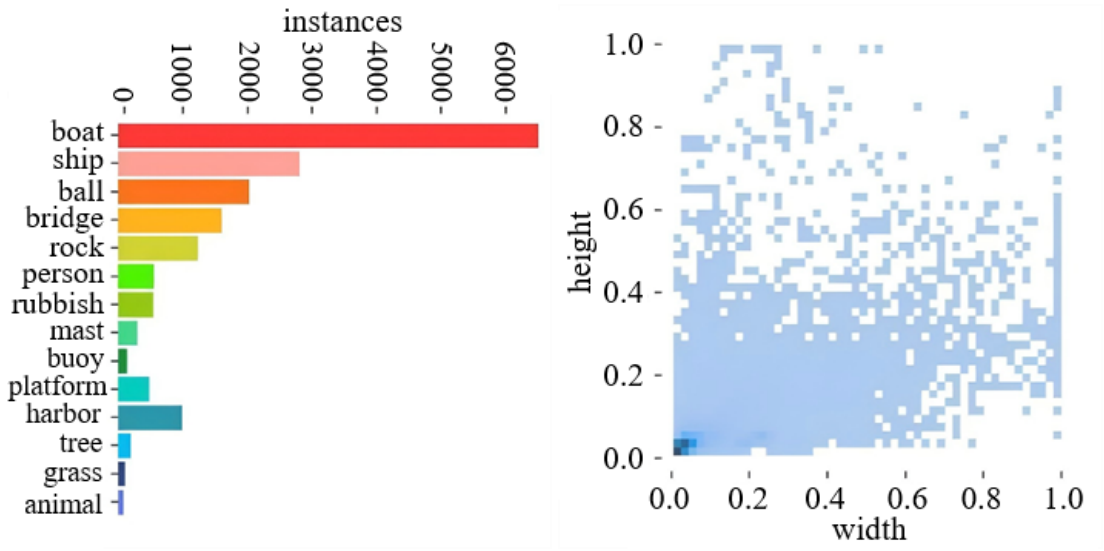
Following feature extraction, the network branches into two heads: one for classification (Conv_Cls) and one for regression (Conv_Reg). The regression output passes through a learnable Scale Layer, which adjusts each feature map element-wise to stabilize bounding box predictions. Meanwhile, the classification head incorporates a channel attention mechanism, allowing adaptive weight allocation based on category-specific relevance, thereby improving performance across varying object types and scales.

4 Experimental Results and Analysis

4.1 Dataset

The Water Surface Object Detection Dataset (WSODD), introduced by Zhou et al. [19], serves as a comprehensive benchmark featuring high-quality annotations and wide scene coverage. It includes imagery collected from five distinct water bodies—Yangtze River, Xuanwu Lake, Nanhaizi Lake, Yellow Sea, and Bohai Sea—with annotations spanning 14 surface object categories, as illustrated in Figure 7.

In total, WSODD comprises 7,467 images and 21,911 labeled instances, of which approximately 53% correspond to small objects. This high proportion of small-scale targets makes the dataset particularly suitable for evaluating lightweight detection models. The category distribution, visualized in Figure 7, reveals a notable class imbalance: categories such as “boat” and “ship” are well represented, while others, including “grass” and “animal”, contain relatively few samples.



(a) (b)

Figure 7. Category distribution of the WSODD dataset: (a) object types; (b) instance counts per class

4.2 Experimental Setup and Evaluation Metrics

To ensure experimental fairness and result reproducibility, all model training and evaluation were performed under a consistent hardware and software environment. Specifically, the experiments were conducted on a Windows 11 system equipped with an NVIDIA GeForce RTX 4060 Laptop GPU (8,188 MiB memory) and an AMD Ryzen 7 7840H CPU featuring 16 cores at 3.8 GHz with integrated Radeon 780M Graphics. The software environment consisted of Python 3.9.18 and the PyTorch 2.2.1 + cu118 deep learning framework. The YOLOv8n model was adopted as the baseline, with its training hyperparameters detailed in Table 1.

Table 1. Training configuration of the baseline YOLOv8n model

Training Parameters	Values
Learning Rate	0.01
Image Size	640*640
Momentum	0.937
Weight Decay	0.0005
Optimizer	SGD
Epochs	200
Batch Size	8

To comprehensively assess the performance of the proposed model in marine object detection tasks—particularly in USV scenarios—multiple evaluation metrics were employed. These include Precision (P), Recall (R), and mAP, which collectively reflect detection accuracy across various object categories. In addition, model complexity and inference efficiency were quantified using key indicators such as the number of parameters, GFLOPs (giga floating-point operations per second), model size, and frames per second (FPS) during inference.

The formal definitions of the primary evaluation metrics are provided in Eqs. (7)–(9). Specifically:

$$Precision = \frac{T_P}{(F_P + T_P)} \tag{7}$$

$$RcallScore = \frac{T_P}{T_P + F_N} \tag{8}$$

Here, T_P denotes the number of correctly detected marine targets, F_P denotes the number of falsely detected non-marine targets, and $\mathrm{F_N}$ denotes the number of missed marine targets.

$$mAP = \frac{1}{n}\sum_{i=1}^{i=n} AP_i \tag{9}$$

Moreover, n represents the total number of target categories in the dataset, i denotes the number of detection instances, and AP stands for the average precision of each category.

4.3 Ablation Study

To evaluate the individual contributions of the C2f-SBS module, MLCA module, and SCLD Head, we conducted a series of incremental ablation experiments based on the YOLOv8n baseline. The results are summarized in Table 2.

Table 2. Ablation results

Experiment	C2f-SBS	MLCA	SCLD	P (%)	R (%)	mAP@ 0.5 (%)	mAP@ 0.5:0.95 (%)	Parameters (M)	GFLOPs	Size (MB)	FPS
1	-	-	-	80%	70.9%	76%	44%	3.0	8.1	6.0	220.3
2	√	-	-	79.3%	69%	74%	44%	2.5	6.9	5.1	213.6
3	-	√	-	83%	69.6 %	77.1%	44.5%	3.0	8.1	6.0	217.2
4	-	-	√	84.1%	70.9 %	76.9%	45.2%	2.3	6.5	4.7	220.1
5	√	√	√	85.5%	69.6%	78.3%	45.7%	1.9	5.3	3.8	208.4

• Experiment 1 establishes the baseline performance using unmodified YOLOv8n.

• Experiments 2 to 4 assess the isolated impact of integrating the C2f-SBS module, the MLCA module, and the SCLD head, respectively.

• Experiment 5 reports the final performance of the full LMS-YOLO model, incorporating all proposed modules.

In Experiment 2, replacing the original C2f structure with the C2f-SBS module leads to a marginal decrease in Precision (P), FPS, and mAP@0.5, yet significantly reduces model size and computational demands—parameters decrease by 15.73%, GFLOPs by 14.81%, and model size drops from 6.0 MB to 5.1 MB. These results suggest that the C2f-SBS module maintains competitive detection performance while improving the model's deployability, especially for embedded or resource-constrained environments.

Experiment 3 shows that integrating the MLCA module into the backbone yields noticeable performance gains: Precision improves by 3%, while mAP@0.5 and mAP@0.5:0.95 increase by 1.1% and 0.5%, respectively. Importantly, this improvement is achieved with negligible increases in parameters and computation. The performance gain can be attributed to MLCA's ability to effectively capture local and global contextual dependencies across spatial and channel dimensions, thus enhancing feature representation. This module proves particularly beneficial in cluttered maritime environments.

In Experiment 4, replacing the original head with the proposed SCLD design results in a 4.1% improvement in Precision and a 1.2% increase in mAP@0.5:0.95, while simultaneously reducing parameter count by 21.44%, GFLOPs by 19.75%, and model size from 6.0 MB to 4.7 MB. The FPS remains stable at around 220, demonstrating that the shared-convolution strategy effectively reduces complexity without compromising inference speed or accuracy.

Experiment 5 evaluates the complete LMS-YOLO model. Compared with the YOLOv8n baseline, LMS-YOLO achieves a 5.5% improvement in Precision, and gains of 2.3% in mAP@0.5 and 1.7% in mAP@0.5:0.95. Moreover, the model achieves a 37.18% reduction in parameters and 34.57% in GFLOPs, with the model size compressed to 3.8 MB. Although the FPS drops slightly (5.4% decrease), the inference speed remains well above real-time requirements for marine object detection. These results highlight LMS-YOLO's ability to effectively balance detection accuracy and computational efficiency, validating its suitability for real-world deployment on low-power platforms.

4.4 Comparative Experiments and Visualization

To further evaluate the detection performance of LMS-YOLO across different categories of marine targets, we conducted comparative experiments using the same hardware configuration and dataset described in Section 4.2. The proposed model was benchmarked against several state-of-the-art detection algorithms, including lightweight variants such as YOLOv5n, YOLOv9t, and YOLOv10n, as well as heavier models like YOLOv7 and YOLOv9c. The experimental results are summarized in Table 3, with the top two values in each evaluation metric highlighted in bold.

While YOLOv9c marginally outperforms LMS-YOLO in terms of Precision and mAP@0.5:0.95, it comes with a significantly larger number of parameters and higher computational cost. Among all lightweight models,

LMS-YOLO achieves the best overall performance. Compared with YOLOv5n, LMS-YOLO shows improvements of 4.2% in Precision, 2.7% in mAP@0.5:0.95, and an additional 50 FPS, albeit with a slight increase in model complexity. Against YOLOv9t and YOLOv10n, LMS-YOLO consistently surpasses them across all evaluation metrics, demonstrating superior performance and better balance between accuracy and efficiency.

These findings confirm that LMS-YOLO not only enhances detection accuracy in marine environments but also maintains an effective trade-off between model compactness and real-time processing capability, making it suitable for deployment in practical maritime applications.

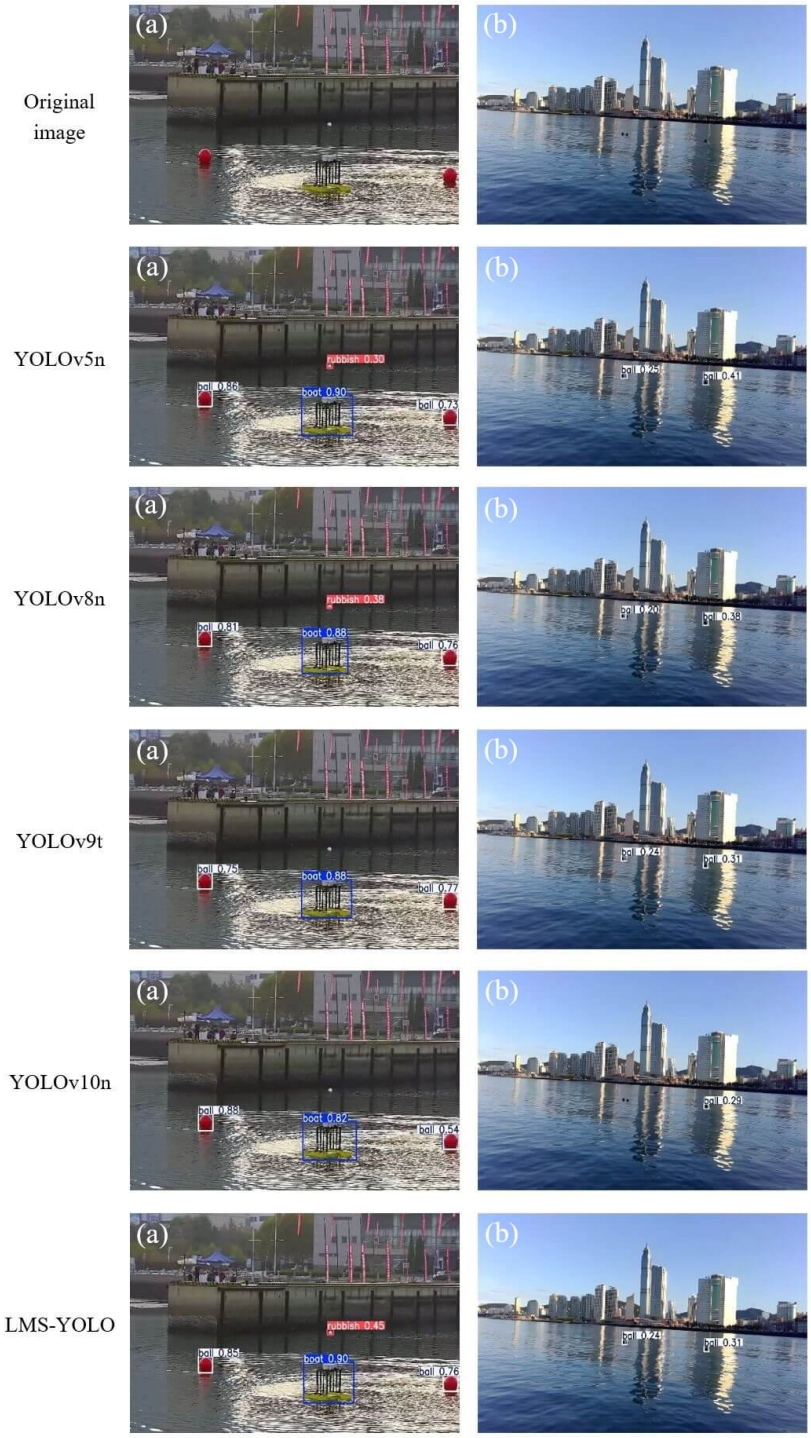


Figure 8. Qualitative comparison of detection results under complex maritime conditions: (a) sunlight reflection; (b) surface reflection

To provide an intuitive understanding of LMS-YOLO's detection robustness under visually complex maritime

conditions, we further conducted visualization experiments using two representative background scenarios:

(a) subtle surface ripples combined with sunlight reflection interference

(b) strong water surface reflection interference

The models selected for comparison include YOLOv5n, YOLOv8n, YOLOv9t, and YOLOv10n, with the results visualized in Figure 8.

Table 3. Performance comparison between LMS-YOLO and mainstream detection models

Model	P (%)	mAP@0.5:0.95 (%)	Parameters (M)	GFLOPs	Fps
YOLOv5n	81.3%	43%	1.7	4.2	78
YOLOv7	85.1	45.5%	37.2	105.2	43
YOLOv8n	80%	44%	3.0	8.1	145
YOLOv9t	80.4%	41.5%	2.6	10.7	56
YOLOv9c	86.9%	46.5%	60.5	264	37
YOLOv10n	75%	39.6%	2.7	8.3	111
LMS-YOLO	85.5%	45.7%	1.9	5.3	128

In scenario (a), LMS-YOLO successfully detected all surface targets, while YOLOv9t and YOLOv10n failed to identify distant "rubbish" objects. In scenario (b), LMS-YOLO again demonstrated superior robustness, detecting all "ball" targets without omission. By contrast, YOLOv5n, YOLOv8n, and YOLOv9t each missed one "ball" target, and YOLOv10n failed to detect two such targets.

These visual results underscore LMS-YOLO's ability to maintain stable and accurate detection performance even under challenging environmental conditions such as specular reflections and dynamic surface disturbances. Its effectiveness in detecting small-scale marine objects within noisy, real-world scenarios highlights its strong generalization ability and practical value.

4.5 Hardware Deployment Experiments

To evaluate the deployment feasibility of the proposed LMS-YOLO model across various computing environments, we conducted inference performance tests on three different hardware platforms:

- an AMD Ryzen 7 7840H CPU (16 cores),
- an NVIDIA GTX 1060 GPU, and
- an NVIDIA RTX 4060 GPU.

The evaluation focused on several key indicators: frames per second, model size, parameter count, and computational cost measured in GFLOPs. The goal was to assess the model's adaptability to resource-constrained edge environments, particularly in the context of real-time marine object detection. The results are summarized in Table 4.

Table 4. Inference performance of LMS-YOLO on different hardware platforms

Hardware Environment	Models	FPS	Size (MB)	Parameters (M)	GFLOPs
AMD Ryzen 7 7840H	YOLOv8n	25.1	6.0	3.0	8.1
	LMS-YOLO	22.2	3.8	1.9	5.3
NVIDIA GTX 1060	YOLOv8n	57.4	6.0	3.0	8.1
	LMS-YOLO	52.2	3.8	1.9	5.3
NVIDIA RTX 4060	YOLOv8n	220.3	6.0	3.0	8.1
	LMS-YOLO	208.4	3.8	1.9	5.3

Although LMS-YOLO exhibits a modest reduction in FPS compared to non-lightweight models, it achieves substantial improvements in deployment efficiency. Specifically:

- The smaller model size reduces both storage and transmission demands;
- The lower parameter count minimizes memory consumption;
- The reduced GFLOPs effectively ease the computational load during inference.

These characteristics collectively enhance LMS-YOLO's compatibility with edge devices, such as embedded systems or mobile marine platforms, where real-time processing, power efficiency, and storage constraints are critical factors. Overall, the experimental results validate LMS-YOLO's deployment flexibility and practical viability across diverse hardware configurations.

4.6 Visualization Analysis

To provide an intuitive assessment of LMS-YOLO's detection performance, we conducted extensive qualitative experiments under a range of challenging environmental conditions, including variations in scene context, illumination levels, and weather phenomena. A comparative analysis with the baseline YOLOv8n model is shown in Figure 9, illustrating LMS-YOLO's consistent ability to accurately detect water surface objects across diverse testing scenarios.

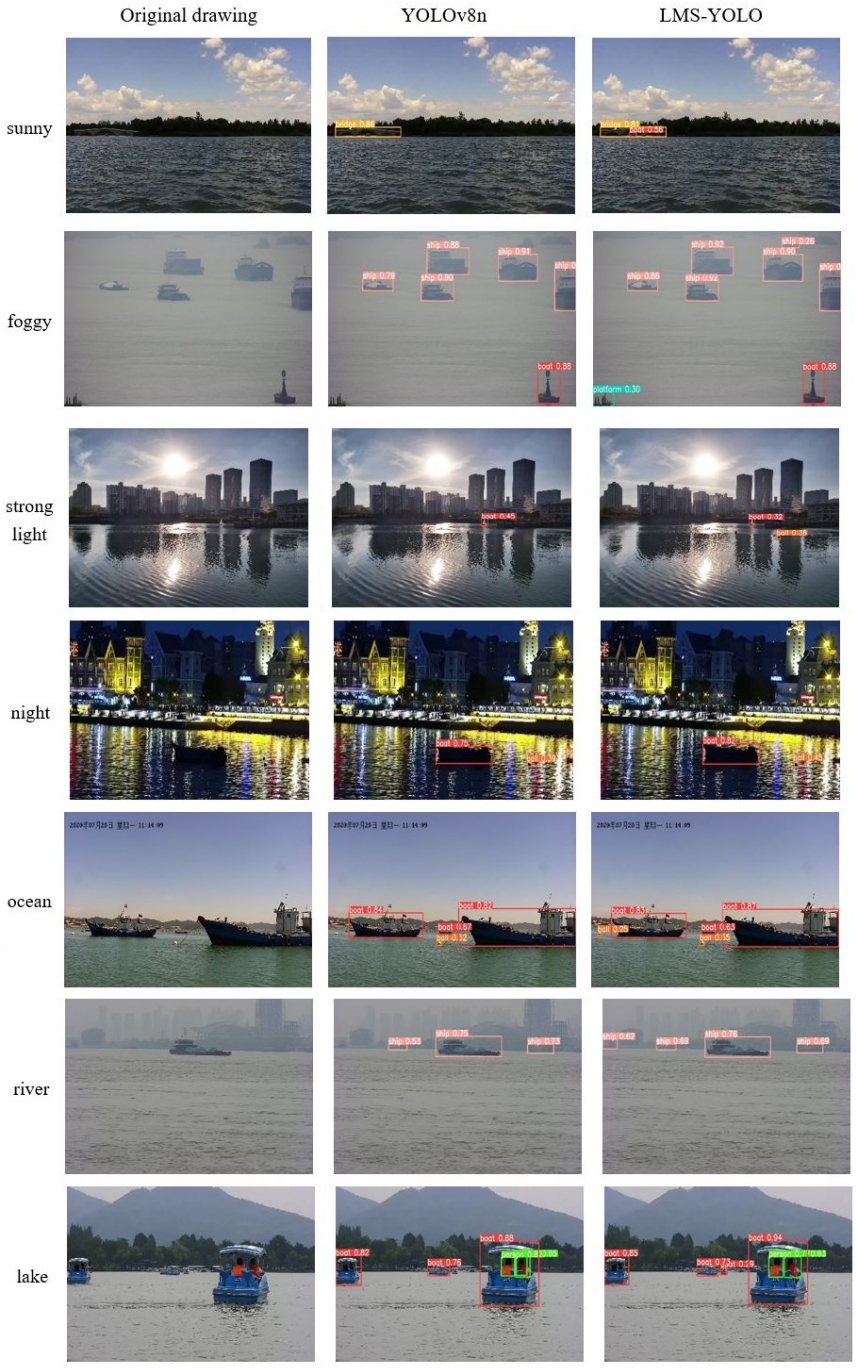


Figure 9. Visual comparison of detection results between YOLOv8n and LMS-YOLO across diverse environmental settings

LMS-YOLO successfully detects targets that are missed by the baseline model in a variety of conditions—for instance, identifying "boat" in sunny environments, "ship" and "platform" under fog, and "ball" under both high-intensity illumination and marine wave interference. It also demonstrates strong performance across different water body types, such as rivers and lakes. Moreover, LMS-YOLO significantly reduces false positives: under nighttime conditions, the baseline model erroneously classifies background features as "ball", whereas LMS-YOLO correctly

suppresses such misclassifications.

These findings confirm LMS-YOLO's robustness in complex environments—including oceans, lakes, rivers, low-light scenes, high glare, and foggy conditions—highlighting its suitability for practical deployment and real-world maritime applications.

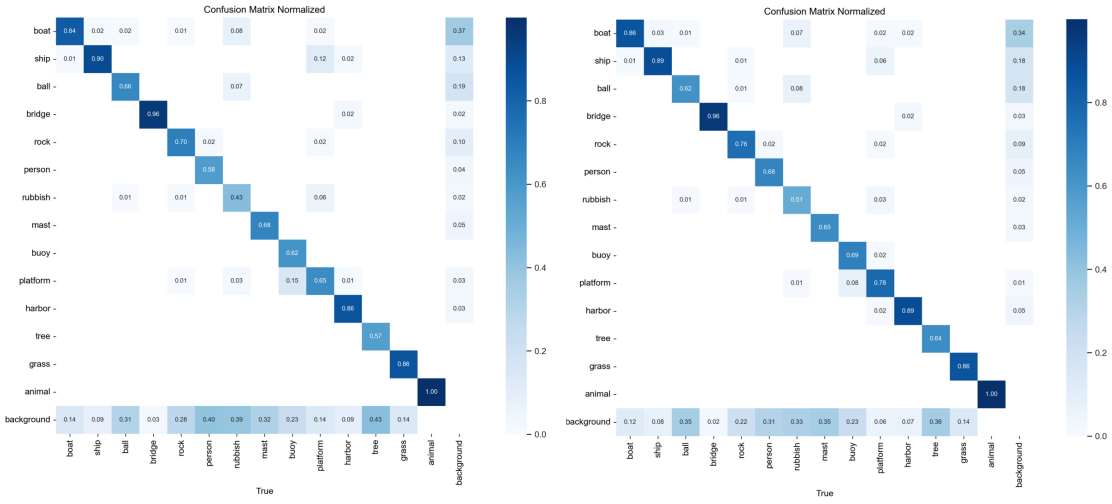
Despite these strengths, certain extreme environments still challenge LMS-YOLO's detection capability. For instance, under intense glare or very low light, several "ball" targets were missed. These limitations are mainly attributed to:

(1) Reduced visual contrast caused by wave motion and high-intensity lighting, which impairs target-background separability;

(2) Low signal-to-noise ratio in nighttime settings, particularly affecting small object detection.

To overcome these issues, future enhancements may focus on integrating illumination-adaptive modules, improving low-light feature extraction, and enhancing robustness to reflection artifacts, thereby enabling more reliable performance under extreme lighting conditions.

To complement the qualitative analysis, we further compared LMS-YOLO and YOLOv8n using a confusion matrix, as shown in Figure 10. In this matrix, true labels are plotted on the horizontal axis and predicted labels on the vertical axis. The diagonal elements represent true positives (TP), while values in the lower-left triangle correspond to false negatives (FN)—missed or misclassified targets—and the upper-right triangle denotes false positives (FP), where background or incorrect objects were incorrectly identified as targets.



(a) (b)

Figure 10. Confusion matrix comparing YOLOv8n and LMS-YOLO predictions across all object categories

A comparison of the diagonal values reveals that LMS-YOLO achieves higher classification accuracy in most categories. For example, detection accuracy for the "platform" class improves from 0.65 to 0.78. Additionally, LMS-YOLO reduces both FN and FP values across most categories, indicating enhanced reliability, improved recall, and lower false detection rates.

4.7 Generalization Experiments

Table 5. Performance comparison between LMS-YOLO and mainstream detection models

Model	P (%)	R (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	Parameters (M)	GFLOPs
YOLOv8n	0.883	0.833	0.879	0.498	1.9	5.3
LMS-YOLO	0.861	0.849	0.891	0.481	3.0	8.1

To comprehensively evaluate the generalization capability of the proposed LMS-YOLO algorithm, we conducted experiments on the FloW-Img dataset [20]. This dataset comprises high-resolution imagery specifically curated for detecting floating debris in real inland water environments, captured from unmanned surface vessels under varied perspectives. It includes 2,000 images with 5,271 finely annotated targets, the majority of which are small objects (less than 32 × 32 pixels)—making it particularly relevant to the objectives of this study.

As shown in Table 5 and Figure 11, LMS-YOLO demonstrates strong generalization performance on FloW-Img when compared to the baseline YOLOv8n model. LMS-YOLO achieves a Precision (P) of 0.883 and Recall (R) of 0.833, while YOLOv8n records 0.861 and 0.849, respectively. This reflects a 2.2% improvement in precision, suggesting more effective reduction of false positives, albeit with a slight decrease in recall.

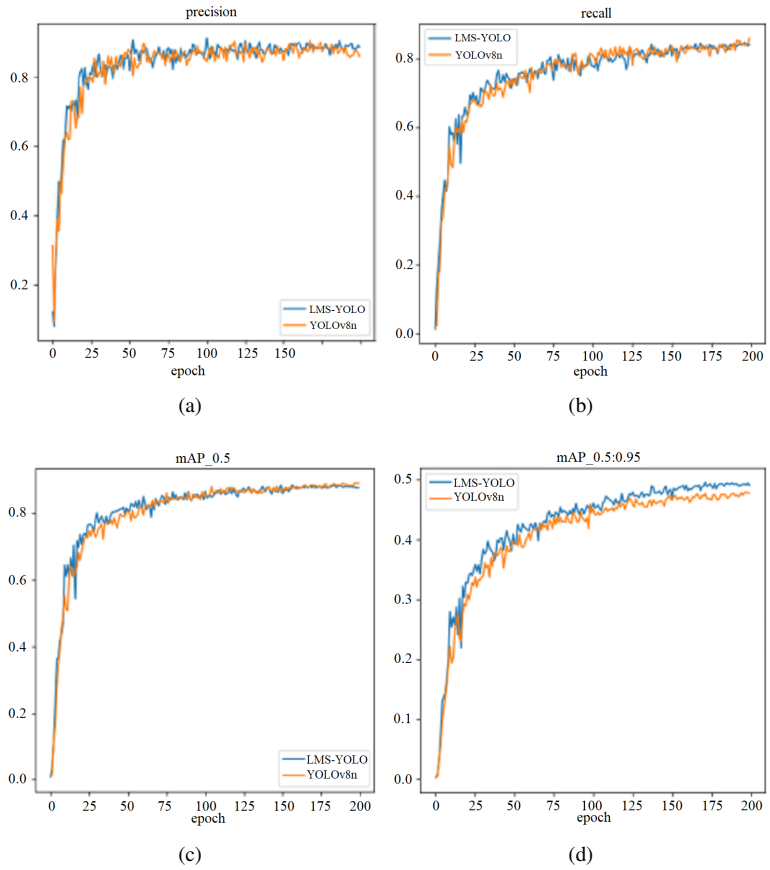


Figure 11. Comparison of key metrics before and after model enhancement on FloW-Img

In terms of mAP metrics, LMS-YOLO obtains 0.879 at mAP@0.5 and 0.498 at mAP@0.5:0.95, whereas YOLOv8n achieves 0.891 and 0.481, respectively. Although LMS-YOLO performs slightly lower at the relaxed IoU threshold (0.5), it clearly outperforms the baseline at the stricter threshold (0.5:0.95), which is more sensitive to boundary localization quality—a critical factor in small object detection.

Figure 12. Visual detection results of LMS-YOLO on floating debris in real inland waters

The improvements in generalization can be attributed to three key architectural enhancements:

• The C2f-SBS module, which improves multi-scale feature representation and fusion;

• The MLCA module, which combines local and global contextual cues, thereby enhancing the network's robustness to background interference;

• The SCLD head, which enables efficient scale-aware detection with reduced model complexity.

These components collectively enhance detection accuracy while maintaining a lightweight architecture, making LMS-YOLO well suited for practical deployment.

As illustrated in Figure 12, LMS-YOLO successfully detects all floating debris targets in real-world images from FloW-Img, confirming its generalization ability and robustness across unseen environments.

5 Conclusions

This study presents LMS-YOLO, a lightweight object detection algorithm tailored for complex water surface environments, addressing the limitations of low detection accuracy and high computational complexity found in existing methods. The model introduces several architectural enhancements:

• a C2f-SBS module incorporating Star Blocks to improve multi-scale feature fusion,

• a shared convolutional detection head to minimize parameter count and computational load, and

• an MLCA module to enhance contextual feature utilization within challenging visual scenarios.

Comprehensive experiments demonstrate that LMS-YOLO achieves a 5.5% improvement in precision, a 2.3% gain in mAP@0.5, and a 1.7% increase in mAP@0.5:0.95, while simultaneously reducing the parameter count by 37.18% and computational cost by 34.57% compared to the YOLOv8n baseline. The algorithm also exhibits strong generalization capability on unseen datasets and maintains stable performance across diverse environmental conditions, including reflections, fog, and varying lighting.

Overall, LMS-YOLO effectively balances detection performance and model efficiency, making it well suited for deployment in real-time applications, particularly on resource-constrained platforms such as USVs. Future work will explore the integration of illumination-adaptive modules and multi-modal fusion strategies to further enhance robustness under extreme visual disturbances.

Author Contribution

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

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] M. J. Karim, M. Nahiduzzaman, M. Ahsan, and J. Haider, "Development of an early detection and automatic targeting system for cotton weeds using an improved lightweight YOLOv8 architecture on an edge device," *Knowl.-Based Syst.*, vol. 300, p. 112204, 2024. https://doi.org/10.1016/j.knosys.2024.112204

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[3] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint, arXiv:1804.02767*, 2018. https://doi.org/10.48550/arXiv.1804.02767

[4] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint, arXiv:2004.10934*, 2020. https://doi.org/10.48550/arXiv.2004.10934

[5] G. Jocher, A. Stoken, A. Chaurasia, J. Borovec, NanoCode012, T. Xie, Y. Kwon, K. Michael, C. Y. L., J. Fang, and et al., "v6.0-YOLOv5n 'Nano' models, roboflow integration, tensorflow export, opencv dnn support," Zenodo, 2021. https://doi.org/10.5281/zenodo.5563715

[6] C. Y. Li, L. L. Li, H. L. Jiang, K. H. Weng, Y. F. Geng, L. Li, Z. D. Ke, Q. Y. Li, M. Cheng, W. Q. Nie, and et al., "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv preprint, arXiv:2209.02976*, 2022. https://doi.org/10.48550/arXiv.2209.02976

[7] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475. https://doi.org/10.1109/CVPR52733.2023.00721

[8] Z. X. Huang, X. N. Jiang, F. L. Wu, Y. Fu, Y. Zhang, T. J. Fu, and J. Y. Pei, "An improved method for ship target detection based on YOLOv4," *Appl. Sci.*, vol. 13, no. 3, p. 1302, 2023. https://doi.org/10.3390/app13031302

[9] X. P. Zhang, Z. Y. Xu, S. Qu, W. Qiu, and Z. Y. Zhai, "Recognition algorithm of marine ship based on improved YOLOv5 deep learning."

[10] Z. K. Jiang, L. Su, and Y. X. Sun, "YOLOv7-ship: A lightweight algorithm for ship object detection in complex marine environments," *J. Mar. Sci. Eng.*, vol. 12, no. 1, p. 190, 2024. https://doi.org/10.3390/jmse12010190

[11] C. L. Li, L. Wang, Y. T. Liu, and S. K. Zhang, "Lightweight water surface object detection network for unmanned surface vehicles," *Electronics*, vol. 13, no. 15, p. 3089, 2024. https://doi.org/10.3390/electronics13153089

[12] J. Wang and H. Zhao, "Improved YOLOv8 algorithm for water surface object detection," *Sensors*, vol. 24, no. 15, p. 5059, 2024. https://doi.org/10.3390/s24155059

[13] B. B. Chen, F. Ding, B. J. Ma, L. Q. Wang, and S. P. Ning, "A method for real-time recognition of safflower filaments in unstructured environments using the YOLO-SaFi model," *Sensors*, vol. 24, no. 13, p. 4410, 2024. https://doi.org/10.3390/s24134410

[14] Q. M. Cheng, H. Y. Huang, Y. Xu, Y. Z. Zhou, H. Y. Li, and Z. Y. Wang, "NWPU-captions dataset and MLCA-net for remote sensing image captioning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–19, 2022. https://doi.org/10.1109/TGRS.2022.3166979

[15] C. J. Feng, Y. J. Zhong, Y. Gao, M. R. Scott, and W. L. Huang, "Tood: Task-aligned one-stage object detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3490–3499. https://doi.org/10.1109/ICCV48922.2021.00349

[16] X. Li, W. H. Wang, L. J. Wu, S. Chen, X. L. Hu, J. Li, and J. H. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 002–21 012.

[17] L. Y. Du and Y. S. Wang, "Bi-YOLO: A novel object detection network and dataset for components of China heritage buildings," *J. Build. Eng.*, vol. 97, p. 110817, 2024. https://doi.org/10.1016/j.jobe.2024.110817

[18] D. W. Yi, H. B. Ahmedov, S. Y. Jiang, Y. R. Li, S. J. Flinn, and P. G. Fernandes, "Coordinate-aware Mask R-CNN with group normalization: An underwater marine animal instance segmentation framework," *Neurocomputing*, vol. 583, p. 127488, 2024. https://doi.org/10.1016/j.neucom.2024.127488

[19] Z. G. Zhou, J. E. Sun, J. B. Yu, K. Y. Liu, J. W. Duan, L. Chen, and C. L. P. Chen, "An image-based benchmark dataset and a novel object detector for water surface object detection," *Front. Neurorobot.*, vol. 15, p. 723336, 2021. https://doi.org/10.3389/fnbot.2021.723336

[20] Y. W. Cheng, J. N. Zhu, M. X. Jiang, J. Fu, C. S. Pang, P. Wang, K. Sankaran, O. Onabola, Y. M. Liu, D. B. Liu, and Y. Bengio, "Flow: A dataset and benchmark for floating waste detection in inland waters," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 953–10 962. https://doi.org/10.1109/ICCV48922.2021.01079