Journal of
Marine Science and Engineering

MDPI

Article

Side-Scan Sonar Small Objects Detection Based on Improved YOLOv11

Chang Zou [1,2,3], Siquan Yu [1,2,*], Yankai Yu [1,2], Haitao Gu [1,2] and Xinlin Xu [3,4]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; zouchang@sia.cn (C.Z.)
[2] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
[4] Institute of Deep-Sea Science and Engineering, Chinese Academy of Sciences, Sanya 572000, China
* Correspondence: yusiquan@sia.cn

Abstract: Underwater object detection using side-scan sonar (SSS) remains a significant challenge in marine exploration, especially for small objects. Conventional methods for small object detection face various obstacles, such as difficulties in feature extraction and the considerable impact of noise on detection accuracy. To address these issues, this study proposes an improved YOLOv11 network named YOLOv11-SDC. Specifically,a new Sparse Feature (SF) module is proposed, replacing the Spatial Pyramid Pooling Fast (SPPF) module from the original YOLOv11 architecture to enhance object feature selection. Furthermore, the proposed YOLOv11-SDC integrates a Dilated Reparam Block (DRB) with a C3k2 module to broaden the model's receptive field. A Content-Guided Attention Fusion (CGAF) module is also incorporated prior to the detection module to assign appropriate weights to various feature maps, thereby emphasizing the relevant object information. Experimental results clearly demonstrate the superiority of YOLOv11-SDC over several iterations of YOLO versions in detection performance. The proposed method was validated through extensive real-world experiments, yielding a precision of 0.934, recall of 0.698, mAP@0.5 of 0.825, and mAP@0.5:0.95 of 0.598. In conclusion, the improved YOLOv11-SDC offers a promising solution for detecting small objects in SSS images, showing substantial potential for marine applications.

Keywords: underwater small object detection; computer vision; YOLOv11

Academic Editor: Weicheng Cui

Received: 3 January 2025
Revised: 15 January 2025
Accepted: 15 January 2025
Published: 18 January 2025

Citation: Zou, C.; Yu, S.; Yu, Y.; Gu, H.; Xu, X. Side-Scan Sonar Small ObjectDetection Based on Improved YOLOv11. *J. Mar. Sci. Eng.* **2025**, *13*, 162. https://doi.org/10.3390/jmse13010162

1. Introduction

The ocean, which covers most of the Earth's surface, plays a vital role in numerous fields, including marine research, resource exploration, and military operations [1]. Detecting small underwater objects has become a crucial task with various applications, such as marine archaeological exploration, environmental monitoring, and underwater defense systems [2–4]. Small objects often have weak acoustic signatures, making them susceptible to being masked by noise or complex seabed structures, presenting significant challenges for detection systems. Accurate and efficient detection of these objects is essential for ensuring maritime safety, advancing scientific research, and supporting underwater operations in complex environments.

Side-scan sonar (SSS) has emerged as an ideal solution for large-scale marine detection due to its wide coverage, high resolution, and reliable imaging capabilities [5–9]. Conventional methods for detecting small objects typically rely on the manual interpretation of sonar images to identify objects. This approach is not only time-consuming but also highly dependent on the operator's expertise, resulting in inconsistent outcomes and limited scalability. Moreover, the complex underwater environment, characterized by noise, clutter, and varying seabed conditions, presents considerable challenges to manual detection.

To address these challenges, the integration of SSS imaging with advanced automated detection techniques presents a promising approach to enhancing the precision and efficiency of small object detection. Detection is typically carried out using sonar systems mounted on the hull of a ship or an underwater vehicle [10]. Sonar images produced by SSS provide detailed morphological data of underwater objects. When combined with sophisticated object detection algorithms, SSS images enable high-accuracy, automated object detection. As a result, research into autonomous small object detection using SSS imagery has attracted increasing interest in recent years.

For an extended period, research on autonomous SSS object detection has primarily focused on conventional methods [11,12]. In two-dimensional sonar images, objects generate prominent backscatter echoes, which appear as bright regions in the sonar image. Additionally, the obstruction of objects causes shadow regions behind them, where sound waves are blocked. The interaction between bright regions and shadows is a conjugate phenomenon, where the position of the shadow is geometrically related to both the bright region and the sonar's height [13–15]. This feature can be exploited for effective object detection.

Traditional underwater small object detection algorithms typically involve the stages of feature extraction followed by classification [16]. Initially, regions of interest are extracted from sonar images, isolating the object areas. Next, the segmented regions are used to extract features. Finally, a classifier categorizes and identifies these extracted features.

Lopera et al. [17] first applied anisotropic diffusion filtering to reduce noise, then used morphological methods to segment bright and shadow regions. They subsequently applied fuzzy morphological techniques to refine the segmentation. By combining over 30 features, including area, contours, and circumscribed ellipses, they used a Markov Chain Monte Carlo algorithm to identify mine-like objects in both real and simulated datasets. However, the Markov Chain Monte Carlo algorithm, while effective, is computationally intensive and not suitable for real-time applications.

Grasso and Spina [18] proposed a non-parametric detection method based on mathematical morphology to detect and estimate the density of small bottom objects in side-scan sonar images. They addressed the challenge of noise interference across various seabed types by designing a system that employs nonlinear filtering to estimate the seabed signal envelope. This approach enabled the segmentation of bright and shadow regions in high-resolution images. To reduce false alarms caused by seabed disturbances, the method considers the spatial proximity of bright and shadow regions as a necessary condition for object presence. However, while the approach demonstrates robustness to seabed disturbances, it requires manual adjustment of the window size of the morphological operation and shadow distance threshold for different seabed types, highlighting its lack of adaptability.

Over the past decade, the adoption of deep learning in this field of research has steadily increased [19–22]. Yamada et al. [23] extracted canonical correlation features for buried objects using a dual-channel standard coordinate decomposition method. They applied various multi-aspect decision-level fusion techniques and employed a backpropagation neural network to classify mines and mine-like objects among non-mine objects. However, although multi-aspect decision-level fusion techniques can improve classification accuracy, their reliance on images from multiple viewpoints reduces the applicability of the method.

Zhu et al. [24] proposed an automated object detection method based on deep learning in underwater SSS images. The study integrates a pre-trained convolutional neural network with a support vector machine to achieve feature extraction and classification of sonar objects. Initially, SSS images are preprocessed using downsampling and histogram equalization. During the detection stage, a matched filter is applied to segment the bright and shadow regions of potential objects, identifying candidate regions. In the classification stage, AlexNet is employed to extract high-dimensional feature vectors from each candidate region. These features are then classified as objects and non-objects using a linear SVM. The matched filter design assumes that objects exhibit distinct bright and shadow regions with a fixed spatial relationship. However, under complex conditions such as non-uniform seabeds or weakly reflective objects, this assumption leads to missed detections.

To address the recurring problem of low precision and recall in small object detection tasks using SSS, this study aims to improve the YOLOv11 model, proposing an enhanced version termed YOLOv11-SDC. Experiments conducted on the SIMD dataset validate the effectiveness of the proposed model. The primary contributions of this study are outlined as follows:

1. A new Sparse Feature (SF) module is proposed, which utilizes feature importance ranking and channel pruning to better focus on key object features in SSS images.
2. A Dilated Reparam Block (DRB) module [25] is introduced and integrated with the newly developed C3k2 module in YOLOv11. This combination expands the receptive field and enhances the model's ability to capture object features.
3. A Content-Guided Attention Fusion (CGAF) module [26] is incorporated to further improve detection performance through multi-scale feature fusion and weighting.

2. Related Works

Numerous researchers have investigated enhancements in underwater small object detection using YOLO-based methods. Fu et al. [27] introduced an improved YOLOv5 model for small objects detection in SSS images. They addressed high rates of missed detections and false alarms by re-clustering anchor boxes using K-means, adding a specialized detection layer for shallow features, and incorporating SE and CBAM attention mechanisms to improve feature extraction. However, adding a dedicated shallow feature detection layer also introduces additional noise, making the model's effectiveness more dependent on the quality of the dataset.

Zhang et al. [28] proposed an enhanced YOLOv7-based approach for detecting small objects in side-scan sonar images. Their method integrates a dedicated detection layer, dual attention mechanisms, and a BiFPN structure for feature recombination. However, the absence of dynamic feature selection limits its ability to emphasize subtle and weak features of small objects, reducing its effectiveness in capturing fine details.

Tang et al. [29] developed an improved underwater object detection model based on YOLOv8 for SSS images. Their approach incorporates three key innovations: a shallow robust feature downsampling module for optimizing shallow feature maps, receptive field convolution to maintain semantic information during downsampling, and a Dysample module for enhancing feature fusion accuracy in the Feature Pyramid Network (FPN). Nevertheless, RFCAConv does not prioritize feature importance at the channel level, which results in critical object details being missed.

Santos et al. [30] introduced an SSS mine dataset containing numerous small underwater objects. The authors applied YOLOv4 to this dataset, achieving 82% precision, 64% recall, and 75% mAP in their experiments. In comparison, YOLOv11 is a newer version of the object detection network. Rahima et al. [31] proposed YOLOv11, making architectural improvements to the YOLO framework and achieving better object detection performance. As illustrated in Figure 1, YOLOv11 builds upon YOLOv8 [32], incorporating the C2PSA and C3k2 modules while maintaining the SPPF.

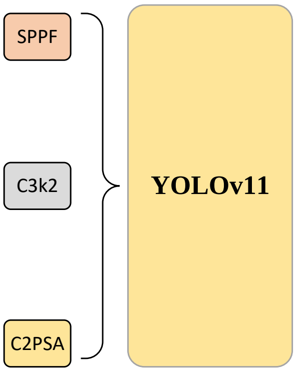


Figure 1. Main modules of YOLOv11.

As illustrated in Figure 2, YOLOv11 is organized into four key modules: input, backbone, neck, and head modules.

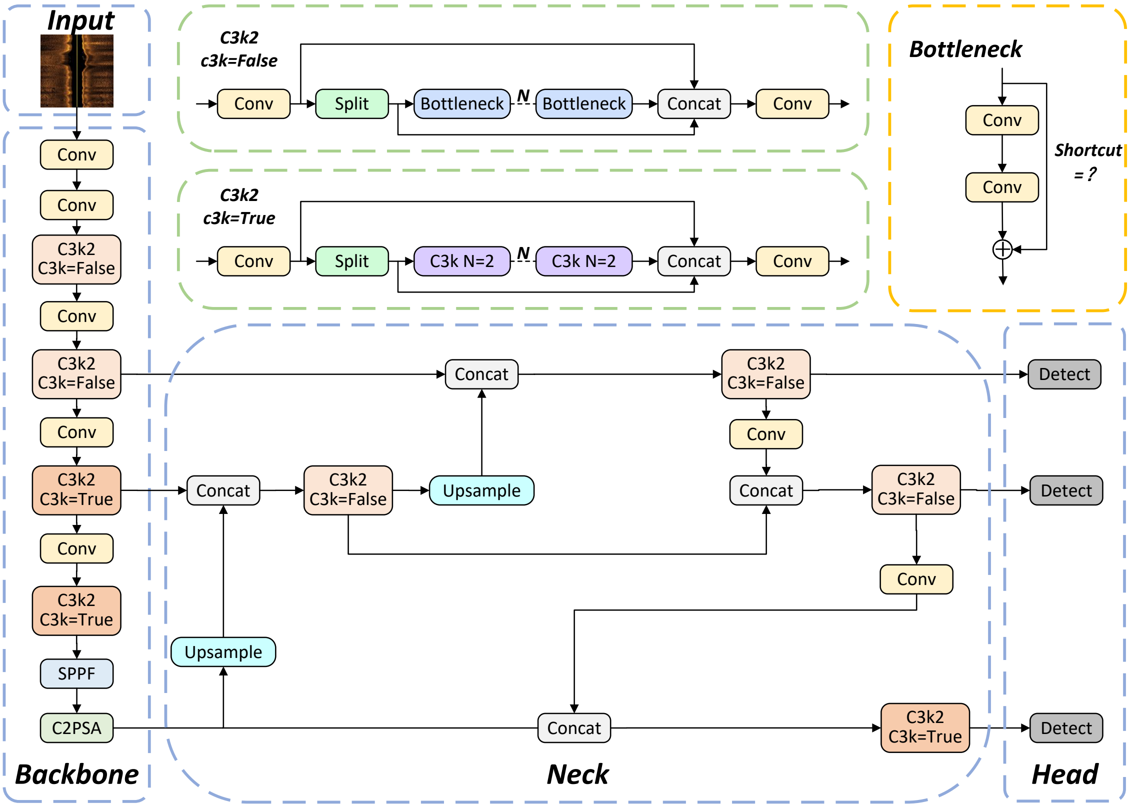


Figure 2. The network structure of YOLOv11.

The input module of YOLOv11 builds upon the adaptive anchor computation technique from YOLOv8 while integrating advanced augmentation methods such as mosaic and mixup. By adopting an anchor-free framework, YOLOv11 eliminates the need for explicit anchor computations, enabling flexible handling of varying input resolutions. Input images are standardized, typically resized to a resolution of 640 × 640 pixels, ensuring that the backbone network processes consistently scaled, high-quality data.

Serving as the central component of YOLOv11, the backbone is responsible for extracting multi-scale features from the input images. Similar to YOLOv8, YOLOv11 uses convolutional layers to downsample images, reducing spatial dimensions while increasing channel depth. However, YOLOv11 introduces the C3k2 module, replacing the C2f module from YOLOv8. Additionally, the C2PSA module is included after the SPPF module to enhance spatial attention in feature maps.

The neck module in YOLOv11 uses the C3k2 module to process features at multiple scales, allowing for more detailed feature extraction. The flexibility of the C3k2 module is evident in its parameterization: when set to false, it adopts a structure similar to the C2f module, whereas when set to true, it replaces the bottleneck with the C3 module.

The head module refines the bounding-box predictions by using distributed focal loss, which enhances the precision of predictions from a probabilistic perspective. The discrete predictions produced by the network are mapped back into continuous coordinate space, generating the final detection bounding boxes.

Experiments conducted with YOLOv11 on the SIMD dataset resulted in a precision of 79.1%, a recall of 65.7%, an mAP@0.5 of 71.9%, and an mAP@0.5:0.95 of 45.8%.

3. The Proposed YOLOv11-SDC Model

This section introduces the network architecture of the proposed YOLOv11-SDC and provides a mathematical analysis of the roles played by the SF, DRB, and CGAF modules within the network. The practical impact of these modules on feature extraction is demonstrated using visualized heatmaps.

3.1. Network Structure

Although YOLOv11 demonstrates improvements in objects detection, it still faces challenges in focusing on the critical features of small objects in SSS images. Specifically, YOLOv11 struggles with identifying key features of small objects, has an insufficient receptive field for precise detection, and exhibits limited capability in feature extraction after multi-scale feature fusion.

To overcome these limitations, we propose YOLOv11-SDC, an enhancement of the YOLOv11 model. In YOLOv11-SDC, a new feature selection method, SF, replaces the SPPF module present in YOLOv11. While the SPPF module helps enhance the receptive field and feature representation through multi-scale pooling, it does not specifically focus on channel selection [33]. On the other hand, the SF module selects the most discriminative channels during forward propagation, effectively reducing feature redundancy. Given the background noise and subtle object features commonly found in SSS images, targeted channel selection enables the model to focus on important feature channels, improving feature contrast. By retaining only the most relevant channels, the SF module refines feature maps before they are processed by the subsequent convolution and detection layers. This feature sparsification helps reduce overfitting risk and enhances robustness in various marine environments.

YOLOv11-SDC also integrates the DRB module into the C3k2 module. By stacking multi-scale dilated convolutions and applying post-aggregation parameter restructuring, DRB combines multiple convolution branches into a large-kernel convolution. This technique expands the receptive field, allowing the model to capture finer object details in SSS images more effectively. Additionally, YOLOv11-SDC includes the CGAF module. Traditional feature fusion methods often overlook the varying importance of different feature sources and are vulnerable to noise interference. The CGAF module, utilizing a multi-attention mechanism, adaptively assigns weights to channels, spatial locations, and individual pixels, highlighting meaningful object features while minimizing irrelevant information.

The architecture of the YOLOv11-SDC model is presented in Figure 3.

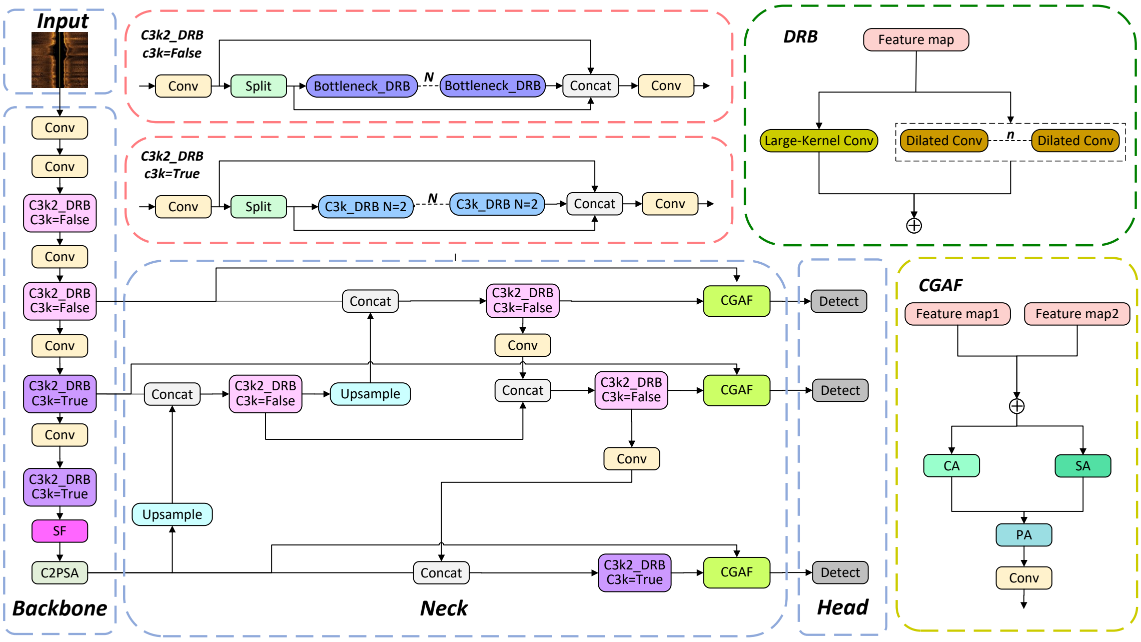


Figure 3. The network structure of YOLOv11-SDC.

3.2. SF Module

Since YOLOv11 fails to emphasize the critical features of small objects, we introduce the SF module. The SF module highlights the most relevant channels, thereby improving the representation of essential features within an image. By incorporating global attention, channel selection, and dimensionality reduction, SF enhances the neural network's focus on the object. The architecture of the SF module is shown in Figure 4, where H and W represent the spatial dimensions of the input feature map and C indicates the number of input channels.

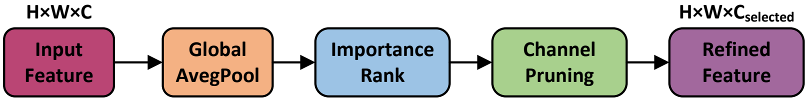


Figure 4. The structure of the SF module.

The SF module includes three key components: global feature aggregation, channel importance ranking, and channel pruning. During the global feature aggregation step, global average pooling [34–36] is applied to each channel to create a global feature representation. The global feature for each channel is represented by y_c, as shown in Equation (1), where $x_c(i,j)$ denotes the pixel value of channel c at position (i,j) and y_c is the global feature of the channel (c).

$$y_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j),\ c \in [1, C] \tag{1}$$

To make better use of the information derived from global average pooling, the average value of the global features for each channel is calculated, then ranked. This process is described in Equation (2).

$$\mu_c = \frac{1}{B} \sum_{b=1}^{B} y_c,\ b \in [1, B] \tag{2}$$

where B represents the batch size and b is the index of the batch. μ_c indicates the average feature of the c-th channel across the entire batch, with a matrix size of C, where each element corresponds to the global feature value of a channel. Next, the channels are ranked based on their global features, and the k channels with the highest feature values are chosen. The feature values (S_k) of these selected k channels are then calculated using Equation (3).

$$S_k = Select(y_c, k) \tag{3}$$

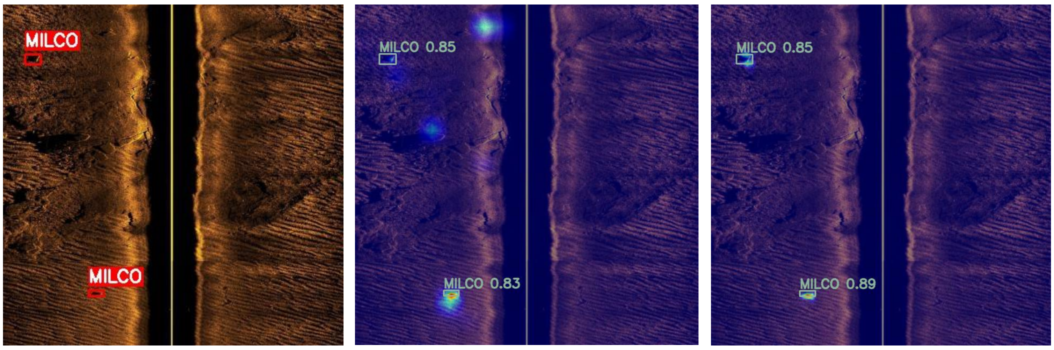
The output feature map is created by using the indices (c_i) of the selected key channels. This process entails choosing the most significant channels from the original feature map and setting all other channels to zero. $S_{c'}$ denotes the feature value of the c-th channel in the resulting output feature map. This operation is outlined in Equation (4).

$$S_c{}' = \begin{cases} S_c & c = c_i \\ 0 & else \end{cases} \tag{4}$$

The output feature map Y is reconstructed using Equation (5).

$$Y = [S_1', S_2', \ldots, S_c'] \tag{5}$$

To further demonstrate the practical impact of the SF module on feature extraction, we conducted a heatmap visualization comparison between YOLOv11s and YOLOv11s-SF, which incorporates the SF module, as shown in Figure 5. The heatmap overlays the model's attention distribution on the original image, providing an intuitive display of the areas the model considers important during detection. In the heatmap, the color intensity represents the level of attention, with brighter colors indicating higher attention and a higher likelihood of containing objects. This visualization not only verifies whether the model correctly focuses on the object regions but also reveals potential areas of false detection.



(a) (b) (c)

Figure 5. Heatmap comparison, where warm colors indicate higher confidence and cool colors indicate lower confidence: (**a**) original image; (**b**) YOLOv11s; (**c**) YOLOv11s-SF.

In the example shown in Figure 5, both YOLOv11s and YOLOv11s-SF successfully detect the objects in the SSS image. However, unlike the heatmap of YOLOv11s, which shows bright areas in the background, the heatmap of YOLOv11s-SF concentrates its bright regions exclusively on the objects. This visualization further illustrates the role of the SF module in channel importance ranking and channel pruning, highlighting its effectiveness in extracting key features of the objects while reducing background interference.

3.3. DRB Module

To overcome the issue of a limited receptive field in YOLOv11 for the detection of small objects, we introduce the DRB module. This module combines dilated convolutions [37] with reparameterization [38] techniques to improve the model's feature representation ability. Dilated convolution increases the receptive field [39] by adding gaps between elements in the convolution kernel, without introducing additional parameters.

As depicted in Figure 6, (a) uses a dilated convolution with a dilation rate of 1, providing each element with a receptive field of 3 × 3; (b) applies a dilation rate of 2, resulting in a receptive field of 7 × 7 for each element; and (c) uses a dilation rate of 4, achieving a receptive field of 15 × 15 per element.

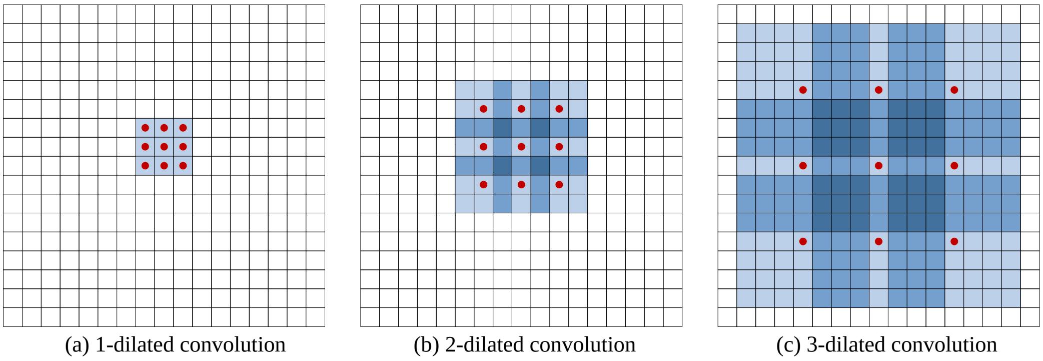

(a) 1-dilated convolution (b) 2-dilated convolution (c) 3-dilated convolution

Figure 6. Receptive field of *l*-dilated convolutions, where blue represents the range of the receptive field.

In DRB, the reparameterization process primarily involves converting the complex multi-branch structure used during training into a more efficient single-branch structure for the inference phase, which is achieved through a series of mathematical transformations and the merging of parameters.

During training, the DRB comprises multiple parallel dilated convolution branches, each with different dilation rates and kernel sizes. Specifically, we define $Conv_0$ as the original convolution branch with weights of W_0, and bias b_0 refers to the batch normalization (BN) [40] layer following the backbone branch, characterized by parameters of γ_0, β_0, μ_0, and σ_0. $Conv_{k,r}$ denotes the k-th parallel dilated convolution branch with a dilation rate of r and weights of $W_{k,r}$. Similarly, $BN_{k,r}$ represents the BN layer following each parallel branch, with parameters of $\gamma_{k,r}$, $\beta_{k,r}$, and $\sigma_{k,r}$. The input and output feature maps are denoted as x and y, respectively, and are assumed to have dimensions of $x, y \in \mathbb{R}^{B\times C\times H\times W}$. Here, B represents the batch size, C is the number of channels, and H and W correspond to the height and width of the feature maps, respectively. In the training phase, the output (y) of the DRB is defined as Equation (6).

$$y = BN_0(Conv_0(x)) + \sum_{k,r} BN_{k,r}(Conv_{k,r}(x)) \tag{6}$$

Each branch's convolution operation is described in Equation (7).

$$\begin{cases} Conv_0(x) = W_0 * x + b_0 \\ Conv_{k,r}(x) = W_{k,r} * x \end{cases} \tag{7}$$

To streamline the model and enhance inference efficiency, DRB merges BN layers with convolution layers in each convolution branch. The output of the BN layer for each branch is defined in Equation (8).

$$BN(Conv(x)) = \gamma \frac{Conv(x) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \tag{8}$$

In Equation (8), ϵ is a small constant added to prevent division by zero. After the BN layer is fused with the convolution layer, the equivalent convolution layer parameters are obtained, as shown in Equation (9).

$$\begin{cases} W' = \gamma \frac{W}{\sqrt{\sigma^2+\epsilon}} \\ b' = \gamma\left(\frac{b-\mu}{\sqrt{\sigma^2+\epsilon}}\right) + \beta \end{cases} \tag{9}$$

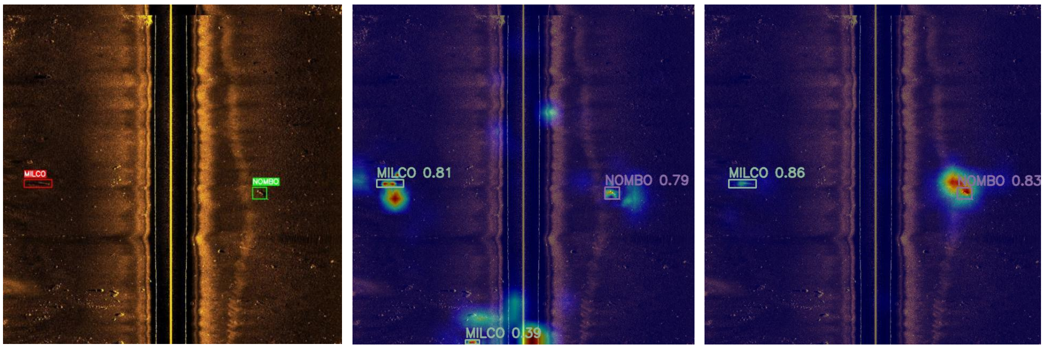
During the inference phase, DRB merges fused convolution branches into a single convolution layer. All branch convolution kernels ($W_{0\prime}$ and $W_{k,r}$) are superimposed onto a larger convolution kernel, and the bias terms ($b_{0\prime}$), so $b_{k,r\prime}$ is accumulated. The merged single convolution layer has weights of W_d, as illustrated in Equation (10).

$$\begin{cases} W_d = W_0' + \sum_{k,r} W_{k,r}' \odot M_{k,r} \\ b_d = b_0' + \sum_{k,r} b_{k,r}' \end{cases} \tag{10}$$

where $\odot$ denotes elementwise addition and $M_{k,r}$ is a matrix used to embed convolution kernels ($W_{k,r\prime}$) with different dilation rates into a larger convolution kernel (W_d), ensuring that the convolution operations of each branch do not spatially overlap. The output of DRB during the inference phase, denoted as Y, is presented in Equation (11).

$$Y = W_d * x + b_d \tag{11}$$

Figure 7 illustrates the differences in attention between YOLOv11s and YOLOv11s-DRB, which incorporates the DRB module. The comparison shows that, compared to YOLOv11s, YOLOv11s-DRB expands the receptive field to capture a broader range of input features, enabling it to better capture global information. As a result, it more accurately identifies the central positions of objects and reduces background interference.



(a) (b) (c)

Figure 7. Heatmap comparison, where warm colors indicate higher confidence and cool colors indicate lower confidence: (**a**) original image; (**b**) YOLOv11s; (**c**) YOLOv11s-DRB.

3.4. CGAF Module

To address the insufficient feature extraction capability of YOLOv11 after multi-scale feature fusion, we introduce the CGAF module. The core idea of the CGAF module is to extract attention weights across multiple scales and dimensions, then fuse two input feature maps to generate the output feature map. CGAF consists of three components: spatial attention (SA), channel attention (CA), and pixel attention (PA). The input feature map is assumed to be $x, y \in \mathbb{R}^{B\times C\times H\times W}$. As defined in Equation (12), an elementwise addition of x and y is performed to form X. All subsequent attention computations (SA/CA/PA) are based on X. Finally, the learned attention map is used to fuse x and y again.

$$X = x + y \tag{12}$$

SA aims to identify the most relevant spatial locations in the input feature map for further processing. This is accomplished by combining average-pooled and max-pooled [41] feature maps along the spatial dimension, followed by convolution to produce a spatial attention map. Initially, channel-wise average pooling, as defined in Equation (13), is applied to reduce the information from C channels into a single-channel representation.

$$X_{avg}(b,1,h,w) = \frac{1}{C}\sum_{c=1}^{C} X(b,c,h,w) \tag{13}$$

Subsequently, channel-wise max pooling, as described in Equation (14), is carried out to determine the maximum activation value at each spatial position across all channels, resulting in an additional single-channel feature map.

$$X_{max}(b,1,h,w) = \max_{c=1,\ldots,C} X(b,c,h,w) \tag{14}$$

The average-pooled and max-pooled feature maps are subsequently merged along the channel dimension, as shown in Equation (15).

$$X_{concat} = \text{concat}(X_{avg}, X_{max}) \tag{15}$$

Finally, a convolution is applied to X_{concat}. The convolution kernel slides over the $H\times$ spatial dimensions, blending and weighting the information from X_{avg} and X_{max}. By learning the parameters (W_s) and the bias (b_s), the convolution kernel automatically determines the optimal weighting scheme. This produces the output feature map (S_{pos}), which captures the importance of each spatial location, as defined in Equation (16).

$$S_{pos} = W_s * X_{concat} + b_s, S_{pos} \in \mathbb{R}^{B\times 1\times H\times W} \tag{16}$$

Based on this, SA offers a combined representation of the “overall intensity” and “local saliency” at each spatial position.

Since different channels in the input feature map often capture distinct feature patterns or semantic information, not all channels hold the same level of importance for a specific task. CA evaluates the relative importance of each channel by first calculating the global activation strength through global average pooling. It then applies a sequence of dimensionality reduction, non-linear activation, and dimensionality expansion, allowing the model to learn channel relationships and assign varying weights. Specifically, as defined in Equation (17), CA computes the average of the input feature map (X) along the spatial dimensions to produce X_s.

$$X_s(b,c,1,1) = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} X(b,c,h,w), X_s \in \mathbb{R}^{B\times C\times 1\times 1} \tag{17}$$

A two-layer 1 × 1 convolutional network is then applied to the channels. The first convolutional layer, as defined in Equation (18), reduces the dimensionality from C to C/r, thereby decreasing the number of parameters and helping to prevent overfitting. $W_{c1} \in \mathbb{R}^{\frac{C}{r}\times C\times 1\times 1}$ and $b_{c1} \in \mathbb{R}^{\frac{C}{r}}$ are the weights and biases of the first 1×1 convolutional layer, while $U \in \mathbb{R}^{B\times\frac{C}{r}\times 1\times 1}$ denotes its output. r is a hyperparameter known as the reduction ratio. It controls how much the channel dimension C is compressed before being restored to its original size, thereby reducing both the model’s parameters and computational cost.

$$U = W_{c1}\times X_s + b_{c1}, U \in \mathbb{R}^{B\times(\frac{C}{r})\times 1\times 1} \tag{18}$$

As defined in Equation (19), this is followed by a ReLU [42] activation function, which enhances the model's non-linear fitting ability.

$$U' = \mathrm{ReLU}(U) \tag{19}$$

Finally, as defined in Equation (20), the second convolutional layer is employed to restore the dimensionality back to the original channel count (C), producing the channel attention weights (C_{ch}) for the input feature map. Similarly, $W_{c2} \in \mathbb{R}^{C \times \frac{C}{r} \times 1 \times 1}$ and $b_{c2} \in \mathbb{R}^{C}$ correspond to the second 1×1 convolutional layer.

$$C_{ch} = W_{c2} \times U' + b_{c2}, C_{ch} \in \mathbb{R}^{B \times C \times 1 \times 1} \tag{20}$$

Channel attention and spatial attention are combined. This results in an initial pixel attention map (P_{1}). As defined in Equation (21), it considers both channel and spatial dimensions.

$$P_{\mathit{1}} = S_{pos} + C_{ch} \tag{21}$$

Then, X and P_1 are concatenated along the channel dimension. For each sample (b), channel (c), and spatial position ($\left(h, w\right)$), a two-channel vector is constructed to form $Z' \in \mathbb{R}^{B \times 2C \times H \times W}$. A learnable linear mapping is then applied to Z'. In the experiment, the mapping is implemented via a convolution operation with a kernel size of 7×7. For the c-th channel of the output, the convolution uses parameters of $W_c \in \mathbb{R}^{2 \times 7 \times 7}$ and bias (b_c). As defined in Equation (22), when the center of the convolution kernel aligns with position $\left(h, w\right)$, the linear response at $\left(h, w\right)$ is given by

$$L_c(b, h, w) = \sum_{i=1}^{2} \sum_{u=-3}^{3} \sum_{v=-3}^{3} W_c(i, u, v) \cdot Z'(b, 2c + i, h + u, w + v) + b_c. \tag{22}$$

$\left(u, v\right)$ ranges from -3 to 3, representing the 7×7 neighborhood centered at $\left(h, w\right)$. One channel of $Z\prime$ is derived from X, and the other is derived from P_{1}. The mapping for each channel is performed independently. Finally, as defined in Equation (23), a sigmoid function [43] is applied to the linear response ($L_c(b, h, w)$) to convert it into an attention weight.

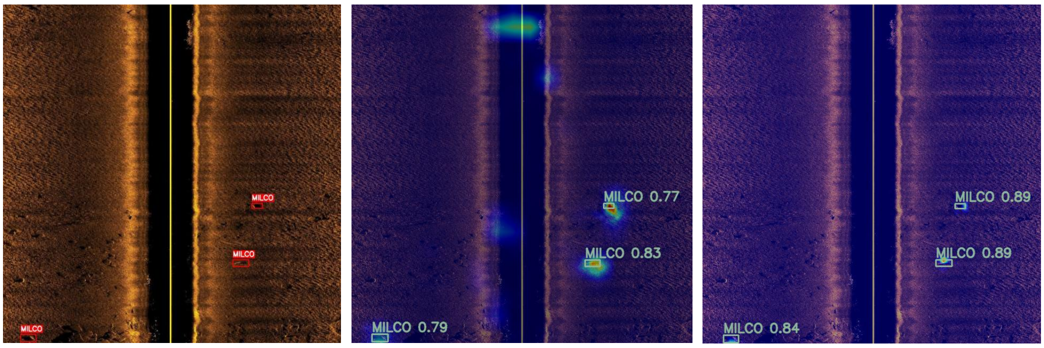
$$P_2(b, c, h, w) = \sigma(L_c(b, h, w)) = \frac{1}{1 + e^{-L_c(b,h,w)}} \tag{23}$$

For the input feature map ($x, y \in \mathbb{R}^{B \times C \times H \times W}$), the features are uniformly combined via $x + y$. The result is further fused with P_2 through Equation (24) ($\circ$ represents elementwise multiplication).

$$P_3 = X + P_2 \circ x + (1 - P_2) \circ y \tag{24}$$

Finally, a 1×1 convolution is applied to P_3 for a linear mapping.

With the introduction of the CGAF module, YOLOv11s was extended to form YOLOv11s-CGAF. As shown in Figure 8, the addition of the CGAF module enables the model to effectively reduce attention on background regions through multi-scale feature fusion and weighting. The bright regions in the heatmap are more concentrated on the object locations. Compared to YOLOv11s, YOLOv11s-CGAF demonstrates a more precise distribution of bright regions, along with a significant improvement in object recognition confidence.



(a) (b) (c)

Figure 8. Heatmap comparison, where warm colors indicate higher confidence and cool colors indicate lower confidence: (**a**) original image; (**b**) YOLOv11s; (**c**) YOLOv11s-CGAF.

4. Experiments

This section presents the evaluation metrics used in the experiments, the hardware and hyperparameter configurations, and the dataset. It also determines the optimal number of feature channels retained by the SF module. Furthermore, ablation experiments demonstrate the superiority of the proposed YOLOv11-SDC.

4.1. Model Evaluation Metrics

In object detection tasks, performance metrics such as Intersection over Union(IoU) [44], precision [45], recall [46], and mAP [47] are commonly used to evaluate a model's performance on a given dataset. During experiments, the model predicts a set of bounding boxes for the input images, each with an associated confidence score. In the object detection process, the predicted bounding boxes are compared and matched with the ground-truth bounding boxes to evaluate prediction accuracy. As defined in Equation (25), the IoU metric is calculated as the ratio of the overlap between the predicted and ground-truth boxes to the area of their union.

$$IOU = \frac{DetectionResult \cap GroudTruth}{DetectionResult \cup GroudTruth} \tag{25}$$

Precision quantifies the proportion of true-positive predictions among all predicted instances. After setting an IoU threshold (0.25 in our experiment), the predicted bounding boxes are matched with the ground-truth boxes. A prediction is considered a true positive (TP) if the predicted bounding box meets the IoU threshold and the predicted category is correct. If the model predicts an object that is not present in the ground truth or assigns the wrong category, the prediction is considered a false positive (FP). Conversely, if a ground-truth object is not detected by the model, it is categorized as a false negative (FN). The definition of precision is expressed as Equation (26).

$$Precision = \frac{TP}{TP + FP} \tag{26}$$

Recall measures the model's ability to identify all true targets in an image. It calculates the proportion of ground-truth objects successfully detected by the model. The definition of recall is expressed as Equation (27).

$$Recall = \frac{TP}{TP + FN} \tag{27}$$

The Precision–Recall (PR) curve represents the trade-off between the precision of correctly identifying positive instances and the recall of detecting all positive instances. In this curve, precision is plotted on the vertical axis, while recall is shown on the horizontal axis. Average Precision (AP) [48] represents the area under the PR curve, condensing it into a single scalar value. The definition of AP is expressed as Equation (28).

$$AP = \int_0^1 Precision(Recall)dRecall \tag{28}$$

For tasks with multiple object categories, average precision (AP) is computed individually for each category. The mAP is then derived by averaging the AP values across all categories. The definition of mAP is expressed as Equation (29), where N is the total number of categories.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{29}$$

4.2. Experimental Environment

The experiment was carried out on a Windows 11 operating system, using PyCharm as the development environment. The YOLOv11s framework from Ultralytics was modified and optimized to address the challenges associated with object detection in SSS images. The hardware configuration used in the experiment is provided in Table 1.

Table 1. Hardware configuration.

Name	Configuration
CPU	Intel(R) Core(TM) i9-14900KF
GPU	NVIDIA GeForce RTX 4090 D
Memory	96GB

The software configuration is shown in Table 2.

Table 2. Software configuration.

Name	Configuration
Python	3.10.14
Pytorch	2.2.2
CUDA	12.1

4.3. Model Hyperparameter Settings

The hyperparameters used for training are shown in Table 3.

Table 3. Hyperparameter settings.

Parameter	Configuration
Learning rate	0.009
Weight decay	0.0005
Batch size	32
Optimizer	SGD
Image size	640 × 640
Epochs	600

Figure 9 illustrates the impact of different numbers of training epochs on the model's performance. Through experimental validation, 600 epochs were identified as the most suitable number of training iterations.

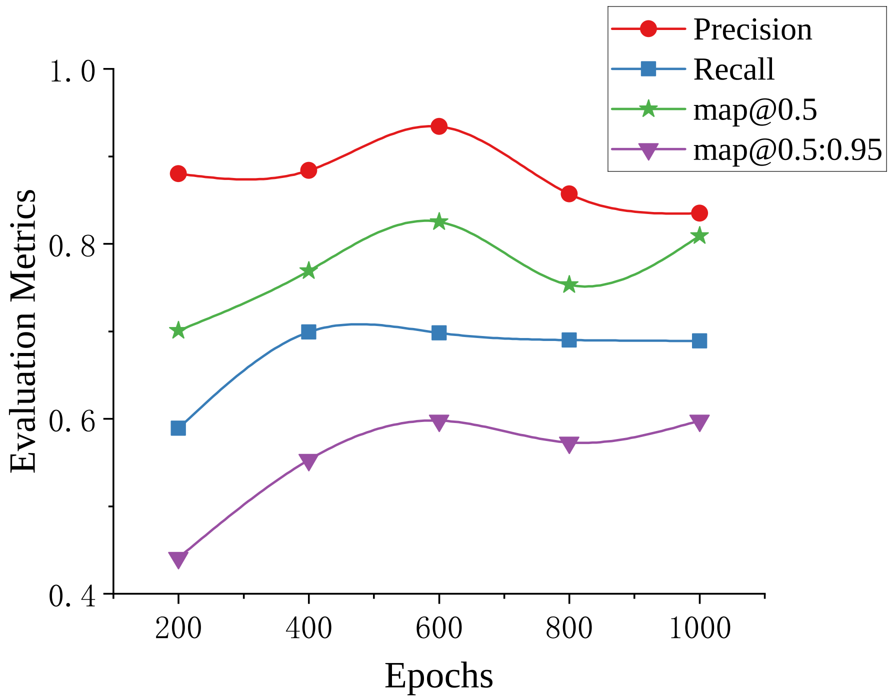


Figure 9. Evaluation metrics with different numbers of epochs.

4.4. Number of Feature Channels Retained by the SF Module

As introduced in Section 3.2, the SF module enhances the focus on key object features through channel importance ranking and pruning. Therefore, the number of retained feature channels impacts the model's detection performance. As shown in Table 4, we conducted experiments to evaluate YOLOv11s-SDC's performance in terms of precision, recall, mAP@0.5, and mAP@0.5:0.95 by retaining 16, 24, 32, 40, and 48 channels, respectively.

Table 4. Comparison of evaluation metrics with different numbers of retained feature channels.

Number	Precision	Recall	map@0.5	map@0.5:0.95
16	0.899	0.701	0.771	0.546
24	0.892	0.709	0.786	0.576
32	0.934	0.698	0.825	0.598
40	0.908	0.708	0.822	0.596
48	0.867	0.701	0.807	0.58

The experimental results show that the model's detection performance is similar when retaining 16 and 24 channels. When the number of retained channels increases to 32, the precision of YOLOv11-SDC improves by 3.5% and 4.2% compared to retaining 16 and 24 channels, respectively, while the recall decreases by 0.3% and 1.1%. Additionally, mAP@0.5 improves by 5.4% and 3.9%, and mAP@0.5:0.95 increases by 5.2% and 2.2%.

As the number of retained channels increases to 40, the recall improves by 1% compared to retaining 32 channels, but the other three metrics decline, with precision notably dropping by 3%. When the number of retained channels further increases to 48, all four metrics decrease compared to retaining 40 channels.

The significant improvement in precision and the slight decrease in recall when retaining 32 channels indicate that the model achieves a better balance between precision and recall. Moreover, the mAP@0.5 and mAP@0.5:0.95 values reach their highest levels under this configuration. Therefore, we selected 32 as the number of retained feature channels for the experiment.

4.5. The SIMD Dataset

The SIMD dataset utilized in this study was obtained from https://figshare.com/articles/dataset/i_Side-scan_sonar_imaging_for_Mine_detection_i/24574879 (accessed on 28 October 2024). It contains 1170 SSS images collected by an AUV equipped with SSS, of which 304 include objects. Since the original dataset has already been augmented by the authors, only 154 of the 304 object-containing images are unprocessed SSS images.

To ensure the reliability of detection performance, the 154 unprocessed images were divided into training, validation, and testing sets in a 70:15:15 ratio, while the remaining augmented images were added to the training set. The basic configuration of the dataset is shown in Table 5.

Table 5. Dataset split settings.

Dataset	Images	Instances
Train	1123	558
Test	24	55
Val	23	54

The object distribution within the dataset is illustrated in Figure 10.

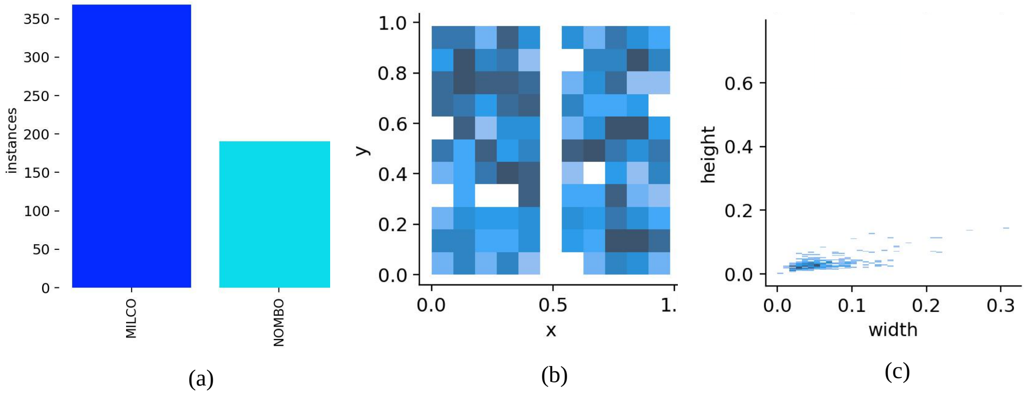


Figure 10. The statistical results of the SIMD dataset: (**a**) bar chart of object counts by category; (**b**) normalized distribution of object locations; (**c**) normalized distribution of object sizes.

4.6. Experimental Results and Analysis

The proposed YOLOv11s-SDC model was tested on the dataset presented in Section 4.3 to evaluate its performance. After 600 training epochs, the model achieved convergence. As illustrated in Figure 11, box_loss measures the difference between the predicted and ground-truth bounding boxes, with smaller values indicating better detection accuracy. cls_loss assesses the gap between the predicted and true class labels, where lower values reflect improved classification accuracy. Moreover, dfl_loss converts the continuous coordinate prediction task into a discrete probability distribution prediction, enabling more accurate localization of coordinates. Smaller values indicate superior prediction accuracy.

As depicted in Figure 12, the YOLOv11s-SDC model exhibits significant enhancements in object detection for SSS compared to the original YOLOv11s. Notably, the accuracy for MICLO rose by 5.1%, while NOMBO experienced an impressive increase of 16.1%. Overall, the accuracy across all categories, evaluated using the mAP@0.5 metric, improved by 10.6%.

Additionally, this study compares the proposed model with several popular models, including YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, and YOLOv11s, evaluating its superiority in four evaluation metrics.

The experimental results, displayed in Figure 13, highlight that YOLOv11s-SDC outperforms all other models across these four metrics at every training stage. The performance curves of YOLOv9s and YOLOv8s are relatively similar, with the two models showing close metrics after convergence in the later stages of training. Except for recall, the three other metrics of these two models are only slightly below those of YOLOv11s-SDC. YOLOv11s achieves a higher recall but exhibits the lowest precision, reflecting the model's inability to balance precision and recall effectively. The overall performance of YOLOv5s is slightly lower than that of YOLOv8s, YOLOv9s, and YOLOv11s. YOLOv10s shows low performance across all four metrics during most of the training stages, making it the least effective model in this experiment, with noticeable gaps in recall and mAP@0.5 compared to the other models.

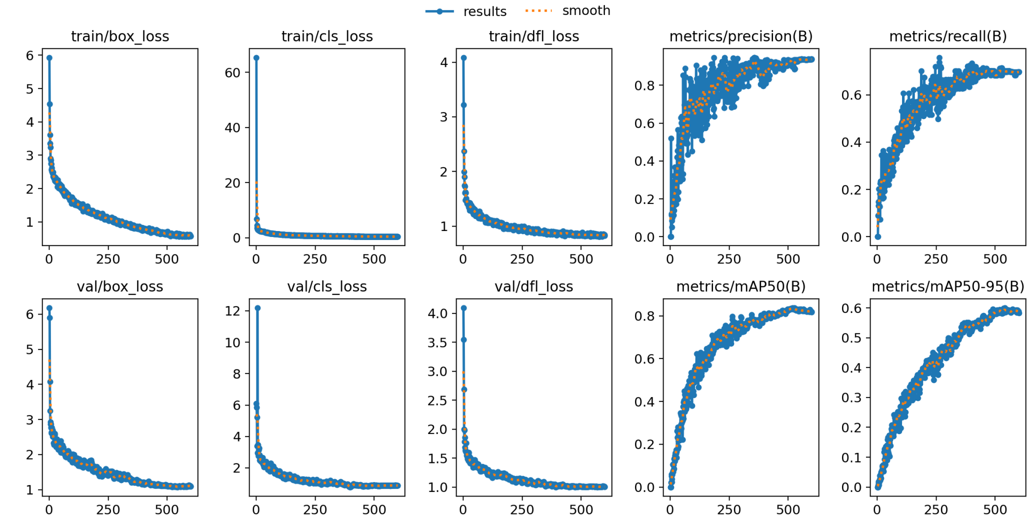


Figure 11. Training and validation losses and metric progression.

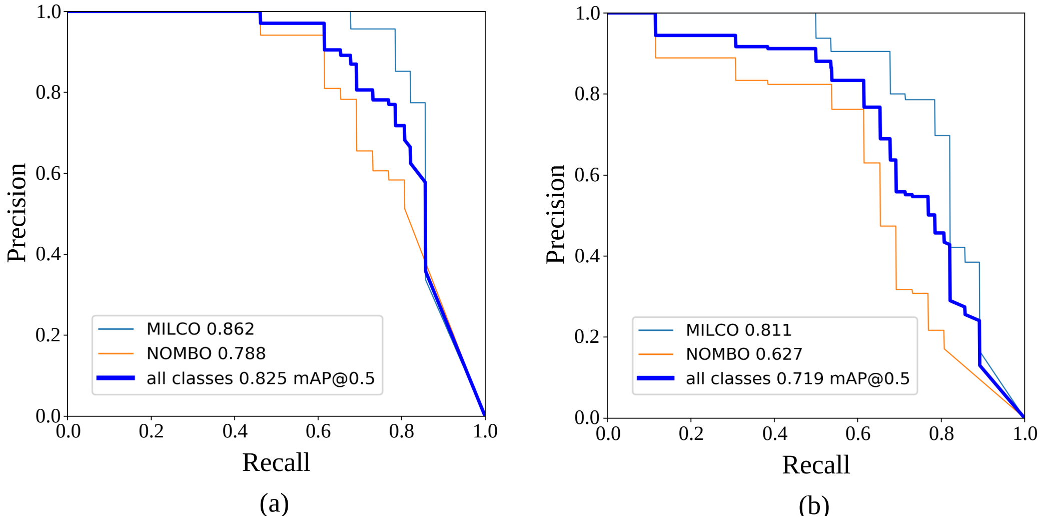


Figure 12. Precision–recall curve: (**a**) YOLOv11s-SDC; (**b**) YOLOv11s.

It is important to note that the authors of the SIMD dataset achieved 82% precision, 64% recall, and 75% mAP in their experiments. In comparison, our experiment achieved 93.4% precision, 70.1% recall, and 82.4% mAP, further emphasizing the superior performance of the proposed YOLOv11-SDC model.

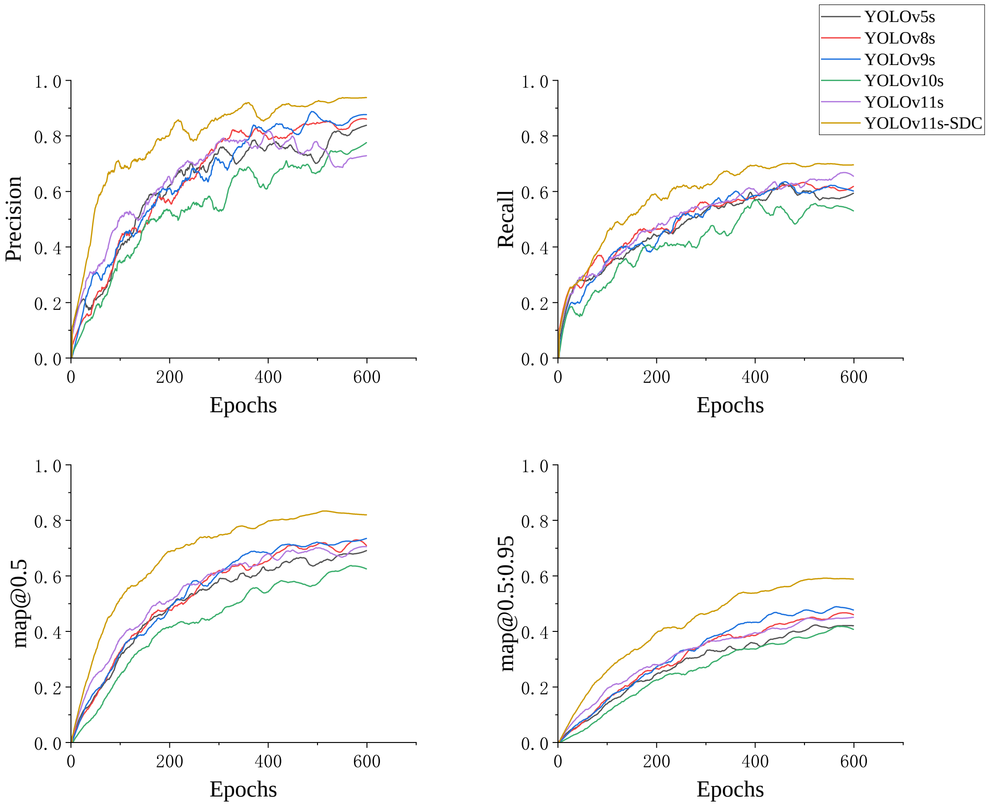


Figure 13. Evaluation and comparison of metric curves of YOLOv11s-SDC and baseline models.

4.7. Ablation Experiments

Ablation experiments were performed to further assess the performance improvements resulting from the integration of the SF, DRB, and CGAF modules into the YOLOv11s-SDC model. The study examined the training performance of the YOLOv11s, YOLOv11s-SF, YOLOv11s-DRB, YOLOv11s-CGAF, YOLOv11s-SF+DRB, YOLOv11s-SF+CGAF, YOLOv11s-DRB+CGAF, and YOLOv11s-SDC models across four evaluation metrics.

As seen in Figure 14, introducing the SF, DRB, and CGAF modules into YOLOv11s individually results in performance improvements across most evaluation metrics, particularly precision, compared to YOLOv11s. The only exception is that YOLOv11s-CGAF shows a slightly lower recall than YOLOv11s. YOLOv11s-SF, YOLOv11s-DRB, and YOLOv11s-CGAF exhibit similar performance in terms of precision and recall. Notably, YOLOv11s-DRB demonstrates good performance in both mAP@0.5 and mAP@0.5:0.95.

With the SF, DRB, and CGAF modules integrated together, YOLOv11s is extended to form YOLOv11s-SDC. The precision curve for YOLOv11s-SDC remains higher than that of the other models for most of the training, ultimately reaching the highest value. In the later stages of training, it also shows better stability and converges more effectively. Likewise, YOLOv11s-SDC demonstrates higher recall than the other models throughout the majority of the training process, with the best convergence occurring in the later stages. In terms of the mAP@0.5 metric, YOLOv11s-SDC consistently outperforms the other models. Even under the stricter mAP@0.5:0.95 metric, YOLOv11s-SDC continues to maintain its advantage, with its curve surpassing all others, indicating superior detection performance.

To validate the necessity of combining the SF, DRB, and CGAF modules,we conducted comparative experiments on YOLOv11s, YOLOv11s-SF+DRB, YOLOv11s-SF+CGAF, YOLO v11s-DRB+CGAF, and YOLOv11s-SDC.

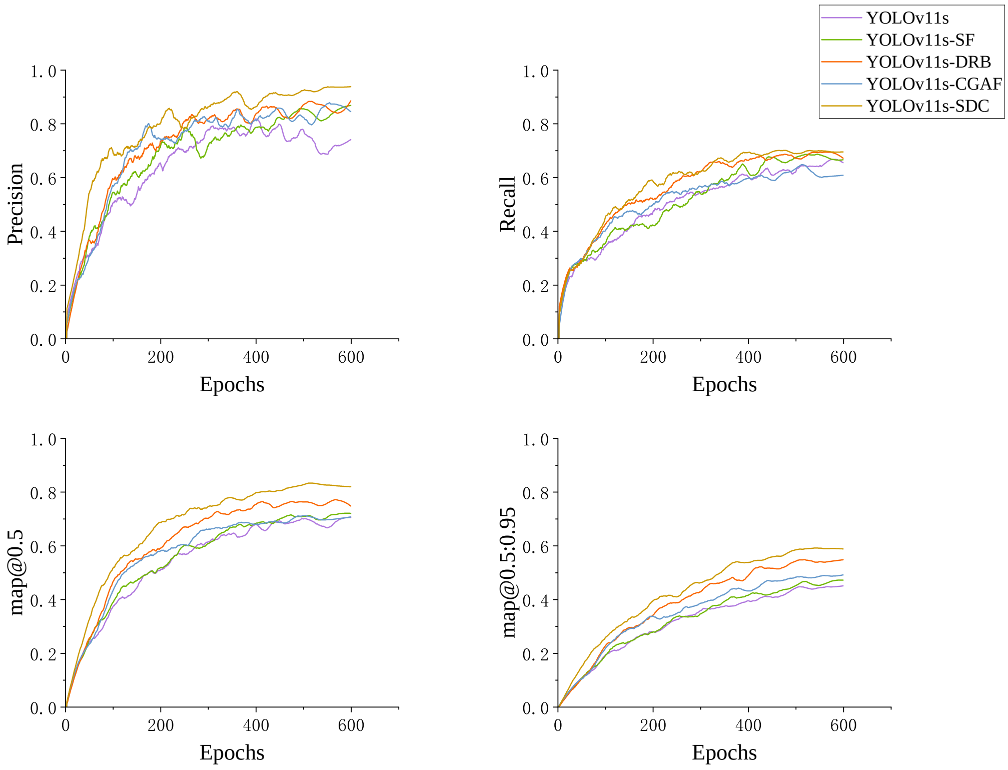


Figure 14. Evaluation metrics comparison curve of YOLOv11s-SDC and single-module models.

As shown in Figure 15, integrating any two of these three modules into YOLOv11 improves all four evaluation metrics relative to the original YOLOv11 model. Furthermore, incorporating all three modules into YOLOv11 results in the YOLOv11-SDC model, which achieves better performance across four evaluation metrics compared to using any two modules. The precision curve for YOLOv11s-SDC remains higher than those of the other models for most of the training, ultimately reaching the highest value. In the later stages of training, it also shows better stability and converges more effectively. Likewise, YOLOv11s-SDC demonstrates higher recall than the other models throughout the majority of the training process, with the best convergence occurring in the later stages. In terms of the mAP@0.5 metric, YOLOv11s-SDC consistently outperforms the other models. Even under the stricter mAP@0.5:0.95 metric, YOLOv11s-SDC continues to maintain its advantage, with its curve surpassing all others, indicating superior detection performance.

As shown in Table 6, YOLOv11s-SF and YOLOv11s-DRB outperform other baseline models in terms of precision and recall, while YOLOv11s-CGAF ranks just below YOLOv9s. Both YOLOv11s-SF and YOLOv11s-DRB achieve higher mAP@0.5 compared to other baseline models. In terms of mAP@0.5:0.95, YOLOv11s-DRB and YOLOv11s-CGAF exceed all other baseline models, with YOLOv11s-DRB demonstrating a particularly significant lead. After incorporating the SF, DRB, and CGAF modules into YOLOv11s, its performance surpasses that of improvements using any two of these modules. By integrating the SF, DRB, and CGAF modules, YOLOv11s-SDC achieves superior performance in four evaluation metrics, surpassing all the models in the experiment.

YOLOv11s-SDC shows a 14.3% increase in precision, a 4.1% increase in recall, a 10.6% improvement in mAP@0.5, and a 14% increase in mAP@0.5:0.95 compared to YOLOv11s. These results further demonstrate the advancements introduced by the YOLOv11s-SDC model.

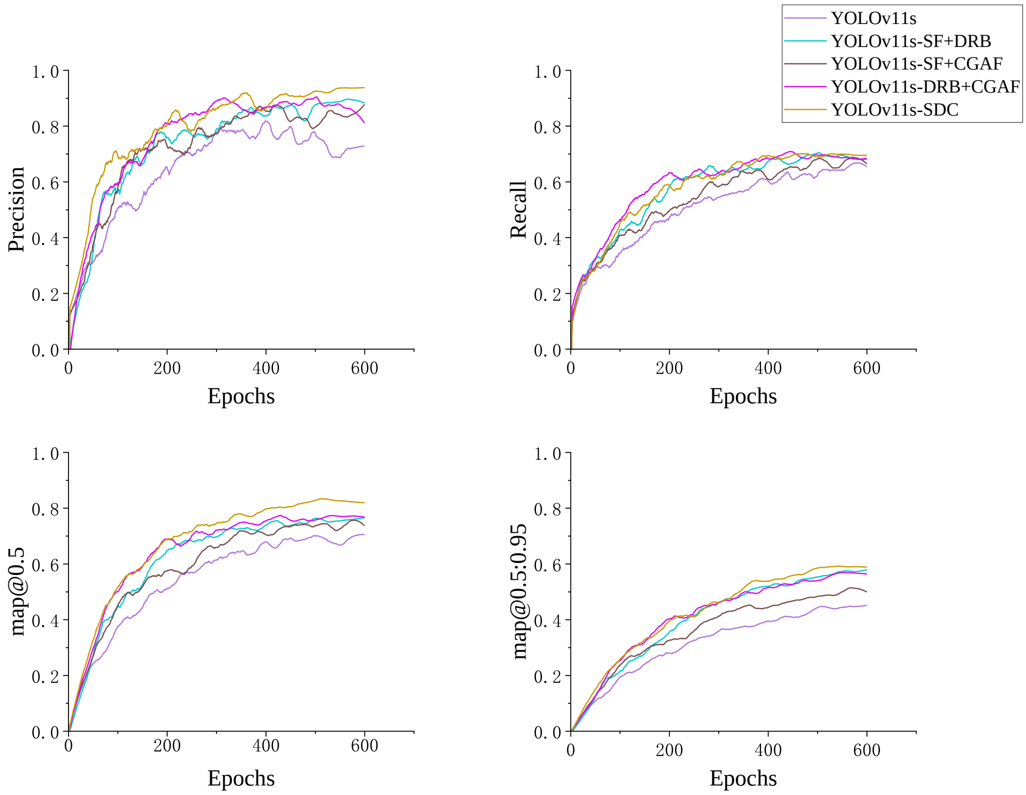
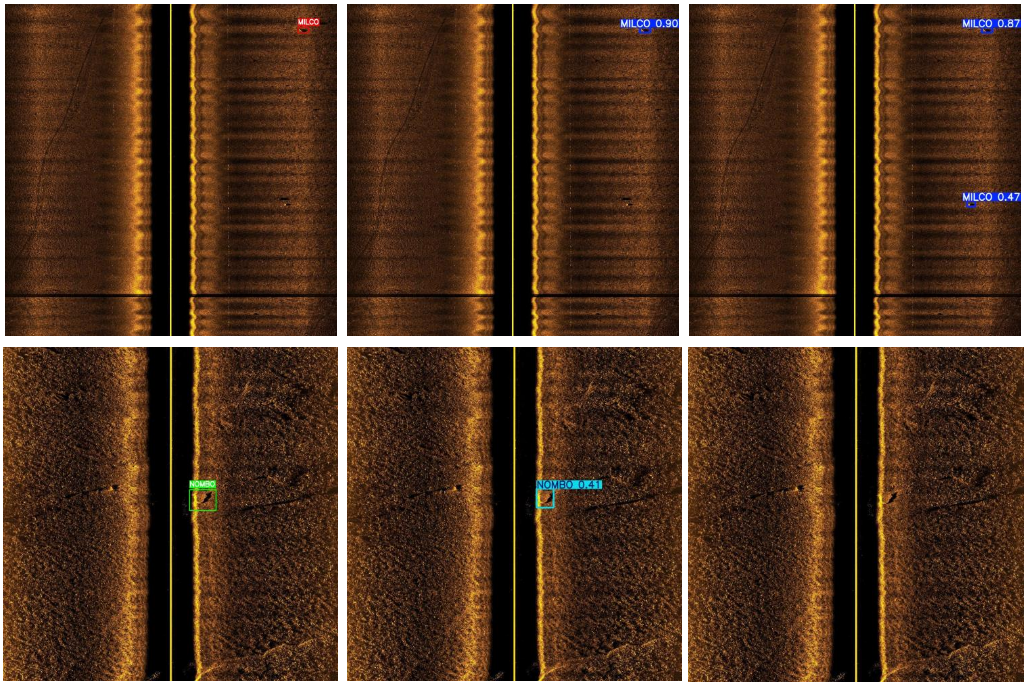


Figure 15. Evaluation and comparison of metric curves of YOLOv11s-SDC and dual-module models.

Table 6. Evaluation metric comparison table of YOLOv11s-SDC and other models.

Model	Precision	Recall	map@0.5	map@0.5:0.95
YOLOv5s	0.814	0.607	0.678	0.434
YOLOv8s	0.729	0.668	0.746	0.473
YOLOv9s	0.878	0.605	0.726	0.49
YOLOv10s	0.789	0.545	0.642	0.425
YOLOv11s	0.791	0.657	0.719	0.458
YOLOv11s-SF	0.882	0.685	0.727	0.478
YOLOv11s-DRB	0.886	0.676	0.762	0.561
YOLOv11s-CGAF	0.856	0.607	0.71	0.494
YOLOv11s-SF+DRB	0.9	0.693	0.759	0.581
YOLOv11s-SF+CGAF	0.826	0.693	0.757	0.518
YOLOv11s-DRB+CGAF	0.873	0.681	0.771	0.572
YOLOv11s-SDC	0.934	0.698	0.825	0.598

Figure 16 compares the detection performance of YOLOv11s-SDC and YOLOv11s. The results show that the proposed YOLOv11s-SDC model not only attains higher classification accuracy but also exhibits a better recall rate compared to YOLOv11s.



(a) (b) (c)

Figure 16. Example of detection result comparison: (**a**) original image; (**b**) YOLOv11s-SDC; (**c**) YOLOv11s.

5. Analysis of Limitations

Section 4 demonstrated the superior performance of YOLOv11s-SDC compared to other models. To further analyze its performance comprehensively, this section conducts a qualitative comparison of the visual differences between correct detections, missed detections, and false detections across different models. Additionally, the trained YOLOv11s-SDC model is applied to noisy SSS images to further evaluate its noise resistance capabilities.

5.1. Qualitative Analysis

Using Figure 17 as an example, the differences in small object detection performance across various models are compared. The SSS image in Figure 17 contains three MILCO objects and six seabed interference objects. For clarity, the six interference objects are numbered from 1 to 6, where 2 is a large interference object and the others are small interference objects.

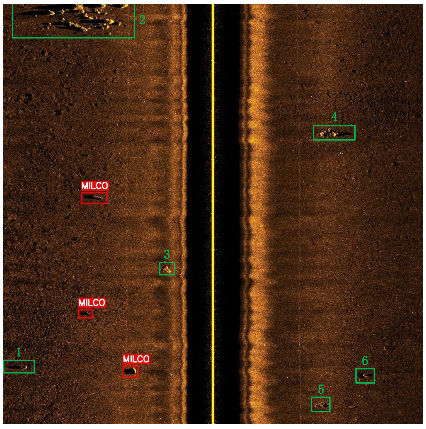
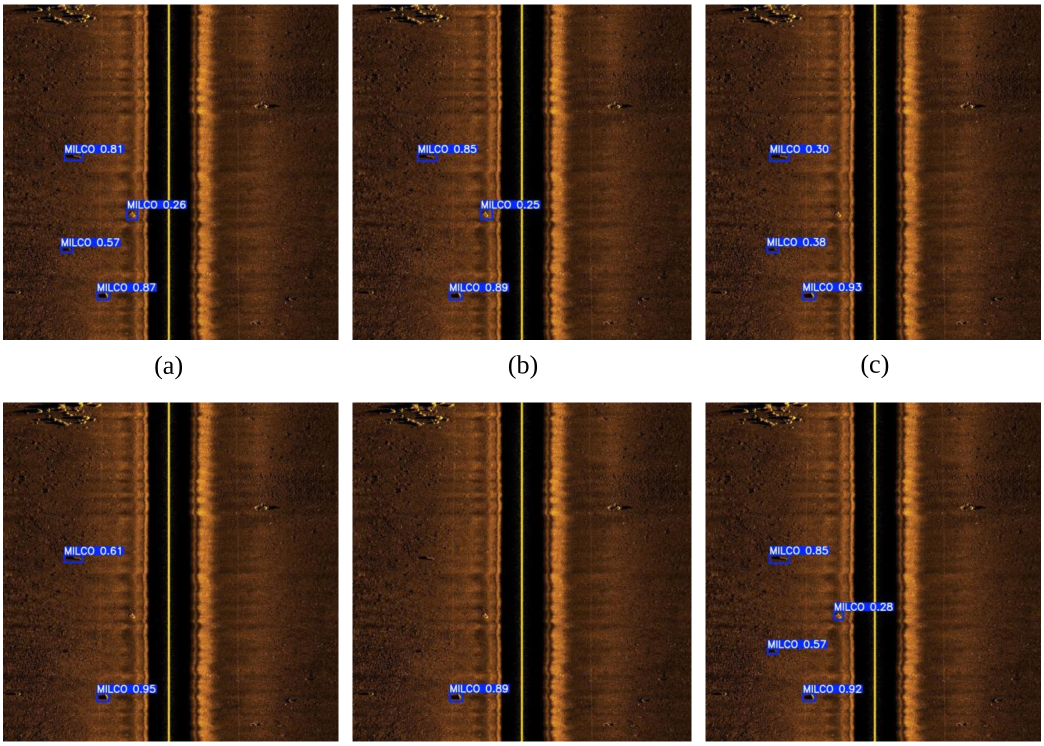


Figure 17. SSS image containing objects and interference.

Figure 18 shows the detection results of YOLOv5s, YOLOv8s, YOLOv9s, YOLOv10s, YOLOv11s, and YOLOv11s-SDC. The results indicate that none of the models mistakenly identified interference objects other than *3* as targets, demonstrating their ability to resist false positives when dealing with large seabed interference objects. However, the models differ in their ability to resist false positives with small interference objects.

YOLOv9s successfully detected all objects, but the low confidence scores of the detections make it unsuitable for applications requiring high confidence. YOLOv5s and YOLOv11s-SDC both identified all three small objects with high confidence but mistakenly classified *3* as a MILCO. YOLOv8s and YOLOv10s detected two of the three objects, with YOLOv8s also misclassifying *3*. YOLOv11s detected only one object, with no false positives.



(a) (b) (c)

(d) (e) (f)

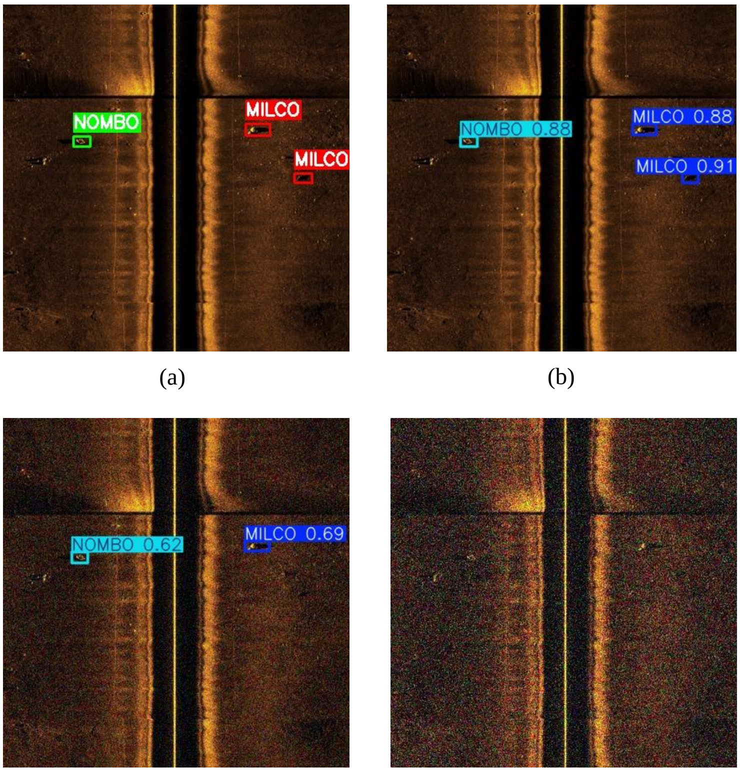
Figure 18. Visual comparison of detection results from different models: (**a**) YOLOv5s; (**b**) YOLOv8s; (**c**) YOLOv9s; (**d**) YOLOv10s; (**e**) YOLOv11s; (**f**) YOLOv11s-SDC.

Due to the limited features and low contrast of small objects in SSS images, especially when small interference objects with similar shapes and echo shadows are present, YOLOv11s-SDC still exhibits instances of misdetection.

5.2. Noise Resistance Capability Analysis

To further evaluate the performance of YOLOv11s-SDC on high-noise SSS images and analyze its noise-resistance capability, we used the SSS image shown in Figure 19a. This image, containing two MILCO objects and one NOMBO object, was modified by applying Gaussian noise of low and medium intensity to simulate real-world conditions. The detection results of YOLOv11s-SDC were then compared under three scenarios: without additional noise, with low-intensity noise, and with medium-intensity noise.

When no additional noise was applied, YOLOv11s-SDC correctly detected and classified all three small objects. After applying low-intensity Gaussian noise, YOLOv11s-SDC was still able to detect and correctly classify one MILCO object and one NOMBO object, but the confidence scores decreased, and one MILCO object was missed. The missed MILCO object was extremely small, with poorly defined bright and shadow regions, making its features nearly indistinguishable after the low-intensity noise was added. When medium-intensity Gaussian noise was applied, YOLOv11s-SDC could no longer detect any small objects.



(a) (b)

(c) (d)

Figure 19. Detection results of YOLOv11s-SDC on SSS images with different levels of noise: (**a**) original image; (**b**) without additional noise; (**c**) with low-intensity noise; (**d**) with medium-intensity noise.

This result indicates that YOLOv11s-SDC has a certain level of noise resistance but is limited in its ability to handle high-noise SSS images, resulting in inadequate detection performance under such conditions.

5.3. Discussion

This section analyzes and compares the visual differences between correctly detected objects, missed objects, and misclassified objects across various versions of YOLO base models and YOLOv11-SDC. It provides an in-depth examination of the performance limitations of YOLOv11-SDC when encountering interference objects that resemble actual objects. Additionally, simulated noise experiments were conducted to evaluate the noise-resistance capability of YOLOv11s-SDC, revealing its detection limitations in high-noise sonar application scenarios.

Due to the challenges of data acquisition, this experiment simulated real high-noise conditions by artificially applying Gaussian noise to SSS images. However, this approach introduces discrepancies compared to real-world data, which also limits the completeness of the analysis.

It is important to note that the quantity and quality of SSS data are major factors limiting model performance. Underwater sonar image collection is both costly and inefficient, resulting in limited datasets for research and analysis [49]. The SIMD dataset used in this study represents five years of effort by its authors; however, the dataset size remains constrained.

6. Conclusions

To overcome challenges in SSS images, such as the similarity between small objects and background textures and the difficulty in feature extraction, this study proposes an enhanced network based on YOLOv11 named YOLOv11-SDC.

Experiments revealed that the newly introduced SF module can automatically extract relevant object features by ranking channel importance and removing low-importance features via channel pruning. This reduces background interference, allowing the model to concentrate more on object-related information in complex scenes. Additionally, this study combined the DRB module with the traditional C3k2 module. By incorporating dilated convolution, the model's ability to capture local features is effectively enhanced. Reparameterization is employed to reduce convolutional kernel parameters, decreasing the model's computational burden. In the YOLOv11-SDC network, a CGAF module is integrated before the original detection module. This module performs joint attention modeling on the spatial, channel, and pixel dimensions of the feature maps, adjusting the weights of each region, channel, and pixel accordingly. As a result, the CGAF module improves the model's ability to recognize and localize objects, especially in SSS images where objects are small, the background is complex, and object contrast is low.

This study used four evaluation metrics: precision, recall, mAP@0.5, and mAP@0.5:0.95. Comparison experiments with YOLOv5, YOLOv8, YOLOv9, YOLOv10, and YOLOv11 demonstrated that YOLOv11-SDC outperforms all existing baseline models across these metrics. When replacing the newly proposed SPPF module in YOLOv11 with the SF module introduced in this study, the YOLOv11s-SF network shows improvements of 9.1%, 2.8%, 0.8%, and 2% in precision, recall, mAP@0.5, and mAP@0.5:0.95, respectively, compared to the YOLOv11s network on the SIMD dataset, confirming the effectiveness of the SF module. Ablation experiments further validated that the YOLOv11-SDC network achieves superior detection performance, highlighting its advantages.

Finally, a qualitative analysis was conducted to explore the performance limitations of YOLOv11-SDC when dealing with interference objects that resemble target objects. The simulated noise experiments further indicated that the detection performance of YOLOv11-SDC is influenced by SSS imaging quality, suggesting room for improvement in its noise resistance. In future work, we will focus on acquiring datasets with diverse small objects and varying SSS image qualities to further optimize the proposed YOLOv11-SDC model.

Author Contributions: Conceptualization, C.Z. and X.X.; methodology, C.Z.; software, C.Z.; validation, C.Z.; resources, S.Y., Y.Y., and H.G.; writing—original draft preparation, C.Z.; writing—review and editing, S.Y., Y.Y., and H.G.; funding acquisition, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 62206274) and the Independent Project Deployed by the Innovative Academy of Marine Information Technology, Chinese Academy of Sciences (Grant No. CXBS202102).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset and code used in this study are publicly available at the following URLs: dataset: https://figshare.com/articles/dataset/i_Side-scan_sonar_imaging_for_Mine_detection_i/24574879 (accessed on 28 October 2024); code: https://github.com/DevDynamoX/YOLOv11-SDC (accessed on 5 January 2025).

Acknowledgments: The authors greatly appreciate the constructive comments of the reviewers and editor.

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

References

1. Wibisono, A.; Piran, M.J.; Song, H.-K.; Lee, B.M. A survey on unmanned underwater vehicles: Challenges, enabling technologies, and future research directions. *Sensors* **2023**, *23*, 7321. [CrossRef] [PubMed]
2. Jian, M.; Yang, N.; Tao, C.; Zhi, H.; Luo, H. Underwater object detection and datasets: A survey. *Intell. Mar. Technol. Syst.* **2024**, *2*, 9. [CrossRef]
3. Tarekegn, A.N.; Cheikh, F.A.; Ullah, M.; Sollesnes, E.T.; Alexandru, C.; Azar, S.N.; Erol, E.; Suciu, G. Underwater Object Detection using Image Enhancement and Deep Learning Models. In Proceedings of the 2023 11th European Workshop on Visual Information Processing (EUVIP), Gjovik, Norway, 11–14 September 2023; pp. 1–6.
4. Zhang, M.; Wang, Z.; Song, W.; Zhao, D.; Zhao, H. Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network. *Appl. Sci.* **2024**, *14*, 1095. [CrossRef]
5. Wei, S.; Leung, H.; Myers, V. An automated change detection approach for mine recognition using sidescan sonar data. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 553–558.
6. Tang, Y.; Wang, L.; Li, H.; Bian, S. Side-scan sonar underwater target segmentation using the BHP-UNet. *Eurasip J. Adv. Sig. Pr.* **2023**, *2023*, 76. [CrossRef]
7. Sinai, A.; Amar, A.; Gilboa, G. Mine-Like Objects detection in Side-Scan Sonar images using a shadows-highlights geometrical features space. In Proceedings of the OCEANS 2016 MTS/IEEE Monterey, Monterey, CA, USA, 19–23 September 2016; pp. 1–6.
8. Çelebi, A.T.; Güllü, M.K.; Ertürk, S. Mine detection in side scan sonar images using Markov Random Fields with brightness compensation. In Proceedings of the 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 20–22 April 2011; pp. 916–919.
9. Barngrover, C.M. *Automated Detection of Mine-Like Objects in Side Scan Sonar Imagery*; University of California: San Diego, CA, USA, 2014.
10. Hożyń, S. A review of underwater mine detection and classification in sonar imagery. *Electronics* **2021**, *10*, 2943. [CrossRef]
11. Ye, X.; Li, C.; Zhang, S.; Yang, P.; Li, X. Research on side-scan sonar image target classification method based on transfer learning. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–6.
12. Sivachandra, K.; Kumudham, R. A Review: Object Detection and Classification Using Side Scan Sonar Images via Deep Learning Techniques. In *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 229–249.
13. Wang, X.; Wang, L.; Li, G.; Xie, X. A robust and fast method for sidescan sonar image segmentation based on region growing. *Sensors* **2021**, *21*, 6960. [CrossRef]
14. Najibzadeh, M.; Mahmoodzadeh, A.; Khishe, M. Active sonar image classification using deep convolutional neural network evolved by robust comprehensive grey wolf optimizer. *Neural Process. Lett.* **2023**, *55*, 8689–8712. [CrossRef]
15. Abu, A.; Diamant, R. Underwater object classification combining SAS and transferred optical-to-SAS Imagery. *Pattern. Recogn.* **2023**, *144*, 109868. [CrossRef]
16. Abu, A.; Diamant, R. A Statistically-Based Method for the Detection of Underwater Objects in Sonar Imagery. *IEEE Sens. J.* **2019**, *19*, 6858–6871. [CrossRef]
17. Lopera, O.; Dupont, Y. Automated target recognition with SAS: Shadow and highlight-based classification. In Proceedings of the 2012 Oceans, Hampton Roads, VA, USA, 14–19 October 2012; pp. 1–5.
18. Grasso, R.; Spina, F. Small bottom object density analysis from side scan sonar data by a mathematical morphology detector. In Proceedings of the 2006 9th International Conference on Information Fusion, Florence, Italy, 10–13 July 2006; pp. 1–8.
19. Wang, J.; Li, H.; Huo, G.; Li, C.; Wei, Y. Multi-modal multi-stage underwater side-scan sonar target recognition based on synthetic images. *Remote Sens.* **2023**, *15*, 1303. [CrossRef]
20. Munteanu, D.; Moina, D.; Zamfir, C.G.; Petrea, Ş.M.; Cristea, D.S.; Munteanu, N. Sea mine detection framework using YOLO, SSD and EfficientDet deep learning models. *Sensors* **2022**, *22*, 9536. [CrossRef] [PubMed]
21. Li, J.; Cao, X. Target recognition and detection in side-scan sonar images based on YOLO v3 model. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 July 2022; pp. 7186–7190.
22. Einsidler, D.; Dhanak, M.; Beaujean, P.-P. A deep learning approach to target recognition in side-scan sonar imagery. In Proceedings of the OCEANS 2018 MTS/IEEE Charleston, Charleston, SC, USA, 22–25 October 2018; pp. 1–4.
23. Yamada, M.; Cartmill, J.; Azimi-Sadjadi, M.R. Buried underwater target classification using the new BOSS and canonical correlation decomposition feature extraction. In Proceedings of the OCEANS 2005 MTS/IEEE, Washington, DC, USA, 17–23 September 2005; pp. 589–596.
24. Zhu, P.; Isaacs, J.; Fu, B.; Ferrari, S. Deep learning feature extraction for target recognition and classification in underwater sonar images. In Proceedings of the 2017 IEEE 56th annual conference on decision and control (CDC), Melbourne, VIC, Australia, 12–15 December 2017; pp. 2724–2731.

25. Ding, X.; Zhang, Y.; Ge, Y.; Zhao, S.; Song, L.; Yue, X.; Shan, Y. UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 5513–5524.
26. Chen, Z.; He, Z.; Lu, Z.-m. DEA-Net: Single Image Dehazing Based on Detail-Enhanced Convolution and Content-Guided Attention. *IEEE Trans. Image Process.* **2023**, *33*, 1002–1015. [CrossRef]
27. Fu, S.; Xu, F.; Liu, J.; Pang, Y.; Yang, J. Underwater small object detection in side-scan sonar images based on improved YOLOv5. In Proceedings of the 2022 3rd International Conference on Geology, Mapping and Remote Sensing (ICGMRS), Zhoushan, China, 22–24 April 2022; pp. 446–453.
28. Zhang, F.; Zhang, W.; Cheng, C.; Hou, X.; Cao, C. Detection of small objects in side-scan sonar images using an enhanced YOLOv7-based approach. *J. Mar. Sci. Eng.* **2023**, *11*, 2155. [CrossRef]
29. Tang, R.; Chen, Y.; Gao, J.; Hao, S.; He, H. Underwater Target Detection Using Side-Scan Sonar Images Based on Upsampling and Downsampling. *Electronics* **2024**, *13*, 3874. [CrossRef]
30. Santos, N.P.; Moura, R.; Torgal, G.S.; Lobo, V.; de Castro Neto, M. Side-scan sonar imaging data of underwater vehicles for mine detection. *Data Brief* **2024**, *53*, 110132. [CrossRef]
31. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:17725.
32. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8, version 8.0.0. Available online: https://github.com/ultralytics/ultralytics (accessed on 15 November 2024).
33. Tang, H.; Liang, S.; Yao, D.; Qiao, Y. A visual defect detection for optics lens based on the YOLOv5-C3CA-SPPF network model. *Opt. Express* **2023**, *31*, 2628–2643. [CrossRef] [PubMed]
34. Hsiao, T.-Y.; Chang, Y.-C.; Chou, H.-H.; Chiu, C.-T. Filter-based deep-compression with global average pooling for convolutional networks. *J. Syst. Archit.* **2019**, *95*, 9–18. [CrossRef]
35. Kumar, R.L.; Kakarla, J.; Isunuri, B.V.; Singh, M. Multi-class brain tumor classification using residual network and global average pooling. *Multimed. Tools Appl.* **2021**, *80*, 13429–13438. [CrossRef]
36. Li, Z.; Wang, S.H.; Fan, R.R.; Cao, G.; Zhang, Y.D.; Guo, T. Teeth category classification via seven-layer deep convolutional neural network with max pooling and global average pooling. *Int. J. Imag. Syst. Technol.* **2019**, *29*, 577–583. [CrossRef]
37. Wang, Z.; Ji, S. Smoothed dilated convolutions for improved dense prediction. In Proceedings of the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 2486–2495.
38. Salimans, T.; Kingma, D.P. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 901–909.
39. Luo, W.; Li, Y.; Urtasun, R.; Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 4898–4906.
40. Bjorck, N.; Gomes, C.P.; Selman, B.; Weinberger, K.Q. Understanding batch normalization. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7694–7705.
41. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In Proceedings of the International Conference on Artificial Neural Networks, Thessaloniki, Greece, 15–18 September 2010; pp. 92–101.
42. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020; pp. 1–7.
43. Han, J.; Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In Proceedings of the International Workshop on Artificial Neural Networks, Torremolinos, Spain, 7–9 June 1995; pp. 195–201.
44. Kang, Z.; Ma, F.; Chen, C.; Sun, J. YOSMR: A Ship Detection Method for Marine Radar Based on Customized Lightweight Convolutional Networks. *J. Mar. Sci. Eng.* **2024**, *12*, 1316. [CrossRef]
45. Zhang, F.; Cao, W.; Gao, J.; Liu, S.; Li, C.; Song, K.; Wang, H. Underwater Object Detection Algorithm Based on an Improved YOLOv8. *J. Mar. Sci. Eng.* **2024**, *12*, 1991. [CrossRef]
46. Li, Q.; Shi, H. YOLO-GE: An Attention Fusion Enhanced Underwater Object Detection Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 1885. [CrossRef]
47. Wang, P.; Yang, S.; Chen, G.; Wang, W.; Huang, Z.; Jiang, Y. A Ship's Maritime Critical Target Identification Method Based on Lightweight and Triple Attention Mechanisms. *J. Mar. Sci. Eng.* **2024**, *12*, 1839. [CrossRef]
48. Chen, Z.; Xie, G.; Deng, X.; Peng, J.; Qiu, H. DA-YOLOv7: A Deep Learning-Driven High-Performance Underwater Sonar Image Target Recognition Model. *J. Mar. Sci. Eng.* **2024**, *12*, 1606. [CrossRef]
49. Li, L.; Li, Y.; Wang, H.; Yue, C.; Gao, P.; Wang, Y.; Feng, X. Side-Scan Sonar Image Generation Under Zero and Few Samples for Underwater Target Detection. *Remote Sens.* **2024**, *16*, 4134. [CrossRef]

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.