Journal of
Marine Science and Engineering

MDPI

Article

DBnet: A Lightweight Dual-Backbone Target Detection Model Based on Side-Scan Sonar Images

Quanhong Ma , Shaohua Jin *, Gang Bian, Yang Cui and Guoqing Liu

Department of Oceanography and Hydrography, Dalian Naval Academy, Dalian 116018, China
* Correspondence: jsh_1978@163.com

Abstract: Due to the large number of parameters and high computational complexity of current target detection models, it is challenging to perform fast and accurate target detection in side-scan sonar images under the existing technical conditions, especially in environments with limited computational resources. Moreover, since the original waterfall map of side-scan sonar only consists of echo intensity information, which is usually of a large size, it is difficult to fuse it with other multi-source information, which limits the detection accuracy of models. To address these issues, we designed DBnet, a lightweight target detector featuring two lightweight backbone networks (PP-LCNet and GhostNet) and a streamlined neck structure for feature extraction and fusion. To solve the problem of unbalanced aspect ratios in sonar data waterfall maps, DBnet employs the SAHI algorithm with sliding-window slicing inference to improve small-target detection accuracy. Compared with the baseline model, DBnet has 33% fewer parameters and 31% fewer GFLOPs while maintaining accuracy. Tests performed on two datasets (SSUTD and SCTD) showed that the mAP values improved by 2.3% and 6.6%.

Keywords: SSS; deep learning; lightweight network; DBnet

check for updates

Academic Editors: Lin Li, Weicheng Cui, Dongmei Chen, Po Hu and Yuanzhi Zhang

Received: 28 November 2024
Revised: 4 January 2025
Accepted: 16 January 2025
Published: 17 January 2025

Citation: Ma, Q.; Jin, S.; Bian, G.; Cui, Y.; Liu, G. DBnet: A Lightweight Dual-Backbone Target Detection Model Based on Side-Scan Sonar Images. *J. Mar. Sci. Eng.* **2025**, *13*, 155. https://doi.org/10.3390/jmse13010155

1. Introduction

The detection and identification of seafloor targets play an extremely important role in underwater search and rescue, marine engineering construction, marine topography and geomorphology measurements, marine resource investigation, etc. However, affected by the complex marine environment, imaging conditions, and measurement means, accuracy and efficiency are relatively low, which makes it difficult to meet the demand; so, improving these aspects has become a crucial research hotspot [1–6]. Side-scan sonar has gained widespread application in seabed target detection due to its affordability, rapid coverage of large areas, and its independence from underwater visibility, as it relies instead on acoustic imaging. This technology is crucial to locate the remains of aircraft, ships, and individuals; pinpoint underwater pipelines; and detect submerged reefs, ores, and mines [7–10]. Nevertheless, the current method of identifying underwater targets in side-scan sonar imagery relies heavily on manual inspection, which is not only subjective and time-consuming but also hinders broader adoption, particularly in scenarios requiring real-time detection, such as those featuring autonomous underwater vehicles (AUVs) [11].

With the rapid advancement of computer vision technology and the development of convolutional neural networks, deep learning approaches have become prevalent in the task of seabed target recognition in side-scan sonar images, offering substantial improvements in accuracy and efficiency over traditional methods [12–15]. Deep learning algorithms for target detection are broadly categorized into two types: two-stage and one-stage models. However, in practice, because it is necessary to generate a regional suggestion network of candidate target frames and then perform their classification and bounding box regression to achieve target detection with the former, there are problems such as complex structure, a significantly large computational volume, and long computation time; therefore, these models are seldom used in practical engineering applications [16]. The core advantage of single-stage target detection models is, firstly, their speed, which allows for efficient and instantaneous detection on account of the simplification of the detection process and a reduction in the consumption of computational resources, which makes these models especially suitable for application scenarios with high real-time requirements. Secondly, their structure is relatively simple and easy to implement and deploy, and the number of hyperparameters that need to be adjusted is reduced, in turn reducing the complexity of model tuning [17]. In terms of detection accuracy, single-stage models tend to demonstrate superior performance in specific scenarios, such as small- and dense-target detection. In addition, single-stage models demonstrate great adaptability and flexibility, can be easily adapted to different task requirements, and are easy to integrate with other image-processing systems or platforms.

Although generic target detection models have achieved certain results in the field of natural or remote sensing imagery, target detection based on side-scan sonar images still presents multiple challenges [18–20].

(1) Side-scan sonar usually relies on mobile platforms such as AUVs to implement detection, and in current AUV seabed obstacle detection missions, traditional intelligent detection models often present deployment difficulties, high power consumption, and slow processing speed due to the large number of parameters and high computational complexity [21,22]. These deficiencies limit the real-time performance and efficiency of AUVs in complex marine environments and increase energy consumption, which affects the operating time and stability of AUVs. Therefore, research on lightweight intelligent detection models becomes particularly important.

(2) Although AUVs are equipped with various detection devices, such as forward-looking sonar, side-scan sonar, and optical cameras [23,24], in some specific mission scenarios, in order to perform detection over as large an area of the seafloor as possible in a short period of time, side-scan sonar with a wide detection range is typically used, which means that the poor features of sparse targets need to be extracted from side-scan sonar images to perform seafloor target detection. Therefore, determining how to make full use of side-scan sonar images to extract more feature information is the key to improving the accuracy of undersea target detection models.

(3) Side-scan sonar continuously sends and receives acoustic signals during AUV traveling, and the collected data are processed and superimposed to form a waterfall image, which means that the original image of the side-scan sonar is usually large in size; a large amount of fine-grained information is lost if the whole image is directly inputted into the network, which also leads to high leakage of small undersea targets (targets with less than 50 pixels × 50 pixels in the image) [25]. Therefore, how to optimize the detection strategy is also an important aspect in solving the problem of real-time undersea target detection based on side-scan sonar.

Aiming at solving the above problems, in this study, we developed a new target detection model, DBnet.

(1) We adopted two kinds of lightweight backbone networks and optimized the neck part of the baseline model. By streamlining the structure, the model in this study presents significantly fewer parameters and less computation while maintaining detection accuracy, achieving a balance between the latter and speed, and meeting the needs of practical engineering applications.

(2) To address the challenge of feature extraction with less valid information in side-scan sonar images, in this study, we developed a dual-backbone network structure. The structure makes use of multiple feature extraction paths to simulate multimodal data fusion so that even if only side-scan sonar images are used as input, an effect similar to that of multimodal data fusion can be achieved, thus improving the diversity of feature extraction and the performance of the detection model.

(3) In order to solve the problem of the large size of the original waterfall map in side-scan sonar, in this study, we adopted the slice-assisted hyper-inference (SAHI) technique, which splits large-size images into multiple small-size images, performs network inference separately, and fuses the detection results of each slice.

2. Related Work

In the field of target detection in side-scan sonar images, the study of lightweight target detection models has gradually become a hotspot. Li [26] significantly improved detection speed and accuracy by replacing the backbone network of YOLO v8s with the GhostNet structure. Moreover, the lightweight attention mechanism Triplet Attention was introduced to optimize feature extraction, and the ECIoU loss function was employed to improve the convergence and recognition accuracy of the model. Yu [11] proposed a real-time automatic target recognition method (TR-YOLOv5s) combining the Transformer module and YOLOv5s to address the problems of target-sparse and feature-poor side-scan sonar images. By introducing an attention mechanism, the focus on target features is enhanced, which improves detection accuracy and efficiency. Zhang et al. [27] combined the Swin Transformer with the YOLO framework for marine target detection. This method allows for the extraction of discriminative features under ocean clutter interference, a reduction in computational complexity, and an improvement in target detection accuracy. Huang [28] employed the Dual Segmented Attention (DSA) mechanism, which efficiently extracts target features through the parallel processing of channel and spatial attention and enhances the ability to extract features with weak boundaries. Li [29] combined Spatial Pyramid Pooling (SPP) and Online Dataset Preprocessing (ODP) for underwater target detection in side-scan sonar images. The method overcomes the input image size limitation and improves the feature extraction capability with SPP, while the diversity and complexity of the dataset is enhanced with ODP, thus improving detection accuracy and efficiency. Ji et al. [30] introduced YOLO-TLA, a lightweight model which includes an extra layer specifically designed for detecting small targets and incorporates the C3CrossCovn module with a global attention mechanism in its backbone network. This design reduces technical complexity and parameter count while enhancing the focus on relevant object attributes and filtering out unnecessary information. Zhang [31] and others applied a combination of the lightweight backbone network Mobile v2 and deep separable convolution on top of the YOLOv4 algorithm, which significantly reduces the number of model parameters, and introduced an attention mechanism in the FPN to learn richer features of small targets. Tang et al. [32] presented a multi-scale sensory field convolutional block attention mechanism, known as AMMRF, which leverages feature map position information to precisely capture inter-channel relationships and enhance the learning of ship–background interactions. In their YOLO-SARSI model, they integrated the AMMRF module within the backbone network for feature fusion, simplifying the baseline model's complex components. This approach effectively decreased the number of parameters and computational load. The intricate feature fusion component of the baseline model was omitted, resulting in a significant reduction in both parameter count and computational complexity.

3. Materials and Methods

The operation flow chart, including an AUV carrying side-scan sonar and the proposed algorithm for the actual measurement process, is shown in Figure 1, and mainly includes the data acquisition and collation part and the use of our algorithm for real-time target detection, performed in two steps. The side-scan sonar data collected by the AUV are spliced into a map; then, the image is sliced and processed with SAHI and inputted into DBnet for the detection of undersea obstacles. Once the AUV detection mission is over, DBnet can be retrained by using the database constructed from the detected target data and the existing data, expanding the sample size in order to improve the detection accuracy of the model.

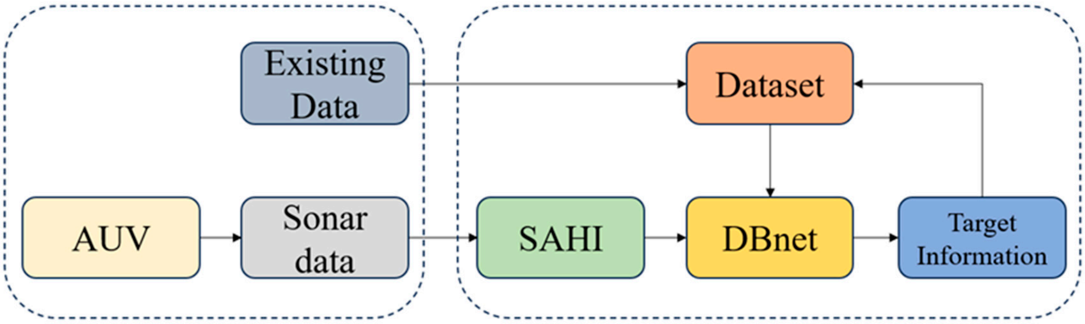


Figure 1. Operation flow chart.

3.1. DBnet

The DBnet detector presented here is implemented within the YOLOv8 network architecture. The YOLOv8 model consists of four primary components, i.e., input, backbone, neck, and output, each incorporating various modules, such as the Conv module, the C2f module, and the SPPF module [33]. One of the innovations in YOLOv8 is the introduction of the C2f structure, which is pivotal for residual feature learning and allows for the efficient capture of gradient flow information. The backbone, comprising five Conv modules, four C2f structures, and one SPPF structure, is responsible for extracting generic target features. The SPPF structure, located in the final layer of the backbone, collects information from sensory fields of different sizes (5, 9, and 13) through a series of consecutive 5 × 5 convolutional kernel max-pooling operations. These feature layers are then combined with their unprocessed counterparts to integrate multi-scale feature information and enhance model performance. The neck segment, positioned between the backbone and the prediction component, is designed to diversify features and bolster model robustness. It incorporates four C2f modules, two Conv modules, and two Upsample operations. Finally, the prediction component serves as the output end of the model and is responsible for delivering the final target detection results. In essence, the neck enriches feature diversity and improves model robustness by incorporating various modules, while the prediction component is responsible for the output of the detection results.

In this study, we developed the DBnet model to address the problems of existing models, including the difficulty in extracting effective information from feature-poor and target-sparse side-scan sonar images, the challenging deployment of these models on mobile platforms with limited computing power such as AUVs, low detection efficiency, significant target leakage, etc. For our model, we designed a parallel lightweight two-branch backbone structure, streamlined the original neck part to achieve high efficiency with a simple structure, and developed the SAHI algorithm for the characteristics of the original waterfall map of side-scan sonar, which greatly improves the accuracy of small target detection, as well as the efficiency. The network structure is shown in Figure 2.

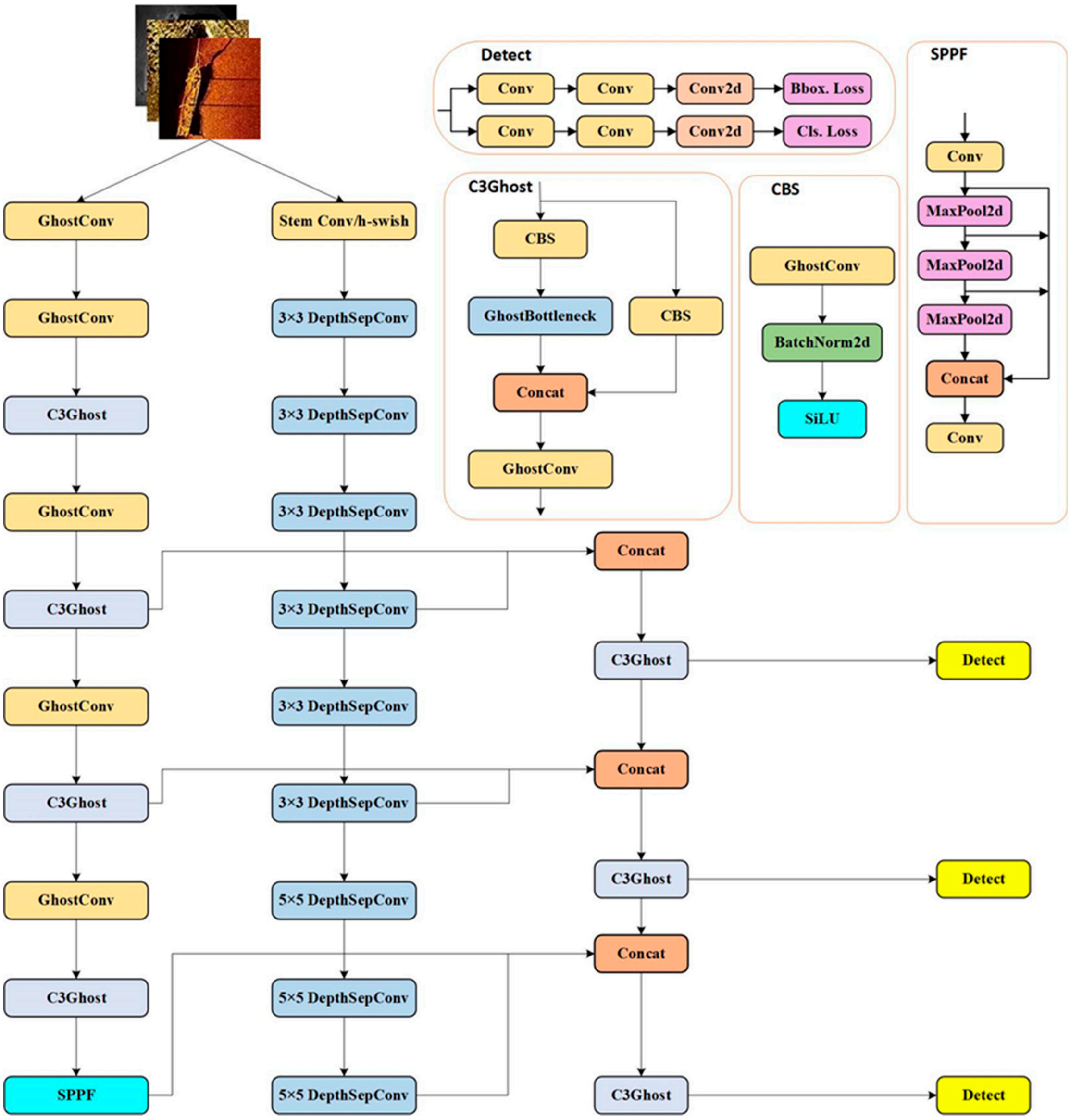


Figure 2. Diagram showing the DBnet model's structure details.

Firstly, we chose PP-LCNet and GhostNet to form the dual backbone of the proposed model, which enables it to fuse the feature information extracted from different backbone networks in the case of the availability of only one data mode. Generally speaking, using different backbone networks allows for the extraction of complementary feature information, and by fusing these features, more comprehensive target information can be captured, thus improving the detection accuracy of the model. Data from a single source are susceptible to noise, occlusion, etc.; the fusion of similar multimodal data can enhance the robustness of the model based on the redundant information among features extracted from different backbones. When the data extracted from one backbone are subject to disturbance, the data from the other backbone can still provide effective information support; in addition, by extracting features through the dual-backbone network, the model can learn more generalized feature representations, which can lead to better performance in different background environments. Specifically, as the first feature extraction backbone, we used GhostNet, which is constructed based on the Ghost module and consists of 6 layers of GhostConv and 4 layers of C3Ghost with SPPF modules. Then, we adopted Depth-Separable Convolution

(DepthSepConv), which is composed of consecutive six layers of 3 × 3 DepthSepConv and three layers of 5 × 5 DepthSepConv.

In the neck segment, the original structure's multiple upsampling steps often introduce adverse effects, such as noise amplification. Therefore, we refrained from using upsampling in the standard YOLOv8 model to merge small-scale high-level convolutional features with large-scale low-level ones. Instead, we optimized the neck part by leveraging sufficiently rich feature maps from dual-stem feature extraction for fusion. This approach reduces the model's computational load, simplifies its complexity, and contributes to a lighter model. After feature extraction, the 5th, 7th and 10th layers of each trunk are extracted for their input into the neck section. In the neck part, the feature maps extracted from each layer are simply fused. It is worth mentioning that we not only streamlined the structure but also used C3Ghost, instead of the C2f module in the original YOLOv8 network, to further reduce the parameters and computation of the model.

To enhance robustness against inference, we employed the SAHI algorithm, which augments the feature information of small targets by leveraging data from image slices. This capability makes SAHI highly adept at detecting small targets, which often occupy a limited number of pixels in an image and lack the necessary detail for traditional detection methods to function effectively. SAHI effectively boosts the feature representation of small targets with slicing and weighted fusion techniques, leading to enhanced detection accuracy. The SAHI algorithm slices the waterfall map formed based on side-scan sonar data acquisition into multiple slices, on which it performs target detection independently. This parallel processing can significantly improve detection efficiency, especially when processing side-scan sonar images or in scenarios with high computational resource requirements. Moreover, by focusing on smaller slices, the SAHI algorithm can reduce the consumption of memory and computational resources. This is especially important for devices with limited computational resources or real-time detection systems.

3.2. SAHI

Distinct from the real-time segmentation methods referenced in [34], target detection necessitates detailed attributes such as the size, shape, and spatial positioning of underwater targets for their precise location and identification. To ensure an underwater target is not divided between images, overlapping coverage between adjacent samples is crucial. For real-time detection, this entails intensive sampling along the track. Each sample undergoes processing with a d × d pixel sliding window, which creates sections of the same size as it moves horizontally along the track intersection. To achieve precise contours and positions through image segmentation or subsequent processing, adjacent blocks (such as P1 and P2 in Figure 3) share a common coverage area. However, an excessively high overlap rate increases the slice count and prolongs inference time, while too low an overlap may result in incomplete target representation. Setting the overlap rate to 20%, meaning the shared area is 20% of the slice size, ensures that the target remains intact and fulfills real-time processing demands. Although using this method can improve accuracy in small-target detection, there is also a need to balance model accuracy and inference time to determine the scale of the sliding slices.

The specific principle is shown in Figure 4, where the original image, I (blue box), is first cut into M × N slices (red box) with a certain overlap, denoted by P1, P2. . . . . .PX; then, each slice is resized while maintaining the aspect ratio. Afterwards, the content in each slice is predicted. As the slice size becomes smaller, the model's detection performance on larger targets decreases. Therefore, in order to detect the latter more accurately, NMS combines the prediction results of the slices and the FI results of the original image, bringing them back to the original size; in the NMS process, the frames with an IoU ratio higher than the pre-set matching threshold (Tm) are matched, and the probability of detection for each match below Td is removed.

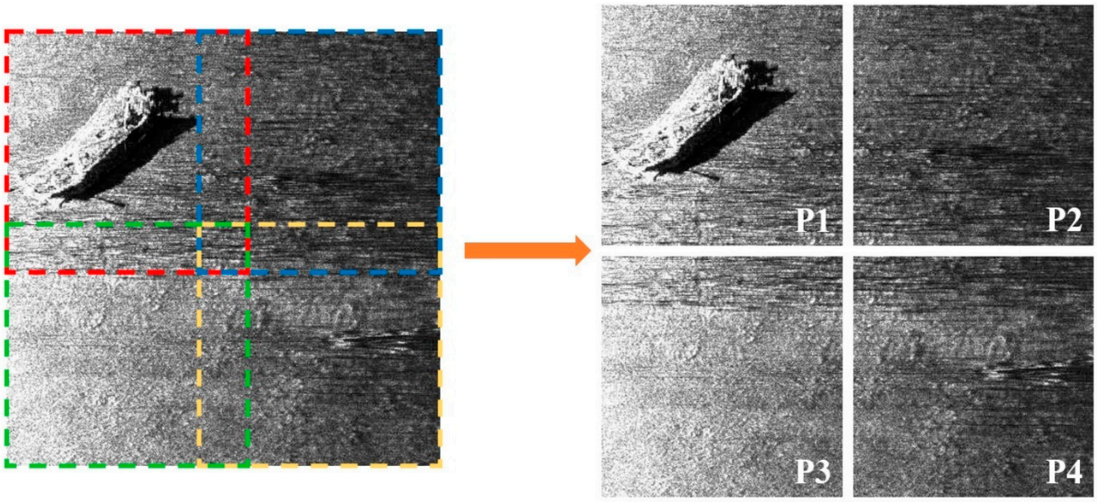


Figure 3. A schematic of the slices generated with SAHI in a sample. The colored dashed boxes indicate the four neighboring slices P1, P2, P3, and P4 corresponding to when X = 4, with a size of d × d pixels.

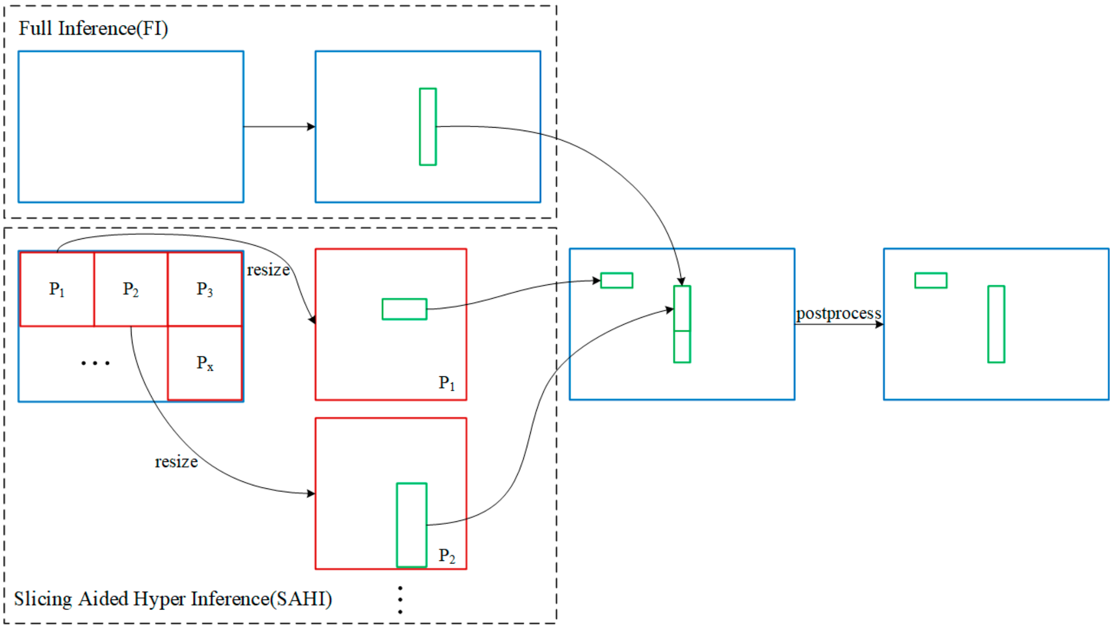


Figure 4. Schematic of SAHI principle.The blue border is the whole image, the red border represents the corresponding slice, and the green border is the detection result.

3.3. PP-LCNet

PP-LCNet is a lightweight convolutional neural network model with the advantages of high efficiency, low latency, and low computational cost [35]. It outperforms other lightweight models in multiple tasks and enhances detection accuracy while working with the fast speed of the MobileNet [36,37] network, which is more suitable for embedded devices and mobile application scenarios.

The YOLOv8 network architecture includes a Darknet-53 backbone module with a deeper structure, enhancing the model's capacity to represent image features. However, this depth leads to increased computational complexity, resulting in longer durations for both model training and inference. The overall structure of PP-LCNet is shown in Figure 5.

Within the module, the stem component employs 3 × 3 standard convolution for feature extraction, primarily targeting the extraction of low-level features from the input image. The depth-separable convolution operations, comprising depth-wise (DW) and point-wise (PW) convolution operations, serve to decrease the number of model parameters and reduce network computation. PP-LCNet, which utilizes locally connected blocks to build an efficient deep neural network, employs DepthSepConv, proposed in MobileNetV1 [14], as the basic module to reduce the computational complexity and improve the generalization ability of the network. This module lacks operations such as shortcuts, thereby eliminating the need for additional operations, such as concatenation or element-wise addition, which hinder the model's inference speed without enhancing accuracy, particularly in smaller models. Furthermore, prior research has indicated that mixing convolutional kernel sizes within the same network layer slows down inference. Therefore, we utilized a uniform kernel size per layer, opting for a larger kernel that balances low latency with high accuracy. Notably, it was discovered that substituting the 3 × 3 kernel with a 5 × 5 kernel only in the network's tail achieved nearly equivalent benefits as replacing kernels throughout the entire network, prompting this substitution only in the tail section. Moreover, the SE module was added to the module at the tail of the network; it dynamically adjusts the importance of different channels in the network by introducing an attention mechanism to increase the model's attention to important features, thus enhancing the salient features, suppressing unimportant features, and improving the discriminative ability. In order to improve the inference speed of the network, the activation function in the convolution module adopts Hard-Swish, an approximation of the Swish function, while the Sigmoid function in the SE module is replaced with the Hard-Sigmoid function, which is less computationally intensive, in order to avoid a large number of exponential operations to improve the computational speed.

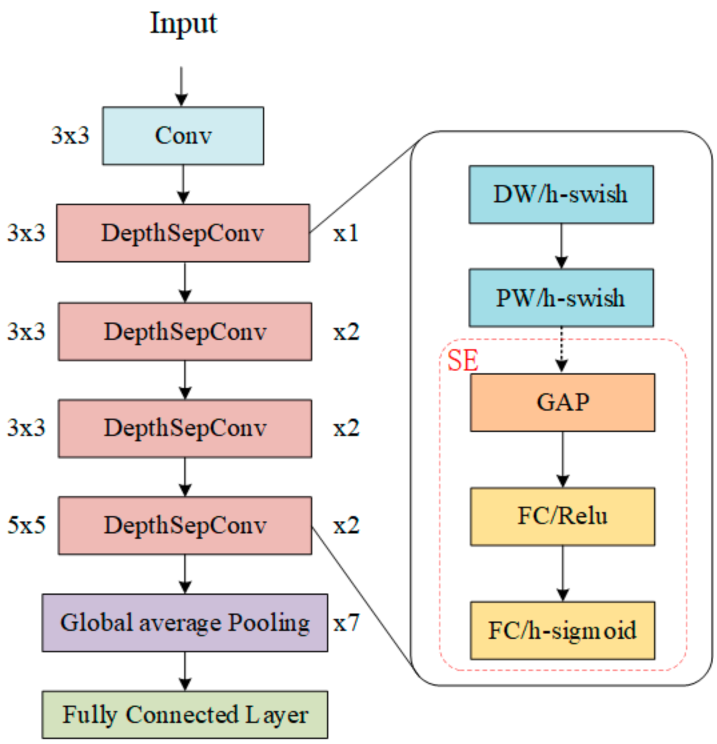


Figure 5. Schematic diagram of PP-LCNet's structure. Conv is a standard 3 × 3 convolution. DepthSepConv denotes depth-separable convolution, where DW denotes depth-wise convolution and PW denotes point-wise convolution. Moreover, SE denotes the Squeeze-and-Excitation module.

3.4. GhostNet

The GhostNet network [38] employs a streamlined design centered around multiple Ghost bottlenecks. Each Ghost bottleneck is constructed by using the Ghost module, which hinges on dividing the convolution operation into two stages. The first stage involves performing a limited number of standard convolution operations. The second stage generates “ghost” feature maps by applying cheap linear convolution operations to the feature maps obtained from the first stage. The first part is a small number of ordinary convolution operations; the second part is a chunked linear convolution operation of the feature maps obtained in the first part, which generates “phantom” feature maps at a small cost. Compared with a normal convolutional neural network, the total number of parameters required and the computational complexity of the Ghost module are reduced, while the output feature maps are of the same size.

The core idea of the Ghost module is to utilize ordinary linear variations to obtain redundant feature maps as a way to improve the computational efficiency of the network. Figure 6a shows the traditional convolutional structure, while the phantom convolution uses depth-wise convolution as a cheaper linear transformation, as shown in Figure 6b, where ϕ denotes the linear transformation. This structure makes the current channel feature relevant only to itself, simulating redundant features on the one hand, and significantly reducing the number of parameters and computation on the other. In differs from regular convolution, which directly produces all feature maps, as GhostConv first executes a convolution operation that yields fewer feature maps. Subsequently, it applies a convolution transformation to these initial feature maps to produce both constant mapping and additional feature maps.

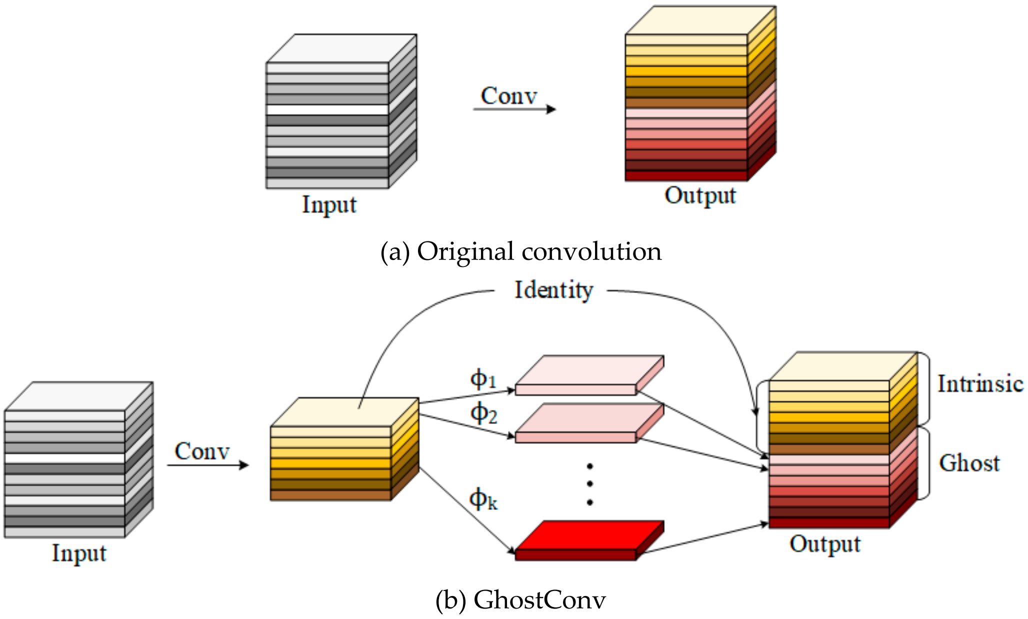


Figure 6. GhostConv operation principle.

This method efficiently decreases both the computational load and the number of parameters, as illustrated by the following comparison with the standard approach.

Let us consider an input data tensor with dimensions C × H × W, which represent the input channels, height, and width of the feature map, respectively. Once a convolution operation is executed, the resulting data tensor features dimensions of $N \times H' \times W'$, which represent the number of output channels, and the height and width of the produced feature map, respectively. Considering a typical convolutional kernel size of K and a linearly transformed convolutional kernel size of D, after S transformations, r_s in Equation (1) is the speedup ratio (here, the computational volume is used as an approximation instead of speedup), which represents the ratio of the number of original convolution operations to the number of computations of the Ghost module. r_c in Equation (2) is the compression ratio, representing the ratio of the number of parameters of the original convolution operation to the number of parameters of the Ghost module:

$$\begin{aligned} r_s &= \frac{N \cdot H' \cdot W' \cdot C \cdot K \cdot K}{\frac{N}{S} \cdot H' \cdot W' \cdot C \cdot K \cdot K + (S-1) \cdot \frac{N}{S} \cdot H' \cdot W' \cdot D \cdot D} \\ &= \frac{C \cdot K \cdot K}{\frac{1}{S} \cdot C \cdot K \cdot K + \frac{S-1}{S} \cdot D \cdot D} \approx \frac{S \cdot C}{S + C - 1} \approx S \end{aligned} \tag{1}$$

$$r_c = \frac{N \cdot C \cdot K \cdot K}{\frac{N}{S} \cdot C \cdot K \cdot K + (S-1) \cdot \frac{N}{S} \cdot D \cdot D} \approx \frac{S \cdot C}{S + C - 1} \approx S \tag{2}$$

Above, N/S is the output channel taught in the first transformation, and S − 1 is included because constant mapping does not need to be computed but counts as part of the second transformation. Therefore, the Ghost module saves computation.

Ghost bottlenecks are bottleneck structures that incorporate Ghost modules as shown in Figure 7. Each Ghost bottleneck comprises two stacked Ghost modules: the first one serves to expand the number of channels (known as the expansion layer), with the ratio between the output and input channel counts defined as the expansion ratio. The second Ghost module then decreases the channel count to align with the channels in the shortcut branch.

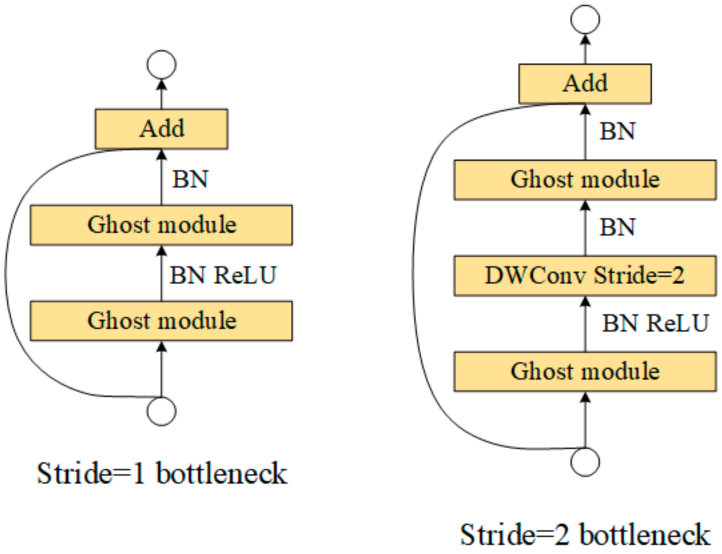


Figure 7. Ghost bottleneck.

To reduce the width and height of the feature layer, we configured the Ghost bottlenecks with a stride of 2, indicating a step size of 2. In this scenario, additional convolutional layers are included within the bottlenecks. Furthermore, in the main part of the bottlenecks, both Ghost modules incorporate a depth-separable convolution operation with a 2 × 2 stride to achieve significant compression in both the width and height of the feature layer.

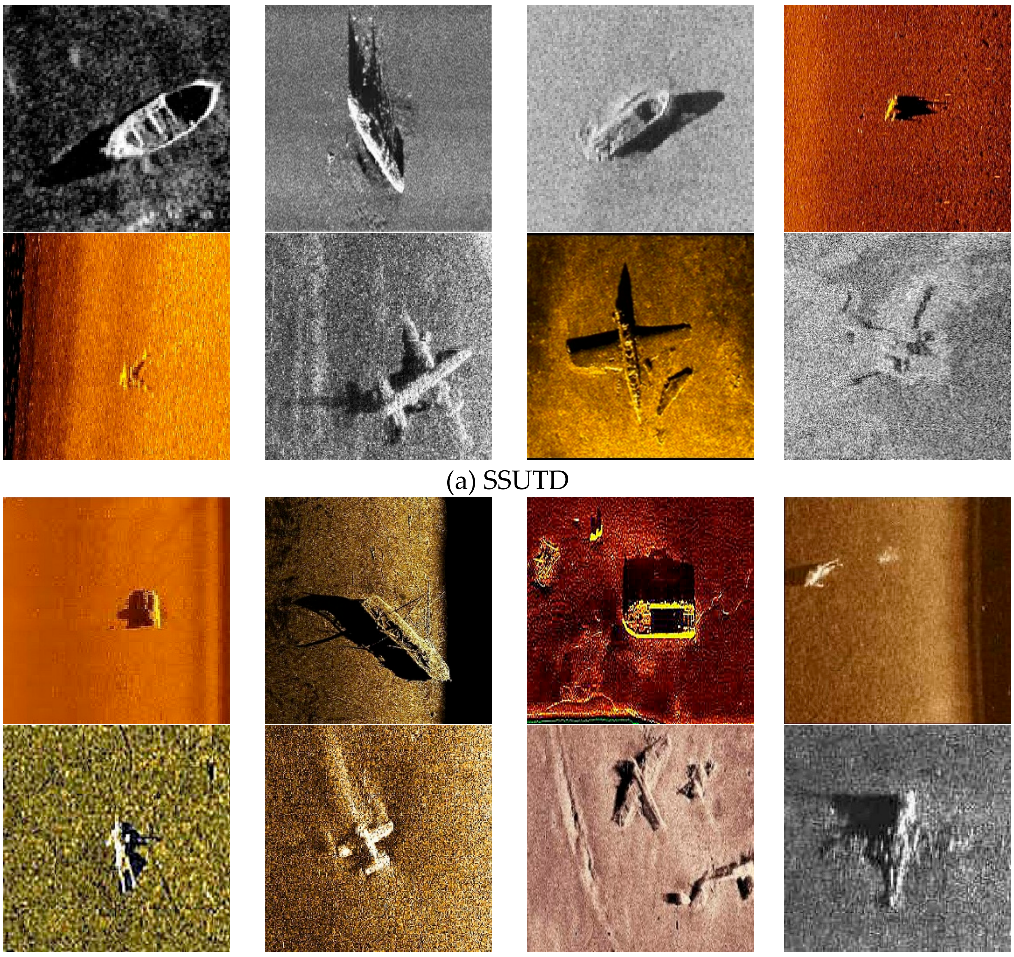
4. Results and Discussion

4.1. Implementation Details

4.1.1. Dataset

Two SSS datasets, SSUTD (Side-Scan Sonar Undersea Target Dataset) and Sonar Common Target Detection Dataset (SCTD), were collected for detector training and testing. The

SSUTD was mainly constructed by various hydrographic departments and companies by using mainstream side-scan sonar instrumentation, and a small portion of the data were collected from the Internet, totaling 1584 images. The SCTD mainly contains side-scan sonar images of airplane wrecks and shipwrecks. Some samples from the SSUTD (dataset A) and SCTD are shown in Figure 8.



(a) SSUTD

(b) SCTD

Figure 8. Selected samples from SSUTD and SCTD, both of which contain side-scan sonar images of airplane wrecks, shipwrecks, and drowned people.

The effectiveness of deep learning algorithms hinges on the quantity and distribution of the training datasets. When training underwater target recognition models, two challenges arise, i.e., a scarcity of images and an imbalance in the number of various target types, which can hinder the performance of deep learning algorithms. To address these issues, data augmentation techniques were employed to produce more synthetic side-scan sonar images with drowned people targets, of which there were limited instances. Commonly used data enhancement algorithms are Gaussian noise, brightness change, image translation, image rotation, image flipping, image scaling, image cropping, etc. [13]. The results obtained with different data enhancement algorithms are shown in Figure 9.

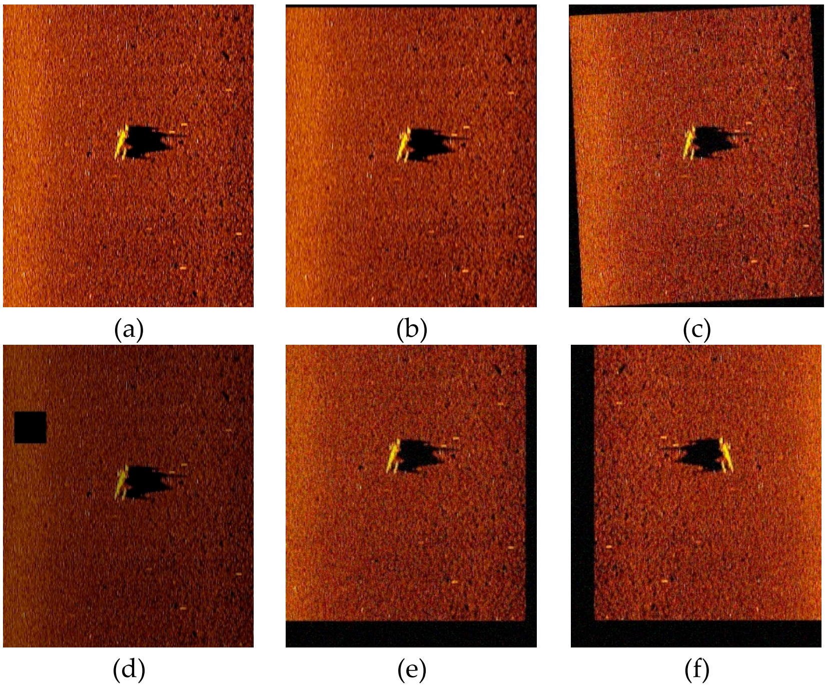


Figure 9. (**a**) The original image; (**b**–**f**) the data enhancement results.

Dataset A consists of 980 images of shipwrecks, 36 images of drowned people and 568 images of airplane wrecks; the SCTD consists of 266 images of shipwrecks, 34 images of drowned people, and 57 images of airplane wrecks; and dataset B consists of data augmented with each drowned people image based on dataset A. The datasets were each divided into three subsets, i.e., training, validation, and test sets, in the ratio of 8:1:1. The detailed division is shown in Table 1.

Table 1. Division of datasets.

Dataset / Target	Training Set			Validation Set			Test Set		
	A	**SCTD**	**B**	**A**	**SCTD**	**B**	**A**	**SCTD**	**B**
Boat	784	213	784	98	27	98	98	26	98
Human	29	27	173	4	4	22	3	3	21
Plane	454	46	454	57	6	57	57	5	57

4.1.2. Detector Training

During model training, we utilized the Adam optimizer, initiating it with a learning rate of 0.001, which was progressively increased to 0.01. To expedite parameter update, we assigned a momentum value of 0.937. Proper regularization was ensured through the careful tuning of the weight decay to 0.0005, aimed at preventing both overfitting and underfitting. The training protocol commenced with a warm-up phase consisting of three rounds, followed by a total of 200 training rounds. All models were trained from the beginning without relying on any pre-trained versions. Given the challenges in obtaining side-scan sonar image data and the limited sample size, the dataset was divided into a training set and a validation set in an 8:2 ratio, aiming to balance model training effectiveness with training efficiency. To expedite the experimental process while maintaining result accuracy, varying levels of data augmentation were applied depending on the model size.

To gain a deeper understanding of the target characteristics within the datasets, we conducted a statistical analysis of the targets' positional distribution across the images.

Additionally, we examined the aspect ratios of specific targets, such as sunken ships and aircraft, in relation to the overall dimensions of the images. The shade of the color represents the quantity, with a darker color indicating a greater quantity. The specific statistics are presented as shown in Figure 10.

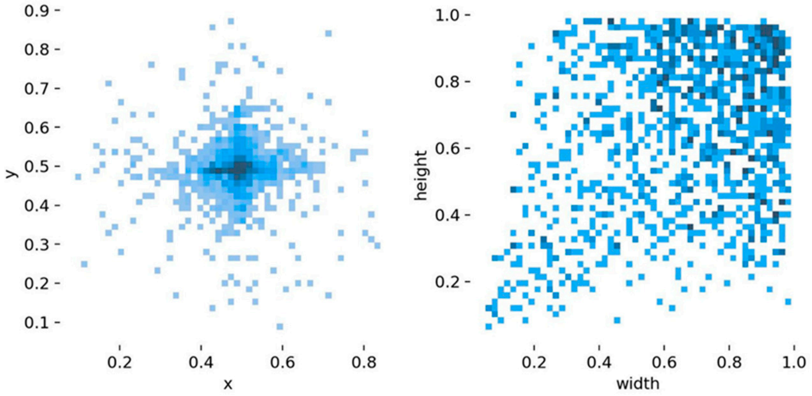


Figure 10. The distributions of targets and their sizes.

4.1.3. Evaluation Metrics

To evaluate the performance of the proposed model, several commonly recognized indicators in object detection tasks were selected, including IoU, P, R, and mAP, which are used to measure the effectiveness of object detection models. Equation (3) shows the IoU calculation formula, where B_{pre} represents the predicted frame and B_{gt} represents the real frame. IoU indicates the degree of overlap between the predicted frame and the real frame, i.e., the intersection ratio between the detection result and the actual labeled frame. With this metric, an IoU (intersection over union) threshold is set, usually to 0.5, to determine the accuracy of the predicted frame. Predictions exceeding this threshold are considered correct, while predictions below this threshold are considered invalid.

$$IoU = \frac{B_{pre} \cap B_{gt}}{B_{pre} \cup B_{gt}} \tag{3}$$

mAP (where the IoU is usually taken as 0.5 or 0.5–0.95) is used to assess the ability of the algorithm to correctly detect the target and is the most important evaluation metric for detection algorithms. According to the accuracy of the prediction frame, the assessment of the target algorithm includes four samples: TP (true positive), where an aircraft wreckage image is correctly predicted as aircraft wreckage; TN (true negative), where a non-aircraft wreckage image is correctly predicted as non-aircraft wreckage; FP (false positive), where a non-aircraft wreckage image is incorrectly predicted as aircraft wreckage; FN (false negative), where an aircraft wreckage image is incorrectly predicted as non-aircraft wreckage. Moreover, from this, several other assessment metrics can be derived:

$$P = \frac{TP}{TP + FP} \tag{4}$$

$$R = \frac{TP}{TP + FN} \tag{5}$$

Average precision (AP) represents the geometric area enclosed by the Precision–Recall (P-R) curve, as illustrated in Equation (6):

$$AP = \int_0^1 p(r)dr \tag{6}$$

Furthermore, Params and FLOPs are indicators of a model's computational space complexity and intensity, respectively, providing insights into the model's complexity and the impact of making the model lightweight. FLOPs, which stands for Floating-Point Operations, denotes the count of floating-point calculations. For convolutional layers, the formula to calculate FLOPs is given as follows:

$$FLOPs = 2 \times \mathrm{H} \times \mathrm{W} \times (C_{in} \times K^2 + 1) \times C_{out} \quad (7)$$

4.2. Performance Evaluation

Figure 11 displays the confusion matrix for the proposed model. The horizontal axis denotes the actual values, while the vertical axis indicates the predicted values. It is evident from the matrix that a significant portion of the predictions align with the actual values. Figure 12 compares the mAP@0.5 curves of the improved model, DBnet, against the YOLOv8 model. Upon convergence, DBnet's curve surpasses YOLOv8's, demonstrating faster and more stable convergence. The experimental data reveal that DBnet not only converges more efficiently (faster and more stably) but also outperforms YOLOv8 in detection quality and model complexity.

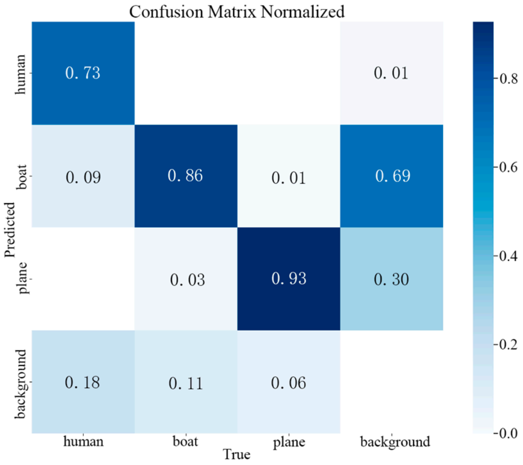


Figure 11. The normalized confusion matrix of the model.

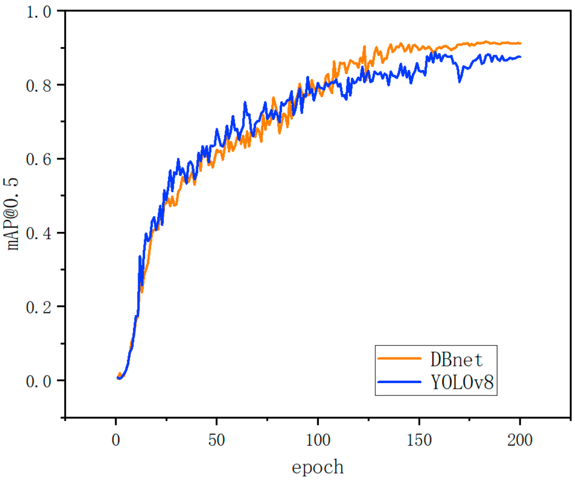


Figure 12. mAP comparison curves of DBnet and baseline model.

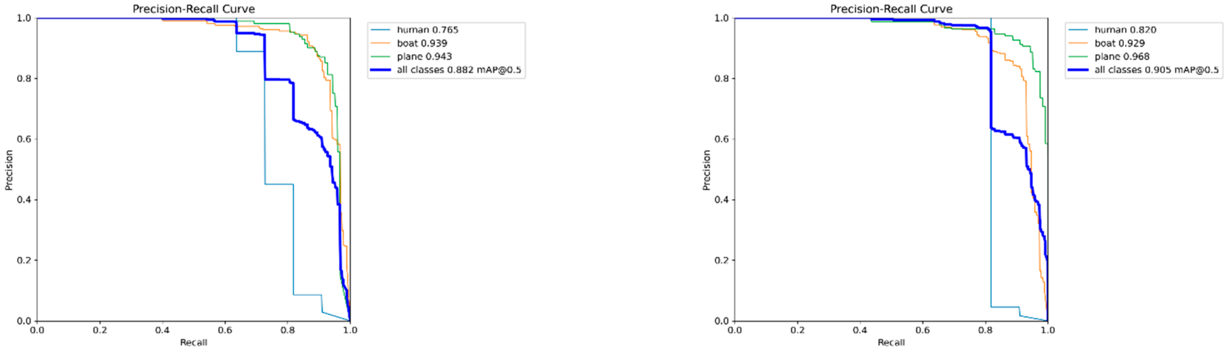
The two detectors were tested on the test set and the comparison of mAP, GFLOPs, and parameters is shown in Table 2.

Table 2. Model performance evaluation.

Detector	mAP@0.5	mAP@0.5-0.95	Parameters	GFLOPs [1]
YOLOv8n	88.2	61.9	3006233	8.1
DBnet	90.5	67.3	2010617	5.6

[1] Giga Floating-Point Operations Per Second (GFLOPs), one billion floating-point operations per second.

A Precision–Recall curve plots precision on the vertical axis and recall on the horizontal axis. Since there is often a trade-off between precision and recall, the P-R curve offers a comprehensive assessment of a model's performance. The P-R curves depicted in Figure 13 showcase the experimental outcomes of YOLOv8 and the proposed DBnet model under identical conditions. These curves present the AP@0.5 values for individual categories and the overall mAP@0.5. Notably, the proposed algorithm boosts mAP@0.5 from 88.2% to 90.5%, marking a 2.3% improvement. It is worth mentioning that in YOLOv8's results, the AP value for the "human" category, characterized by small target sizes, was only 76.5%, but it increased to 82% with the proposed improvements. This particular category saw significant enhancements compared with the baseline model. Despite the reduction in AP for the detection of shipwreck targets, the enhancement in the detection of aircraft wreckage is significant, and overall, there is a significant improvement in detection accuracy compared with the YOLOv8 model.



(a) (b)

Figure 13. P-R curves of YOLOv8n and DBnet: (**a**) P-R curve of YOLOv8n detector; (**b**) P-R curve of DBnet detector.

For inference, we used the SAHI algorithm, which utilizes the principle of sliding slices to crop the large-size input map into multiple slices with a certain overlap rate before inputting it into the detector in order to improve accuracy in detecting small targets, which is implemented as shown in Figure 14.

It can be found that the shipwrecks in the image can be accurately identified after using SAHI as in Figure 15b, which avoids the situation in Figure 15a, in which the terrain undulation in the background is misjudged as a shipwreck. It can be seen that the use of SAHI can effectively improve the detection accuracy of the model for small targets, especially for large-size images such as side-scan sonar waterfall maps, as it prevents small targets from being missed and large targets from being falsely detected due to the image compression caused by the input network, thus optimizing overall model performance.

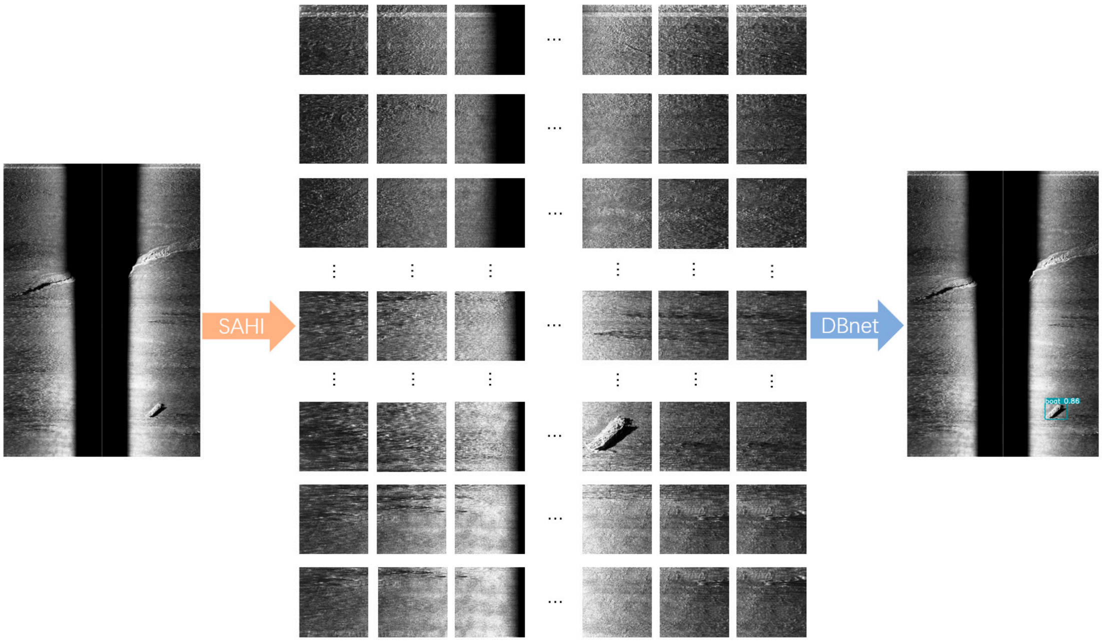


Figure 14. The orange arrows in the figure represent the slicing operation on the original large-size image, and the blue arrows represent the input of each slice into the DBnet detector for prediction.

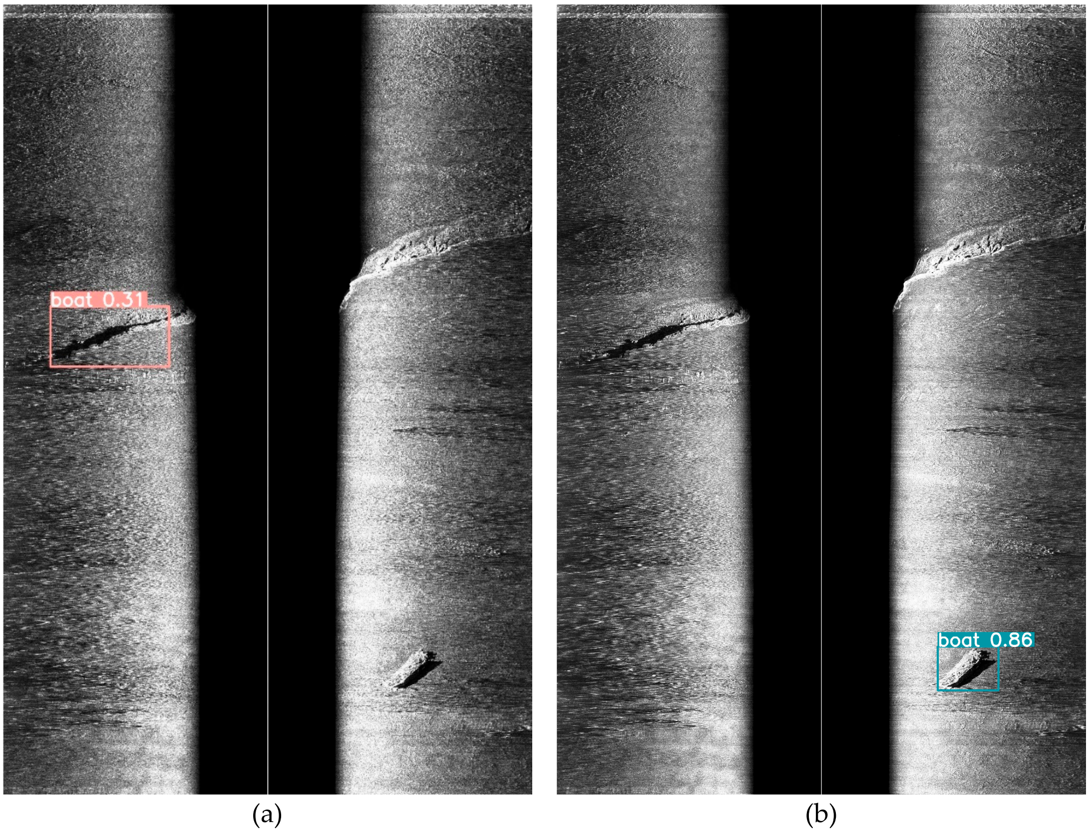


Figure 15. Graphs showing comparison of detection effects. (**a**) shows the results of detection of side-scan sonar images using the baseline model. (**b**) The result of using DBnet to detect the side-scan sonar image.

4.3. Ablation Study

To verify the impact of different backbone network choices on model performance and the generalization of the model on different datasets, we conducted comparative experiments. Firstly, in terms of the selection of backbone networks, the proposed dual-backbone detector model compensates for the sparse characteristics of target features in side-scan sonar images. At the same time, in order to achieve the lightweight implementation of the model and facilitate engineering deployment, we chose the current mainstream lightweight backbone network for the combination experiments. The experimental results are shown in Table 3.

Table 3. Experimental results on different datasets. Bolded font indicates proposed method.

Backbone	Params	GFLOPs	mAP@0.5	mAP@0.5-0.95	mAP@0.5	mAP@0.5-0.95	mAP@0.5	mAP@0.5-0.95
Dataset			SSUTD		SSUTD + Data Augmentation		SCTD	
Darknet-53 (YOLOv8)	3006233	8.1	88.2	64.5	98.3	77.6	73.8	48
MobileNetV3	2351771	5.7	80.1	58.5	96.7	73.6	62.4	45.9
PP-LCNet	1727461	5	84.2	56.3	96.9	72.4	62.9	45.8
EfficientNet	1907061	5.6	85.6	61.2	98.1	75.9	68.4	42.4
ShuffleNetv2	1710809	5	85.4	52.7	96.9	70	63.4	40.2
GhostNet	1714661	5	89.8	64.3	98.4	77.4	77.6	45
GhosNet + ShuffleNetv2	2511993	7	89.6	65.7	98.2	76.5	74.8	50.9
PP-LCNet + ShuffleNetv2	1439461	4.5	85.9	57.2	96.1	67.7	69.9	50
PP-LCNet + GhostNet	2010617	5.6	90.5	67.3	98.5	76.8	80.4	55.1

From the table data, we can see that DBnet marks a significant improvement in the mAP value compared with most of the single-backbone models considered. Although the number of parameters and the GFLOPs of the proposed dual-backbone network are not the smallest metrics, in the comprehensive detection performance index of mAP, the proposed algorithm not only shows better detection performance but also has the great advantage of the model's lightweight degree compared with other networks, which can meet the demand of real-time target detection in side-scan sonar images and lightweight engineering deployment.

Since the proposed detector uses a dual-backbone network to extract features through class multimodal data, which is very different from the baseline model (YOLOv8), we streamlined the neck part of the baseline model. Specifically, we discarded the use of upsampling in the original model to fuse small-size features from high-level convolution with large-size ones from low-level convolution because upsampling often results in some negative effects, such as noise amplification. Therefore, we simplified the neck part, and in order to verify the results of the simplification, we performed the following experiment. The experimental results are shown in Table 4.

Table 4. Results of simplified neck experiment.

Module	mAP@0.5 (%)	mAP@0.5-0.95 (%)	Params	GFLOPs
Original neck	90.2	68.1	2195129	5.8
Simplified neck	90.5	67.3	2010617	5.6

The experimental results show that using the original structure in the neck not only results in redundant computation but also hardly contributes to the enhancement in the model's effectiveness. Therefore, we introduced the streamlined neck structure into the model's structure, which can adequately provide the fusion of the features extracted from the dual feature extraction backbone and, at the same time, contribute to making the model lightweight to a certain extent.

5. Conclusions

Aiming at addressing the problems of existing target detection models with many parameters, long computation time, and high computing requirements, we developed DBnet, and in order to make it lightweight, we selected a lightweight backbone network; at the same time, in order to solve the problem of accuracy loss caused by this choice, we designed a two-branch structure, that is, we employed a dual-backbone network for target feature extraction. This enables the model to mine more target feature information from the image and realize efficient feature extraction and fusion even if the input consists of single-mode data. In addition, on account of this two-branch structure, we only need to fuse the corresponding feature layers in the neck to allow the model to better characterize the target. While maintaining high detection accuracy, the number of parameters and computation amount of the model are greatly reduced, achieving a balance between detection speed and accuracy. Finally, for addressing the problem that original waterfall maps of side-scan sonar are large in size and prone to the loss of detail information after their input into the network, we adopted the slice-assisted hyper-inference (SAHI) technique, which splits large-size images into multiple small-size images for inference, improving target detection accuracy by fusing the detection results of each slice. Compared with the baseline model, DBnet presents 33% fewer parameters and 31% less computation (GFLOPs) while maintaining accuracy, which is especially important in resource-limited environments. The effectiveness of DBnet is further confirmed by test results on the SSUTD and SCTD, with the mAP values improving by 2.3% and 6.6%, respectively. In addition, the lightweight design of DBnet makes it easier to deploy in engineering applications, especially in mission scenarios such as AUV underwater target detection, that require real-time detection capabilities.

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

Funding: This research study was funded by the National Natural Science Foundation of China under grant 41876103.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Access to the data will be considered by the authors upon request.

Acknowledgments: We would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions that have greatly improved the quality of this paper. We thank our colleagues for their helpful suggestions during the experiment. Finally, we are grateful for access to the Ultralytics-Yolov8 open-source project.

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

References

1. Yang, D.; Wang, C.; Cheng, C.; Pan, G.; Zhang, F. Semantic Segmentation of Side-Scan Sonar Images with Few Samples. *Electronics* **2022**, *11*, 3002. [CrossRef]
2. Xie, Y.; Bore, N.; Folkesson, J. Bathymetric Reconstruction from Sidescan Sonar With Deep Neural Networks. *IEEE J. Ocean. Eng.* **2023**, *48*, 372–383. [CrossRef]
3. Chen, Z.; Wang, Z. Research on Underwater Target Detection Using Side-scan Sonar and Multibeam Sounding System. *Hydrogr. Surv. Charting* **2013**, *33*, 51–54.
4. Tang, Y.; Wang, L.; Jin, S.; Zhao, J.; Huang, C.; Yu, Y. AUV-Based Side-Scan Sonar Real-Time Method for Underwater-Target Detection. *J. Mar. Sci. Eng.* **2023**, *11*, 690. [CrossRef]

5. Wang, J.; Zhou, J. Comprehensive Application of Side-scan Sonar and Multi-beam System in Shipwreck Survey. *China Water Transp.* **2010**, *10*, 35–37.
6. Liu, C. *The Comparative Analysis of Multi-Beam Sounding System, Side-Scan Sonar and Magnetometer in the Wreck Detection*; China University of Geosciences: Beijing, China, 2015.
7. Zhao, J.; Li, J.; Li, M. Progress and Future Trend of Hydrographic Surveying and Charting. *J. Geomat.* **2009**, *34*, 25–27.
8. Wang, J.; Cao, J.; Lu, B.; He, B. Underwater Target Detection Project Equipment Application and Development Trend. *China Water Transp.* **2016**, *11*, 43–44.
9. Neupane, D.; Seok, J.-H. A Review on Deep Learning-Based Approaches for Automatic Sonar Target Recognition. *Electronics* **2020**, *9*, 1972. [CrossRef]
10. Ge, Q.; Ruan, F.; Qiao, B.; Zhang, Q.; Zuo, X.; Dang, L. Side-Scan Sonar Image Classification Based on Style Transfer and Pre-Trained Convolutional Neural Networks. *Electronics* **2021**, *10*, 1823. [CrossRef]
11. Yu, Y.; Zhao, J.; Gong, Q.; Huang, C.; Zheng, G.; Ma, J. Real-Time Underwater Maritime Object Detection in Side-Scan Sonar Images Based on Transformer-YOLOv5. *Remote Sens.* **2021**, *13*, 3555. [CrossRef]
12. Wang, R.; Li, J.; Duan, Y.; Cao, H.; Zhao, Y. Study on the Combined Application of CFAR and Deep Learning in Ship Detection. *J. Indian Soc. Remote Sens.* **2018**, *46*, 1413–1421. [CrossRef]
13. Li, W.; Zhang, J.; Liu, L.; Zhou, J.; Sui, Q.; Li, H. CFAR Algorithm Based on Different Probability Models for Ocean Target Detection. *IEEE Access* **2021**, *9*, 154355–154367. [CrossRef]
14. Lakshmi, M.D.; Murugan, S.S. Keypoint-based mapping analysis on transformed Side Scan Sonar images. *Multimed. Tools Appl.* **2020**, *79*, 26703–26733. [CrossRef]
15. Williams, D.P. The Mondrian Detection Algorithm for Sonar Imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1091–1102. [CrossRef]
16. Tang, Y.; Jin, S.; Bian, G.; Zhang, Y.; Li, F. Wreckage Target Recognition in Side-scan Sonar Images Based on an Improved Faster R-CNN Model. In Proceedings of the International Conference on Big Data & Artificial Intelligence & Software Engineering, Bangkok, Thailand, 30 October–1 November 2020; pp. 348–354.
17. Jia, R.; Lv, B.; Chen, J.; Liu, H.; Cao, L.; Liu, M. Underwater Object Detection in Marine Ranching Based on Improved YOLOv8. *J. Mar. Sci. Eng.* **2024**, *12*, 55. [CrossRef]
18. Wang, J.; Wang, Q.; Gao, G.; Qin, P.; He, B. Improving Yolo5 for Real-Time Detection of Small Targets in Side Scan Sonar Images. *J. Ocean Univ. China* **2023**, *22*, 1551–1562. [CrossRef]
19. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10778–10787.
20. Tang, Y.; Bian, S.; Zhai, G. *Improved YOLOv5 Method for Detecting Shipwreck Target with Side-Scan Sonar*; Geomatics and Information Science of Wuhan University: Wuhan, China, 2024; pp. 1–11. [CrossRef]
21. Li, L.; Li, Y.; Yue, C.; Xu, G.; Wang, H.; Feng, X. Real-time underwater target detection for AUV using side scan sonar images based on deep learning. *Appl. Ocean. Res.* **2023**, *138*, 103630. [CrossRef]
22. Zhou, J.; Si, Y.; Chen, Y. A Review of Subsea AUV Technology. *J. Mar. Sci. Eng.* **2023**, *11*, 1119. [CrossRef]
23. Wang, Q.; Zhang, Y.; He, B. Intelligent Marine Survey: Lightweight Multi-Scale Attention Adaptive Segmentation Framework for Underwater Target Detection of AUV. *IEEE Trans. Autom. Sci. Eng.* **2024**, 1–15. [CrossRef]
24. Song, S.; Liu, J.; Guo, J.; Wang, J.; Xie, Y.; Cui, J.-H. Neural-Network-Based AUV Navigation for Fast-Changing Environments. *IEEE Internet Things J.* **2020**, *7*, 9773–9783. [CrossRef]
25. Yulin, T.; Jin, S.; Bian, G.; Zhang, Y. Shipwreck Target Recognition in Side-Scan Sonar Images by Improved YOLOv3 Model Based on Transfer Learning. *IEEE Access* **2020**, *8*, 173450–173460. [CrossRef]
26. Li, R.; Li, Y.; Qin, W.; Abbas, A.; Li, S.; Ji, R.; Wu, Y.; He, Y.; Yang, J. Lightweight Network for Corn Leaf Disease Identiffcation Based on Improved YOLO v8s. *Agriculture* **2024**, *14*, 220. [CrossRef]
27. Zhang, Q.; Li, L.; Zhang, Z.; Yin, S.; Ma, L. Marine target detection for PPI images based on YOLO-SWFormer. *Alex. Eng. J.* **2023**, *82*, 396–403. [CrossRef]
28. Huang, H.; Zuo, Z.; Sun, B.; Wu, P.; Zhang, J. DSA-SOLO: Double Split Attention SOLO for Side-Scan Sonar Target Segmentation. *Appl. Sci.* **2022**, *12*, 9365. [CrossRef]
29. Li, J.; Chen, L.; Shen, J.; Xiao, X.; Liu, X.; Sun, X.; Wang, X.; Li, D. Improved Neural Network with Spatial Pyramid Pooling and Online Datasets Preprocessing for Underwater Target Detection Based on Side Scan Sonar Imagery. *Remote Sens.* **2023**, *15*, 440. [CrossRef]
30. Ji, C.L.; Yu, T.; Gao, P.; Wang, F.; Yuan, R.-Y. Yolo-TLA: An Efficient and Lightweight Small Object Detection Model based on YOLOv5. *J. Real-Time Image Process.* **2024**, *21*, 141. [CrossRef]
31. Zhang, M.; Xu, S.; Song, W.; He, Q.; Wei, Q. Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]

32. Tang, H.; Gao, S.; Li, S.; Wang, P.; Liu, J.; Wang, S.; Qian, J. A Lightweight SAR Image Ship Detection Method Based on Improved Convolution and YOLOv7. *Remote Sens.* **2024**, *16*, 486. [CrossRef]
33. Ma, Q.; Jin, S.; Bian, G.; Cui, Y. Multi-Scale Marine Object Detection in Side-Scan Sonar Images Based on BES-YOLO. *Sensors* **2024**, *24*, 4428. [CrossRef]
34. Akyon, F.C.; Onur Altinuc, S.; Temizel, A. Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 966–970.
35. Cui, C.; Gao, T.; Wei, S. PP-LCNet: A Lightweight CPU Convolutional Neural Network. *arXiv* **2021**, arXiv:2109.15099. [CrossRef]
36. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520. [CrossRef]
37. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
38. Kai, H.; Wang, Y.; Tian, Q. GhostNet: More Features from Cheap Operations. *arXiv* **2020**, arXiv:1911.11907. [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.