Hindawi
Journal of Sensors
Volume 2022, Article ID 5827499, 12 pages
https://doi.org/10.1155/2022/5827499

Research Article

An Improved Object Detection Method for Underwater Sonar Image Based on PP-YOLOv2

Fang Wang[1], **Huitao Li**[1], **Kai Wang**[2], **Lichen Su**[3], **Jing Li**[1], **and Lili Zhang**[1,4]

[1]*College of Information Engineering, Beijing Institute of Petrochemical Technology, Beijing 102617, China*
[2]*Institute of National Defense Science and Technology Innovation, Academy of Military Sciences, Beijing 100036, China*
[3]*School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China*
[4]*Xufeng Technology Co., Ltd., Yinchuan 750011, China*

Correspondence should be addressed to Jing Li; bipt_lijing@bipt.edu.cn

Received 24 March 2022; Revised 24 September 2022; Accepted 21 October 2022; Published 21 November 2022

Academic Editor: Grazia Iadarola

Forward-looking sonar is widely used in underwater obstacles and objects detection for navigational safety. Automatic sonar images recognition plays an important role to reduce the workload of staff and subjective errors caused by visual fatigue. However, the application of automatic object classification in forward-looking sonar is still lacking, which is due to small effective samples and low signal-to-noise ratios (SNR). This paper proposed an improved PP-YOLOv2 algorithm for real-time detection, called as PPYOLO-T. Specifically, the proposed method first resegments the sonar image according to different aspect ratio and filters the acoustic noise in various ways. Then, attention mechanism is introduced to improve the ability of network feature extraction. Finally, the decoupled head is used to optimize the multiobjective classification. Experimental results show that the proposed method can effectively improve the accuracy of multitarget detection task, which can meet the requirement of robust real-time detection for both raw and noised sonar targets.

1. Introduction

In recent years, with the rapid development of the "Autonomous underwater vehicle (AUV)", underwater seaway safety has become one of the important research hotspots. The obstacles, large rocks, and piers in the water will greatly affect the path planning and task execution of AUV, and more seriously, safety accidents may occur. As a kind of high resolution, multipurpose marine detection equipment, forward-looking sonar, installed in front of AUV, is an easily accessible and economical device to obtain images of the underwater obstacles and objects. It is widely applied in various fields such as automatic obstacle avoidance, seabed mapping, ecological monitoring, and pipeline inspection.

The forward-looking sonar can record the back-scattered echo intensity of the object and generate sonar images with different gray levels according to the echo intensity, which is called the reflection intensity imaging of the object. Compared with other acoustic detection systems, the advantages of object detection using forward-looking sonar are as follows:

(i) High data density and high resolution

(ii) Large coverage and strong recognition ability for underwater objects with special shapes

(iii) Easy installation and low cost

However, the traditional sonar system cannot automatically obtain the accurate positioning information of underwater object. It requires manual identification or off-line ashore postprocessing, which seriously affects the real-time and initiative of underwater task execution of AUV. And the recognition and classification accuracy is affected by unclear image edge and multi-image-noise because of the complexity of sound propagation in water medium and the characteristics of sound wave.

Many researchers have studied the automatic object detection from sonar images, such as traditional artificial

design feature method [1], machine learning (ML) method [2], and deep learning (DL) method [3]. Relying on artificially designed features, traditional sonar image object detection methods cannot make full use of the deep features of sonar image and lack of robustness and generalization ability. The method [3–5] based on deep learning has gradually become the mainstream method of sonar image object recognition, because of its powerful automatic feature extraction capability.

According to whether a region proposal is generated and used, DL-based object detection methods can be divided into two-stage model and one-stage model. Following the idea of traditional object detection, two-stage models first generate a large number of regional suggestions in the detection process and then generate fixed-size feature maps to perform localization and classification tasks, respectively. Region-based convolutional neural network (R-CNN) [3] is the first two-stage model. It innovatively utilized convolutional neural network (CNN) to extract image features. Other typical two-stage models include Faster R-CNN [6] and cascade R-CNN [7], which are proposed successively to improve the detection efficiency.

The single-stage model does not generate regional suggestions; therefore, the calculation is relatively small, the detection rate is fast, and the real-time performance is high, but the accuracy of the model detection is sacrificed. YOLO (you only look once) model [5] is a typical one-stage model, which is commonly known as YOLOv1. YOLOv3 [8] and YOLOv5 are famous variants of YOLOv1. PP-YOLOv2 [9] adopts a set of optimization strategies to improve the accuracy of the detector and achieve a very high cost-performance ratio. Its mean average precision (mAP) is 45.9, and the frame per second (FPS) achieves 72.9.

At present, sonar object detection is still a very challenging task due to the problems of multiple scales, dual priorities, speed, limited data, and class imbalance. These problems have a big effect on the real-time detection accuracy. To implement the real-time object recognition in sonar images efficiently, the following works have been done in this paper.

(i) PP-YOLOv2 is first introduced to the underwater obstacles and objects detection for real-time sonar image object detection

(ii) Some useful preprocessing methods are presented for sonar image, including noise reduction, image resizing, and CutMix [10].

(iii) Some updates to PP-YOLOv2 are proposed including a backbone network with an attention mechanism and decoupled head, which finally forms a high-performance sonar image multiobject detector called PPYOLO-T

The paper is structured as follows: section Related Work is an overview of the related work; section Methodology describes our method for constructing sonar multiobject detection network based on PP-YOLOv2, followed by extensive experiments for evaluating the proposed method in section Experiments. We conclude this paper in section Conclusions.

2. Related Work

Sonar image-oriented object detection is of great significance to underwater detection. It has been studied for many years. Traditional sonar image object detection methods are mainly based on artificial design features. Myers and Fawcett [1] proposed a sonar image object detection algorithm based on template matching (TM), where objects were located and classified by using the features of the template designed manually. Much work was devoted to artificial design features. Some useful features include physical characteristics of foreground and background [11], context information [12], and statistics about the environment [2].

However, traditional sonar image object detection methods cannot make full use of the deep features of sonar image for decision making. At the same time, they are usually lack of robustness and generalization ability. All these limit the application of traditional methods. In recent years, researchers [13–15] have introduced deep learning-based object detection method into sonar image object detection and achieved some good achievements. At present, the mainstream deep learning detection algorithms can be divided into two series of R-CNN [3] based and YOLO [5] based.

2.1. R-CNN Based Methods. In recent years, convolutional neural networks (CNN) have been widely used in classification tasks. The region-based convolutional neural network (R-CNN) proposed by Girshick et al. [3] firstly introduced the convolutional neural network into the object detection task. It greatly makes up for the defects of traditional object detection algorithms such as deformable part model (DPM) [16] in high complexity and high computation. Then, spatial pyramid pooling (SPP) [17] and Fast R-CNN [6] further improved the accuracy of object detection on natural image data sets Pascal VOC [18] and MS COCO [19]. Faster R-CNN [6] saved the calculation cost of regional proposals by introducing region proposal network (RPN), enabled end-to-end training of the whole model, and improved detection efficiency. Cascade R-CNN [7] used cascade regression as a resampling mechanism to improve intersection-over-union (IoU) value of the proposal stage by stage, so that the resampled proposals of the previous stage can adapt to the next stage with a higher threshold.

The above R-CNN-based methods are two-stage model. In the detection process, a large number of regional suggestions are generated or referenced, and fixed-size feature maps are generated based on this, so as to perform localization and classification tasks, respectively. As a result, R-CNN-based models usually have large number of model parameters and slow detection speed, which makes it difficult for real application.

2.2. YOLO-Based Methods. YOLO [5] is a brand new network different from the regional convolutional neural network. It transforms the problem of object detection into a regression problem. The classification probability and location information of the object can be given only with one neural network and one single detection. This gives YOLO

a huge advantage in terms of infer time and detection accuracy, making it possible for real-time application. The models [8, 20, 21] from YOLOv2 [22] to YOLOX [23] constantly improve the model in terms of performance and speed. Methods for improvement include new backbones such as Darknet-19 [22] and Darknet-53 [8], adding SPP layer, new training strategy of multiscale and exponential moving average (EMA), and decoupled head.

Unlike YOLOv4 and YOLOv5 that explore various complex backbone and data augmentation methods, PP-YOLO [4] is based on YOLOv3 and only relies on Mixup and keeps improving model performance through reasonable combination of tricks. PP-YOLOv2 [9] adopts a set of optimization strategies to improve the accuracy of the detector and achieve a very high cost-performance ratio (mAP 45.9 and 72.9 FPS) on the premise of almost not increasing model parameters and computation (flops).

This paper focuses on multiple target detection of underwater sonar images. Different from existing methods, we explore some preprocessing methods to improve the model robustness and design a new detection model based on PP-YOLOv2 to improve the detection accuracy for underwater sonar images.

3. Methodology

This section first shortly reviews the PP-YOLOv2 and then elaborates the proposed method for sonar object detection, which is called PPYOLO-T in this paper.

PP-YOLOv2 is an optimization model based on PP-YOLO [4] and YOLOv3 [8]. It uses the same backbone network with PP-YOLO, called ResNet50-vd [24], and more tricks are added, which can improve the model accuracy without introducing extra computation as much as possible. Specifically, it uses path aggregation network [25] (PANet) to aggregate the top-down information in the detection neck, applies the mish activation function [26] in the detection neck instead of the backbone, and increases the input size and applies a soft label format for the IoU aware loss.

The challenges of sonar object detection lie many facts, such as low SNR, complex background, and small object. It is difficult to achieve ideal results by directly applying the existing model to the sonar image detection task. In this paper, we first preprocess sonar image, and then based on PP-YOLOv2, propose PPYOLO-T for better sonar multiobject detection. Figure 1 shows the overall flow of our method. There are mainly three parts: sonar image preprocessing, trainning of PPYOLO-T, and target detection for a new sonar image.

3.1. Preprocessing. Due to the complexity of sonar equipment in the market, the resolution of sonar images is usually different, and there are many noises. To train a robust model for sonar image detection, preprocessing is necessary. As shown in Figure 1, the preprocessing for model training includes noise reduction, image resizing, anchor resizing, and data augmentation. For sonar image detection in real application, only noise reduction is used.

3.1.1. Noise Reduction. There is a lot of acoustic noise in sonar image. In this paper, we leverage Gaussian filtering, median filtering, and bilateral filtering to reduce noises in the original sonar images. Figure 2 shows the noise reduction process.

Gaussian filter is a kind of signal filter for signal smoothing. We use it to improve the SNR of sonar images. The following equation shows its calculation formula, in which, (x, y) is the current coordinate point, (x_0, y_0) is the central coordinate point, and σ is a Gaussian smooth curve.

$$G(x, y) = \frac{1}{2\pi\sigma^2} \times \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma^2}\right). \quad (1)$$

Median filtering is a kind of nonlinear signal processing technology which can suppress the noise effectively based on the sorting statistics theory. It has a good filtering effect on the pulse noise. Especially when filtering the noise, it can protect the edge of the signal from being blurred. The following equation shows its calculation formula, in which $f(x, y)$ and $g(x, y)$ are the original image and the processed image, respectively, and W is a two-dimensional template, which is a 3×3 region in this paper.

$$g(x, y) = \text{med}\{f(x - k, y - l), (k, l \in W)\}. \quad (2)$$

Bilateral filtering is a popular noise filtering method. It is optimized on the basis of Gaussian, superimposed the consideration of pixel value. The filtering effect is more effective to preserve edge. Therefore, it is beneficial to the edge detection of stereo object in underwater sonar image. The following equation shows its calculation formula, in which $1/W_p$ stands for a normalization factor, G_{σ_s} is space weight, and G_{σ_r} is range weight.

$$\text{BF}[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q. \quad (3)$$

3.1.2. Image Resizing. In many cases, sonar images collected by different devices vary in both resolution and image size. However, DL-based methods need to use unified image size for model training. This paper proposes a simple but effective method to resize sonar images to the same size without loss of resolution.

Figure 3 shows the image resizing process. We first obtain a list of various length-width ratios by statistics on the data, such as 2048×768 and 2048×512. Then, in each iteration sample of model training, we randomly segment the original image according to length-width ratio list and complete the segmented image with gray bars for the deficiency based on a predefined image size, such as 640×640 or 768×768. Finally, we use the reconstructed images with unified size for model training.

Instead of stretching a sonar to a uniform size, we normalize the image size by cutting and filling. This does not deform the image and preserve the original resolution of the image. Therefore, the generalization ability and recognition effect of

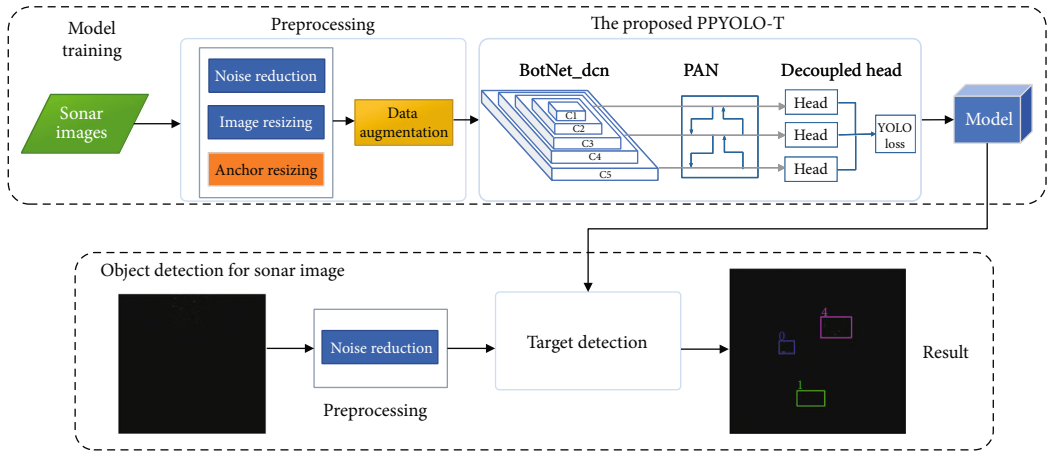

Figure 1: The process of sonar image target detection.

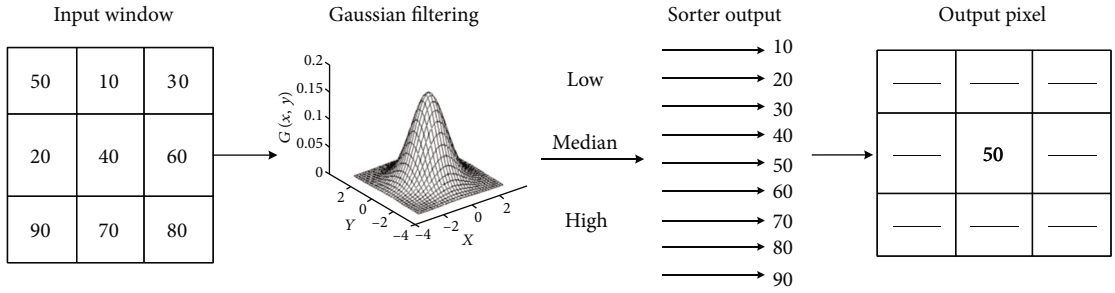

Figure 2: Preprocessing of noise reduction for sonar image.

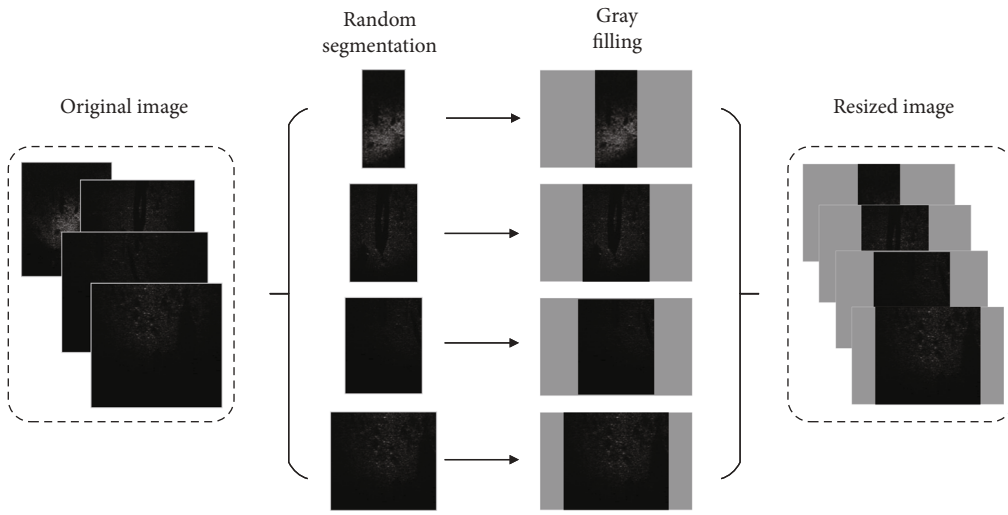

Figure 3: The process of image resizing.

the model can be improved. We will illustrate this through ablation experiment in section Experiments.

3.1.3. Anchor Resizing. Based on PP-YOLOv2 for improvement, the proposed PPYOLO-T is also anchor-based detection method. Anchor is actually a set of preset bounding boxes of different scales and sizes. During network training, the real bounding position is offset from the preset bounding position. In PP-YOLOv2, anchor box is preseted based on COCO data set. In this paper, we resize the anchor box based on real sonar images, in the account of small object detection. Specifically, we leverage K-means [27] algorithm to cluster all the labeled boxes. The parameter K is set to 9 following PP-YOLOv2. The mean bounding positions of

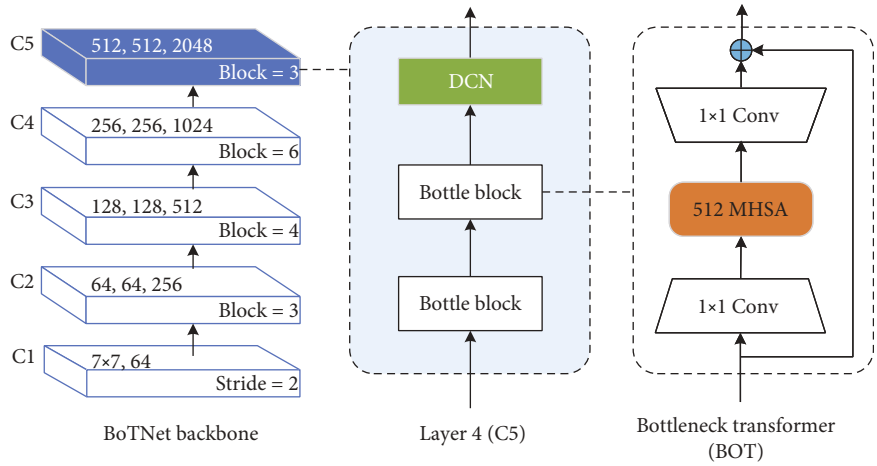

Figure 4: The structure of our backbone network.

each cluster are selected as the preset anchor box. Anchor resizing for sonar images is proved to be effective. Details are shown in section Experiments.

3.1.4. Data Augmentation. Data augmentation is an effective technique for improving the accuracy of image-related tasks such classification and object detection. In this paper, flipping, random expansion, CutMix, and Mosaic [28] are used for sonar image preprocessing. Mixup [29] was used for data augmentation in PP-YOLOv2, and good results were achieved on COCO data set. However, when we apply Mixup to sonar image preprocessing, it finally backfired. This may be because the SNR of sonar image is much lower than ordinary RGB images. Mixup is to overlap two photos together. If the objects in sonar image overlap, the difference between the overlapped object features and the original object features will be too large. Inspired by this, other preprocessing that may change the appearance of the original image, such as adjusting brightness, is also excluded for data augmentation of sonar image.

3.2. The Proposed PPYOLO-T. After data preprocessing, sonar images are transmitted to PPYOLO-T for model training, and then the trained model is generated for sonar image object detection. As shown in Figure 1, the overall architecture of PPYOLO-T consists three parts, namely, the backbone BotNet-dcn, the neck PAN, and the decoupled head. Among which, BotNet-dcn and decoupled head will be elaborated in this subsection. We omit the details about PAN since it is the same as used in PP-YOLOv2.

3.2.1. BotNet-dcn. In original PP-YOLO and PP-YOLOv2 [9], ResNet50-vd-dcn is applied to extract feature maps at different scales. ResNet [30] has been widely used in a variety of feature extraction applications. But most recently, attention mechanism [31] has been gradually applied in the field of machine vision. There already exist some backbones used in image feature extraction such as BotNet [32] and Swin transformer [33].

In this work, BotNet is chosen for the backbone in considering of efficiency. Swin transformer, which is

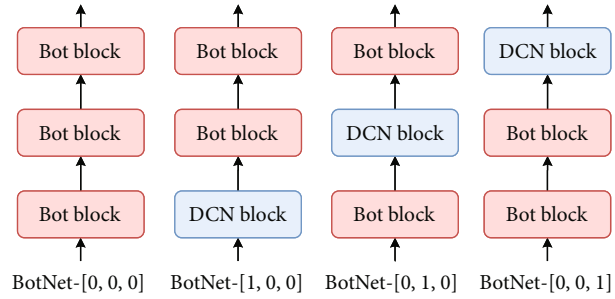

Figure 5: Possible DCN position in BotNet.

stacked by the attention mechanism, will lead to a significant decline in the efficiency of object detection. While BotNet only replaces some 3×3 convolution layers with multihead self-attention (MHSA). Its reasoning efficiency is much higher than that of Swin transformer, and the number of parameters is even lower than that of the original ResNet. Figure 4 shows the backbone network designed for sonar object detection.

Following the way of PP-YOLO [4], we replace some convolution layers in BotNet with deformable convolution networks (DCNs) in the consideration that directly replacing BotNet with ResNet will hurt the performance of PP-YOLO detector. Also, in order to balance the efficiency and effectiveness, we only replace MHSA in the last stage with DCNs, as shown in Figure 4. We denote this modified backbone as BotNet-dcn.

As shown in Figure 5, there are several positions for the replacement of MHSA with DCNs. We will demonstrate their effectiveness through ablation experiments in section Experiments. Our experimental result shows that replacing MHSA in the last stage with DCNs performs the best.

To sum up, the proposed BotNet-dcn first leverages three CNN layers to extract the local features of the image and then uses two multihead attention module layers to integrate the global features and further utilize deformable convolutional network for further adjustment and finally output the feature map. We will prove its effectiveness in experiments.

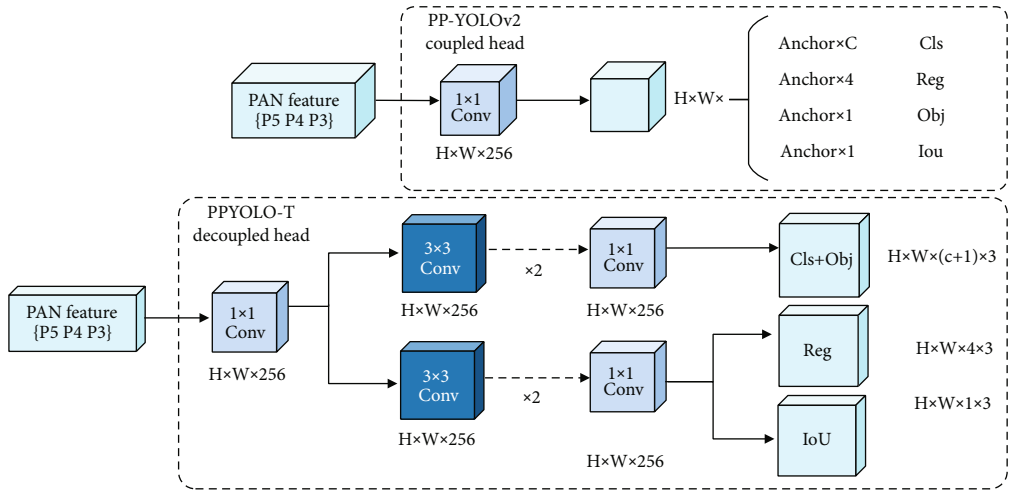

Figure 6: Illustration of the difference between PP-YOLOv2 head and the proposed PPYOLO-T head.

3.2.2. Decoupled Head. Head is a part of model structure to predict object category and position (bounding box). Decoupled head has been widely used in most of one-stage and two-stage detectors [34, 35]. However, as YOLO series' backbones and feature pyramids (e.g., feature pyramid network [36] and pixel aggregation network [25]) continuously evolving, their detection heads remain coupled as shown in Figure 6. YOLOX, proposed by Ge et al. [23], shows that replacing coupled head with decoupled head can greatly improve the model performance. Based on PP-YOLOv2, this paper proposes a decoupled head for sonar object detection.

As shown in Figure 6, the coupled head used in PP-YOLOv2 generates $3 \times (1 + 4 + 1 + \text{Classes})$ channels through 3×3 convolution, and the extra channel is used to calculate IoU aware loss for smoothly processing of prediction information. Differently, we use decoupled head following YOLOX. But unlike the decoupled head used in YOLOX, we put category prediction and object prediction in the same branch and extend another branch to calculate IoU aware loss for smoothly processing of prediction information.

The calculation of loss is given by Equation (4), which includes three parts: confidence loss, classification loss, and location loss.

$$L(o, c, O, C, l, g) = \lambda_1 L_{\text{conf}}(o, c) + \lambda_2 L_{\text{cla}}(O, C) + \lambda_3 L_{\text{loc}}(l, g). \tag{4}$$

To be specific, confidence loss is calculated by using binary cross-entropy, as shown in Equation (5). The confidence means whether there is a center point at this grid, that is, whether there is an object. $o_i \in [0, 1]$ represents the IoU of the predicted object bounding box and the real object bounding box. c is the predicted value and $\hat{c}_i$ stands for the confidence score computed by the Sigmoid function. N is the number of positive and negative samples.

$$L_{\text{conf}}(o, c) = -\frac{\sum_i (o_i \ln(\hat{c}_i)}{N} - \frac{\sum_i (1 - o_i) \ln(1 - \hat{c}_i))}{N} \hat{c}_i = \text{Sigmoid}(c_i). \tag{5}$$

Classification loss is also calculated by using binary cross-entropy. In Equation (6), $O_{ij} \in \{0, 1\}$ represents whether there is an O_{ij} object in the bounding box of the predicted object. C_{ij} stands for the predicted value, and N_{pos} is the number of positive samples.

$$L_{\text{cla}}(O, C) = -\frac{\sum_{i \in \text{pos}} \sum_{j \in \text{cla}} ((O_{ij} \ln(\hat{C}_{ij})}{N_{\text{pos}}} - \frac{\sum_{i \in \text{pos}} \sum_{j \in \text{cla}} (1 - O_{ij}) \ln(1 - \hat{C}_{ij}))}{N_{\text{pos}}} \hat{C}_{ij} = \text{Sigmoid}(C_{ij}). \tag{6}$$

During training, the squared error loss is used for location loss calculation. Equation (7) shows the computing method.

$$L_{\text{loc}}(t, g) = \frac{\sum_{i \in \text{pos}} (\sigma(t_x^i) - \hat{g}_x^i)^2 + (\sigma(t_y^i) - \hat{g}_y^i)^2}{N_{\text{pos}}} + \frac{\sum_{i \in \text{pos}} (t_w^i - \hat{g}_w^i)^2 + (t_h^i - \hat{g}_h^i)^2}{N_{\text{pos}}}, \tag{7}$$

where $\hat{g}_x^i = g_x^i - c_x^i$, $\hat{g}_y^i = g_y^i - c_y^i$, $\hat{g}_w^i = \ln(g_w^i / p_w^i)$ and $\hat{g}_h^i = \ln(g_h^i / p_h^i)$, representing the coordinates x, y, width, and height of the center point of the labeled box in the training data. t_x, t_y, t_w, and t_h are the regression parameters of network prediction.

The effectiveness of the proposed decoupled head will be illustrated in the following experiments.

3.3. Detection for Underwater Sonar Image. In real application, the detector is required to have good real-time performance. Our PPYOLO-T keeps only one processing step that is noise reduction, since the underwater sonar images have a high signal-to-noise ratio. After noise reduction, the trained model will detect objects in the image and give the results of different categories. It is able to detect multiple

objects in a time and is also friendly to small ones. We will show its effectiveness in the following section.

4. Experiments

In this section, we conduct extensive experiments to assess the effectiveness of our proposed PPYOLO-T on multiobject recognition for sonar images. We first introduce the data set and then elaborate the experimental settings, followed by results and discussions.

4.1. Data Set. In this experiment, we use the forward sonar data from Ocean Space Environment Awareness (Orca) open-source project (https://code.ihub.org.cn/companies/vgz4xa2q, 2022-08-12). One can access the data from github (https://github.com/violetweir/PPYOLO-T/tree/main/dataset, 2022-08-12). There are 5,000 images in total, of which 4,000 for training and 1,000 for test. Table 1 shows details about object categories and the number of sonar images for each category. It can be seen that the number of objects is nearly the same with number of images, indicating that most of sonar images have only one object.

Figure 7 shows some examples from each object category. We can see that the underwater sonar images vary in size, and the resolution is so low that it is hard for us to recognize an object at a glance.

4.2. Experiment Setup. The proposed PPYOLO-T in this paper was developed based on the PaddleDetection framework (https://github.com/PaddlePaddle/PaddleDetection, 2022-08-12), and its source code is opened at github (https://github.com/violetweir/PPYOLO-T, 2022-08-12). All the experiments were conducted on a server equipped with two NVIDIA GeForce 1080Ti GPUs (12GB) and Ubuntu 20.04 operating system.

4.2.1. Baselines. We compare the proposed PPYOLO-T with the following state-of-the-arts on multiobject detection in image area:

(i) Faster R-CNN [6] is one of the representative algorithms of the classic R-CNN series. It is mainly derived from the improvement of the previous version of Fast R-CNN, including the integration of feature extraction, proposal extraction, bounding box regression, and classification into one network. All of these make the overall performance greater with improvement in speed

(ii) Cascade R-CNN [7] is improved based on the Faster R-CNN. Faster R-CNN has only one R-CNN network, while Cascade R-CNN cascades multiple R-CNN networks based on different IoU thresholds to continuously optimize detection results

(iii) PP-YOLOv2 [9] is more industrialized object detection network, compared with Faster R-CNN and Cascade R-CNN. Starting from PP-YOLO, it is improved by gradually adding modules that contribute to performance improvement without increasing reasoning time through incremental ablation. Its high precision and high speed makes it competitive

Table 1: Eight object categories and their numbers of images and objects in forward sonar data.

(a) Object information in training data

Object category	Number of objects	Number of images
Ball	1943	1941
Circle cage	386	383
Cube	1752	1749
Cylinder	402	401
Human body	684	683
Metal bucket	403	402
Square cage	655	655
Tyre	852	850
Total	7077	4000

(b) Object information in test data

Object category	Number of objects	Number of images
Ball	595	595
Circle cage	86	86
Cube	424	423
Cylinder	39	39
Human body	379	379
Metal bucket	44	43
Square cage	169	169
Tyre	108	108
Total	1844	1000

4.2.2. Parameter Settings. In this paper, the used open-source models of Faster R-CNN, Cascade R-CNN, and PP-YOLOv2 are provided by PaddleDetection, which is the same as used for the proposed PPYOLO-T. In our experiments, regular random initialization is used for each method. The proposed BotNet-dcn used the same pretrained model as ResNet50-vd-dcn used in PP-YOLOv2. Ideally, it is better to retrain a model for BotNet-dcn, but we did not in considering the needs of huge computing power and time consumption. Our experimental results show that even using the not ideal pretrained model, the proposed method performs best on sonar object detection task. For Faster R-CNN and Cascade R-CNN, we used the pretrained model provided by PaddleDetection for each backbone networks. A small batch of 8 images was distributed on 2 GPUs. Other parameters can be found in the score code.

4.2.3. Evaluation Metrics. In this paper, the performance of the proposed model is evaluated mainly by mean average precision (mAP). It is a quantitative indicator for evaluating the effectiveness of multicategory object detection. The calculation formula is as follows:

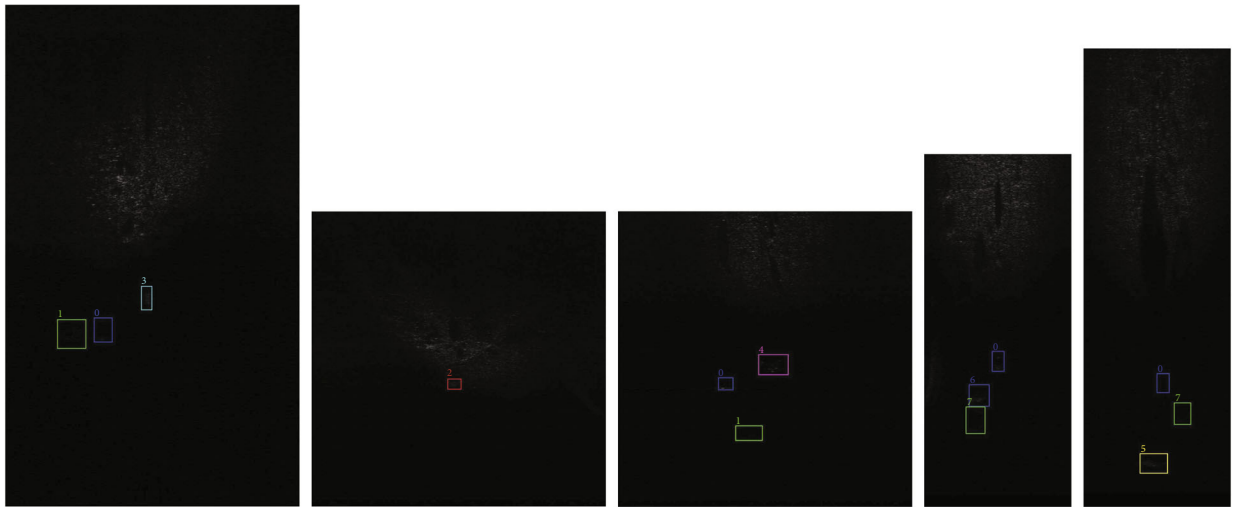

Figure 7: Object examples from each category: 0 ball, 1 circle, 2 cubes, 3 cylinders, 4 human bodies, 5 metal buckets, 6 square cages, and 7 tyres.

$$\mathrm{AP} = \sum_{r=0}^{1} (r_{n+1} - r_n) P_{\mathrm{interp}}(r_{n+1}), \tag{8}$$

with

$$P_{\mathrm{interp}}(r_{n+1}) = \max_{\tilde{r}:\tilde{r} \geq r_{n+1}} P(\tilde{r}), \tag{9}$$

where $P(\tilde{r})$ is the measured precision at recall $\tilde{r}$, r taking the maximum precision whose recall value is greater or equal than r_{n+1}.

Intersection-over-union (IoU) is an indicator based on the Jaccard similarity coefficient and evaluates the overlap between two bounding boxes. It can measure the regression precision of object detection. The formula of IoU is as follows:

$$\mathrm{IoU} = \frac{S_{\mathrm{overlap}}}{S_{\mathrm{union}}}. \tag{10}$$

In which, S_{overlap} is the overlap area between the predicted box and the ground truth box, and S_{union} is the joint area of the predicted box and the ground truth box.

For each object detection, if the result matches some ground truth box with IoU > 0.5, we mark it as positive, otherwise mark it as negative. Calculated on this basis, the resulting mean average precision is marked as $\mathrm{AP}_{0.5}$. And so on, we can get $\mathrm{AP}_{0.75}$ and $AP_{0.95}$. What is more, the average of $\mathrm{AP}_{0.5}$ to $\mathrm{AP}_{0.95}$, in which IoU increases by 0.5 each time, is denoted as $\mathrm{AP}_{(0.5:0.95)}$ in this paper.

Apart from the mAP, we also test inference time and FPS for each model, to analyze the real-time performance of different models. The inference time is the time required by the algorithm to process each image, and FPS means the number of frames per second processed by some algorithm.

4.3. Ablation Experiments for PPYOLO-T. In order to test the effect of the proposed model, we conducted ablation experiments on forward sonar object detection, as shown in Table 2. We present the effectiveness of each module in an incremental manner. Inference time and FPS are different from those in YOLOv4 [20] and PP-YOLO [4], where decoding and NMS inference are not considered, but all inference processes are added in our experiments.

Table 2: Ablation experiments for the proposed method on forward sonar data set.

	Method	$\mathrm{AP}_{(0.5:0.95)}$ (%)	Info time (ms)	FPS (f/s)
A	PP-YOLOv2	49.36	31.1	32.15
B	A + mosaic+resize anchor	50.54	31.1	32.15
C	B + BotNet-dcn	51.43	35.6	28.14
D	C + decoupled head	52.63	38.5	26.18
E	D + image size 768	53.14	41.2	24.27
F	E + BotNet-dcn 101	**53.51**	50.5	19.82

(i) *A*. First of all, we follow the original design of PP-YOLOv2 to build our baseline. Due to the large difference between sonar image and image in COCO data set, Mixup and other preprocessing methods such as brightness adjustment acting on COCO data set are not useful for improving the accuracy of sonar object detection and will increase the CPU preprocessing time. Therefore, Mixup and brightness-related preprocessing are deleted in this test. Finally, the performance of PP-YOLOv2 is shown in the first line of Table 2, where the average AP from $\mathrm{AP}_{0.5}$ to $\mathrm{AP}_{0.95}$ is 49.6%, the reasoning speed was 31.1 ms, and the FPS is 32.15 f/s

(ii) $A \longrightarrow B$. For forward sonar images, we used mosaic for data augmentation. Other preprocessing is the same with *A*. In addition, since the object size of sonar image is too different from that of COCO data set, we use K-means [37] algorithm to resize

Table 3: Ablation experiment for MHSA and DCN position.

Backbone	c5-DCN	$AP_{(0.5:0.95)}$ (%)	$AP_{0.5}$ (%)	$AP_{0.75}$ (%)
BotNet50	(0, 0, 0)	52.2	91.7	51.3
The first stage	(1, 0, 0)	51.7 (-0.5)	92.3 (+0.6)	52.3 (+1.0)
The middle stage	(0, 1, 0)	51.9 (-0.3)	92.7 (+1.0)	51.1 (-0.2)
The last stage	(0, 0, 1)	**52.6 (+0.4)**	89.6 (-2.1)	**56.4 (+5.3)**

anchor box based on training data, in which the cluster number K is set to the number of objects. As can be seen from Table 2, the precision in terms of $AP_{(0.5:0.95)}$ increased by 1.18%. This indicates that it is necessary to select the preprocessing method according to the characteristics of sonar image. Meanwhile, mosaic and anchor resizing are useful selections

(iii) $B \longrightarrow C$. The second refinement with a positive effect on PP-YOLOv2 that we found was BotNet-dcn. We attempted to add attention mechanisms to the PP-YOLOv2 backbone network. In consideration of efficiency and accuracy, we chose BotNet as the backbone and replaced its MHSA in the last stage with DCNs as shown in Figure 5. The reason why we replace MSHA in the last stage will be shown through the following experiment. To this end, BotNet-dcn boosts $AP_{(0.5:0.95)}$ performance from 50.54% to 51.43%

(iv) $C \longrightarrow D$. Decoupled head is the third refinement with a positive effect. Compared with couple head used in PP-YOLOv2, the proposed decoupled head takes into account the differences in the content concerned with classification and positioning. Therefore, using different branches for computations are conducive to effect improvement. By leveraging decoupled head, the accuracy in terms of $AP_{(0.5:0.95)}$ was improved by a further 1.2%

(v) $D \longrightarrow E$. Underwater sonar detectors usually detect objects in a large area, so there are many small objects in the generated sonar images. The image size of PP-YOLOv2 is 640, since larger image size can result more anchors. Considering that meticulous anchors will benefit small objects, we increased image size from 640 to 768. The $AP_{(0.5:0.95)}$ performance further increased to 0.5%.

(vi) $E \longrightarrow F$. Considering the low SNR of sonar images, we try to improve feature extraction ability of backbone network by deepening network structure. In this test, we replaced the original BotNet-dcn-50 with BotNet-dcn-101. The performance further increased by 0.37% $AP_{(0.5:0.95)}$

(vii) One can see that from $A \longrightarrow F$, with the continuous improvement of $AP_{(0.5:0.95)}$ performance, the reasoning speed is decreasing. Although each change is slightly slower than original PP-YOLOv2, such a significant gain promotes us to adopt them in our final model. For more details, please refer to our code. It is also worth noting that from $E \longrightarrow F$, deepening backbone network improved the accuracy slightly but reduced the efficiency significantly. Therefore, it is not recommended to stack more networks deeply

4.4. Position Selection for DCN. In this subsection, we used the original BotNet50 as the baseline and tested the effect of different DCN replacement positions, respectively. As shown in Table 3, the performance varies with the different DCN positions. On the whole, DCN in the last stage boosted $AP_{(0.5:0.95)}$ performance by 0.4%, while DCN in other positions leaded to declines in the performance. Specifically, A $P_{0.5}$ performance increased when DCN was in the first two stages but decreased in the last stage. DCN in all positions have positive effect on $AP_{0.75}$ performance, and that of at last stage performed the best with a 5% improvement. We finally chose to replace MHSA with DCN in the last stage based on the whole performance.

4.5. Comparison with Other Popular Detectors. We compared the proposed PPYOLO-T with PP-YOLOv2, Faster R-CNN, and Cascade R-CNN. To show the effectiveness of our proposed BotNet-dcn, we compared them with different backbones such as ResNet, Swin-tiny, and BotNet. Comparison of the results on sonar data set with other state-of-the-art object detectors is shown in Table 4.

From Table 4, we draw the following observations:

(i) *Comparison between YOLO and R-CNN*. We can see that on the task of sonar object detection, models from the YOLO series are superior to R-CNN based models, both in terms of accuracy and efficiency

(ii) *Backbone comparison*. ResNet got the fastest inference speed but the lowest accuracy. Through partially adding attention mechanisms, the proposed BotNet-dcn boosted the performance significantly at the expense of slight reasoning efficiency reduction. Swin-tiny, which is purely superimposed with attention mechanisms, hurts dramatically the reasoning efficiency while the accuracy is not superior to the proposed BotNet-dcn

(iii) *Image size comparison*. We can see that by expanding the image size of PPYOLO-T from 640×640 to 768×768, the detection accuracy is further

Table 4: Comparison of the speed and accuracy of different object detectors on forward sonar data set.

Method	Size	Backbone	Infer time (ms)	FPS (f/s)	$AP_{(0.50:0.95)}$ (%)	$AP_{0.5}$ (%)	$AP_{0.75}$ (%)
Cascade R-CNN	800 × 1333	ResNet50	80.97	12.35	46.83	86.57	47.53
		Swin-tiny	101.13	9.87	48.75	89.13	46.37
		BotNet50-dcn	91.82	10.88	47.63	87.36	45.18
Faster R-CNN	800 × 1333	ResNet50	75.64	13.22	45.78	84.57	43.58
		Swin-tiny	97.38	10.27	49.80	89.3	47.92
		BotNet50-dcn	85.91	11.64	46.51	87.63	47.63
PP-YOLOv2	640 × 640	ResNet50-dcn	31.10	32.15	50.54	89.26	49.26
		Swin-tiny	38.17	25.53	51.14	90.47	51.64
		BotNet50-dcn	35.53	28.14	51.43	91.26	49.31
PPYOLO-T	640 × 640	ResNet50-dcn	33.48	29.87	51.37	92.31	50.19
		Swin-tiny	43.16	23.17	52.34	91.63	51.62
		BotNet50-dcn	38.19	26.18	52.63	89.38	56.43
PPYOLO-T	768 × 768	ResNet50-dcn	45.18	22.13	52.36	91.38	51.67
		Swin-tiny	75.47	13.25	52.96	92.64	52.47
		BotNet50-dcn	54.73	18.27	53.14	90.88	55.37

Table 5: Detection precision, recall, and F1-score for different target categories when IoU is equals to 0.5.

Category	Precision	Recall	$F1$
Ball	96.1	98.5	97.3
Cube	96.0	98.6	97.3
Tyre	89.1	97.2	92.9
Circle cage	89.8	94.2	91.9
Human body	90.5	93.4	91.9
Square cage	88.8	94.1	91.4
Metal bucket	89.6	93.2	91.4
Cylinder	76.5	94.9	84.7
Average	89.6	95.5	92.4

improved. We find that most of the objects in underwater sonar images are small. The larger image size, the more prediction boxes. We think that is why enlarging image size works. It also can be seen that large image size damages the speed. But compared with 800 × 1333, which is the best size used in models of R-CNN series, the proposed PPYOLO-T used a smaller image size and got better mAP performance

4.6. Performance on Different Objects. In this subsection, we further evaluate the performance of the proposed PPYOLO-T on different objects. Detection precision, recall, and $F1$ -score are used as the evaluation matrix. In this evaluation, we set the IoU to 0.5 which is commonly used in real application. Table 5 shows the detailed values of each category. Figure 8 further shows the P-R curve.

We can see that the overall performance of our PPYOLO-T is very well on the task of sonar image multiobject detection. The average precision is 89.6%, and the average recall is over 95%, such that the F_1 score is up to 92%.

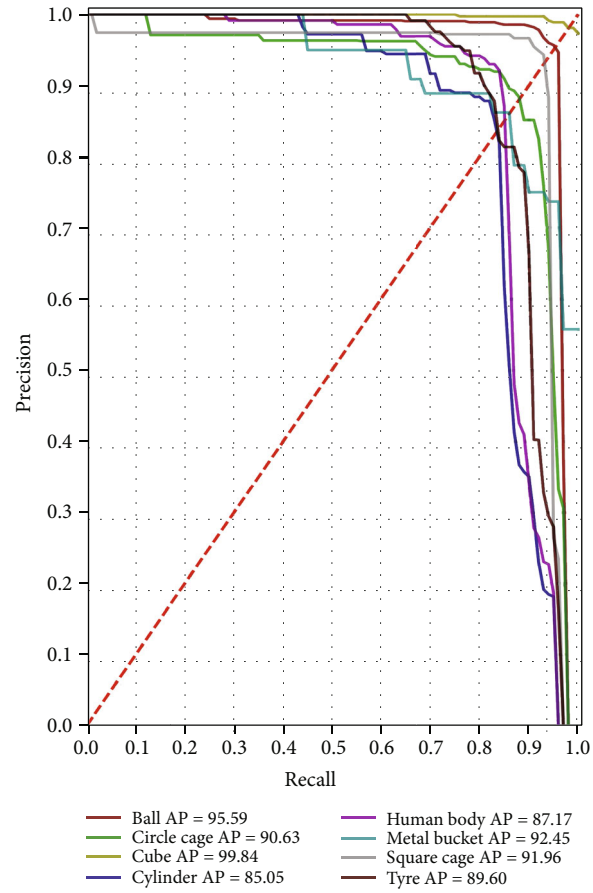

Figure 8: Precision-recall curve for different objects.

From the P-R curve in Figure 8, we can see the similar results. Specifically, it performs well categories of cube, ball, metal bucket, square cage, and circle cage but not very well on cylinder and human body categories. This may because

the shape of human body and cylinder in underwater sonar images is relatively irregular. There are also many areas without objects in GT boxes. This may lead to the IoU of the prediction box and the real box less than the threshold. In this case, it will be regarded as a negative sample, and thus has a negative effect on model learning.

5. Conclusions

This paper presents some useful preprocessing methods for sonar image and some updates to PP-YOLOv2, which forms a high-performance sonar image multiobject detector called PPYOLO-T. By introducing attention mechanism and decoupled head, PPYOLO-T achieves significant improvement of detection accuracy with slightly speed reduction. Compared with state-of the- art models of R-CNN series, it achieves the best speed and accuracy. However, there still are some interesting future work. For example, we can further optimize some structures of attention mechanism to improve detection speed following the most recent work [38].

Data Availability

All the data are available in these following links: (1) https://code.ihub.org.cn/companies/vgz4xa2q; (2) https://github.com/violetweir/PPYOLO-T/tree/main/dataset; (3) https://github.com/PaddlePaddle/PaddleDetectionhttps://github.com/violetweir/PPYOLO-T.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the General Project of Science and Technology Plan of Beijing Municipal Education Commission (No. KM202210017006), the 2021-2023 Young Talents Promotion Project of Beijing Association for Science and Technology, the research project of Digital Education in Beijing (BDEC2022619048), the Natural Science Foundation of Ningxia (2022AAC03757), the Cross-Disciplinary Science Foundation from Beijing Institute of Petrochemical Technology (No. BIPTCSF-006), and the Teaching Reform project of Beijing Institute of Petrochemical Technology (Nos. ZDFSGG202103001, ZDKCSZ202103002, and ZDKCSZ202203004).

References

[1] V. Myers and J. Fawcett, "A template matching procedure for automatic target recognition in synthetic aperture sonar imagery," *IEEE Signal Processing Letters*, vol. 17, no. 7, pp. 683–686, 2010.

[2] J. Groen, E. Coiras, and D. Williams, "Detection rate statistics in synthetic aperture sonar images," in *Proceedings of the 3rd International Conference and Exhibition on Underwater Acoustic Measurements: Technologies and Results*, Nafplion, Greece, 2009.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[4] X. Long, K. Deng, G. Wang et al., "Ppyolo: an effective and efficient implementation of object detector," 2020, https://arxiv.org/abs/2007.12099.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, realtime object detection," in *In Proceedings of The Ieee Conference on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[7] Z. Cai and N. Vasconcelos, "Cascade RCNN: delving into high quality object detection," in *Proceedings of the Ieee Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, USA, 2018.

[8] J. Redmon and A. Farhadi, "YOLOV3: An incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[9] X. Huang, X. Wang, W. Lv et al., "PPYOLOV2: a practical object detector," 2021, https://arxiv.org/abs/2104.10419.

[10] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Y. Cutmix, "Regularization strategy to train strong classifiers with localizable features," in *In Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

[11] E. Dura, Y. Zhang, X. Liao, G. J. Dobeck, and L. Carin, "Active learning for detection of mine-like objects in side-scan sonar imagery," *IEEE Journal of Oceanic Engineering*, vol. 30, no. 2, pp. 360–371, 2005.

[12] D. P. Williams and E. Fakiris, "Exploiting environmental information for improved underwater target classification in sonar imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 10, pp. 6284–6297, 2014.

[13] D. Neupane and J. Seok, "A review on deep learning-based approaches for automatic sonar target recognition," *Electronics*, vol. 9, no. 11, p. 1972, 2020.

[14] H. T. Nguyen, E.-H. Lee, C. H. Bae, and S. Lee, "Multiple object detection based on clustering and deep learning methods," *Sensors*, vol. 20, no. 16, p. 4424, 2020.

[15] M. Valdenegro-Toro, "End-to-end object detection and recognition in forward-looking sonar images with convolutional neural networks," in *In 2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pp. 144–150, 2016.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[18] S. M. Mark Everingham, L. EslamiVan, C. K. I. Gool, J. W. Williams, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[19] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft COCO: common objects in context," in *13th European Conference on Computer Vision*, pp. 740–755, Zurich, Switzerland, 2014.

[20] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOV4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[21] X. Zhu, S. Lyu, W. Xu, and Q. Zhao, "TPH-YOLOV5: improved YOLOV5 based on transformer prediction head for object detection on drone-captured scenarios," in *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2778–2788, 2021.

[22] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *In Proceedings of The Ieee Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271, 2017.

[23] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: exceeding YOLO series in 2021," 2021, https://arxiv.org/abs/2107.08430.

[24] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, 2019.

[25] S. Liu, Q. Lu, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, USA, 2018.

[26] D. Misra, "Mish: A Self Regularized Non-Monotonic Neural Activation Function," 2019, https://arxiv.org/abs/1908.08681.

[27] K. Krishna, M. Narasimha, and Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[28] W. Hao and S. Zhili, "Improved mosaic: algorithms for more complex images," *Journal of Physics: Conference Series*, vol. 1684, article 012094, 2020.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," 2017, https://arxiv.org/abs/1710.09412.

[30] F. He, T. Liu, and D. Tao, "Why ResNet works? Residuals generalize," *IEEE Transactions On Neural Networks And Learning Systems*, vol. 31, no. 12, pp. 5349–5362, 2020.

[31] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[32] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *In Proceedings of The Ieee/Cvf Conference on Computer Vision and Pattern Recognition*, pp. 16519–16529, 2021.

[33] Z. Liu, Y. Lin, Y. Cao et al., "Swin transformer: hierarchical vision transformer using shifted windows," in *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, Montreal, Canada, 2021.

[34] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: fully convolutional one-stage object detection," in *In Proceedings Of The Ieee/Cvf International Conference On Computer Vision*, pp. 9627–9636, 2019.

[35] Y. Wu, Y. Chen, Y. Lu et al., "Rethinking classification and localization for object detection," in *In Proceedings of the Ieee/Cvf Conference on Computer Vision and Pattern Recognition*, pp. 10186–10195, Seattle, WA, USA, 2020.

[36] S.-W. Kim, H.-K. Kook, J.-Y. Sun, M.-C. Kang, and S.-J. Ko, "Parallel feature pyramid network for object detection," in *In Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 234–250, Munich, Germany, 2018.

[37] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[38] M.-H. Guo, C.-Z. Lu, Z.-N. Liu, M.-M. Cheng, and S.-M. Hu, "Visual attention network," 2022, https://arxiv.org/abs/2209.00224.