IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 July 2024, accepted 7 August 2024, date of publication 14 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443337

RESEARCH ARTICLE

Detection, Tracking and Enumeration of Marine Benthic Organisms Using an Improved YOLO+DeepSORT Network

JIAN LIU[1], QIAN LI[1], SHANTAO SONG[1], AND KALIYEVA KULYASH[2]

[1]Institute of Automation, Qilu University of Technology (Shandong Academy of Sciences), Shandong Provincial Key Laboratory of Robot and Manufacturing Automation Technology, Jinan 250014, China

[2]Department of Mathematics, Al-Farabi Kazakh National University, 050040 Almaty, Kazakhstan

Corresponding author: Qian Li (261232784@qq.com)

This work was supported in part by the Qilu University of Technology (Shandong Academy of Sciences) Major Innovation Special Project under Grant 2023HYZX01, and in part by Shandong Provincial Science and Technology Innovation Project under Grant 2022TSGC2460.

ABSTRACT For marine ranching, efficiently and accurately detecting, tracking, and enumeration of benthic organisms can help farmers understand the growth and population changes of marine products, avoid high-risk tasks, and analyze changes in the marine ecological environment. To address the problems of target occlusion, low detection accuracy, and numerous small targets in existing marine organism detection models in complex seabed environments, an improved YOLOv5+DeepSORT algorithm for detecting and tracking benthic organisms is proposed. This algorithm integrates the Global Context Block attention mechanism with the BottleneckCSP module to form a new BottleneckCSPGC module, enhancing feature extraction capabilities. Replace the original loss function with the Normalized Wasserstein Distance (NWD) loss function to improve the detection accuracy of small targets. Finally, experimental results show that the accuracy on the underwater dataset reached 87.1% mAP@0.5 and 53.3% mAP@0.5:0.95, which are 1.8% and 4.0% higher than YOLOv5, respectively. The use of DeepSORT for tracking and counting provides technical support for marine ranching supervision.

INDEX TERMS Benthic organisms, YOLOv5, DeepSORT, global context block, NWD loss function.

I. INTRODUCTION

The ocean is a treasure trove of resources and strategic space that supports future development [1]. Benthic organisms play a crucial role in marine ecosystems [2], and efficient and accurate identification and counting of these organisms can help analyze changes in the marine ecosystem, enabling timely responses. Currently, the identification and counting of benthic organisms primarily rely on manual observation or analyzing underwater images, methods that are both hazardous and inefficient. Therefore, it is imperative to find a fast, accurate, and safe alternative to manual observation and counting.

With the rapid advancement of deep learning and computer vision, more and more researchers are focusing on using underwater robots for visual environment perception [3]. Further tasks such as object detection and tracking, like drainage pipe defect detection [4] and underwater moving target detection with artificial lateral line systems [5], are also being conducted. In terms of object detection, Xiao et al. [6] proposed a lightweight feature extraction module called GGS, which combines traditional down-sampling algorithms with depthwise separable convolution down-sampling, and introduces a parameter-free attention mechanism to extract multi-scale features from input data, focusing on target information. Compared to the YOLOv5s source code, their model reduces parameter size by 94%, doubles detection speed, and the weight file is only 1.08M, 92% smaller, the disadvantage is that the detection accuracy has decreased. Liu A new YoLoWaternet (YWnet) model was proposed by Liu et al. [7],which introduces a Convolutional Block Attention Module (CBAM) to enhance feature extraction from blurred images in the initial stages of the network. They also created a new feature fusion network called CRFPN to convey

The associate editor coordinating the review of this manuscript and approving it for publication was Xuebo Zhang.

important information and detect underwater objects. This model features a new type of feature extraction module, the Skip Residual C3 module (SRC3), and employed a decoupled head to separate the regression and classification tasks, improving detection efficiency. The EIoU loss function was used to accelerate convergence. Experiments show that the YWnet algorithm achieves a detection accuracy of 73.2% mAP and 39.3% mAP50-95 on underwater datasets, representing an improvement of 2.3% and 2.4% over YOLOv5, but the downside is that the increase in parameter size has slowed down the model's inference speed. In terms of object tracking, Azhar et al. [8] used the Deep SORT framework to create a people tracking system for crowd monitoring. Unlike object detection frameworks like CNN, this system can not only detect people in real time but also use learned information to track their trajectories until they leave the camera's frame. The system uses YOLO for people detection and then processes detected people frame-by-frame using Deep SORT to predict their movement paths. The system can successfully detect and track the movement paths of individuals at an average speed of 2.59 frames per second (FPS). However, the drawback is that the low frame rate may cause the video stream to appear discontinuous, affecting the viewing experience, especially in scenarios requiring real-time monitoring. Qiu et al. [9] proposed a pedestrian counting scheme based on YOLOv5 and DeepSORT for multi-object detection and tracking. Using network weights trained on the COCO dataset, they combined the YOLOv5 detector and DeepSORT tracker to detect and track pedestrians, counting the number of entries and exits to control the number of floating individuals. Experiments conducted in streets and subway stations demonstrated that this algorithm is suitable for tracking and counting in high-density crowds, providing high system accuracy and robustness while maintaining real-time performance. The disadvantage is that it may perform poorly in other more complex or special scenarios, such as at night or under extreme weather conditions.

Based on the aforementioned research, this paper proposes an improved YOLOv5+DeepSORT algorithm for the detection and tracking of marine benthic organisms. The algorithm effectively addresses issues present in existing marine organism detection models under complex seabed environments, such as target occlusion, low detection accuracy, and the prevalence of small targets. The improvements in target detection focus on two main aspects:

Firstly, by integrating the Global Context Block (GC) attention mechanism with the BottleneckCSP module, forming a new BottleneckCSPGC module, we enhance the feature extraction capability, thereby improving the model's ability to detect occluded targets. Secondly, we replace the original loss function with the Normalized Wasserstein Distance (NWD) loss function, which increases the detection accuracy for small targets. Building on this foundation, DeepSORT is used to track benthic organisms, with an additional feature for counting their numbers.

II. RELATED WORK

A. OBJECT DETECTION ALGORITHMS

The development of object detection technology based on convolutional neural networks (CNNs) has accelerated recently, surpassing traditional methods in many detection applications [10]. These technologies can be broadly categorized into two types: Single-stage (One-Stage) and two-stage (Two-Stage) object detection algorithms represent two main approaches in the field. The Regions with Convolutional Neural Networks Features (R-CNN) series is a typical example of two-stage object detection algorithms [11], such as Fast R-CNN [12], Faster R-CNN [13], and Mask R-CNN [14]. These algorithms are known for their high detection accuracy but relatively slow detection speed. On the other hand, The Anchor-Free series [15], Single Shot MultiBox Detector (SSD) series [16], and You Only Look Once (YOLO) series [17] are representative of one-stage object detection algorithms. While these detection algorithms generally have lower accuracy, they are known for their fast detection speed and are widely used in the industry.

In 2014, the R-CNN (Regions with Convolutional Neural Networks) was proposed for two-stage object detection, marking the entrance of object detection into the deep learning era. However, due to its long computation time, it was not practical for real-world applications. Building on this foundation, researchers subsequently introduced algorithms such as SPP-Net, Fast R-CNN, and Faster R-CNN, which improved computation speed by over a hundred times, making them viable for practical use. As these algorithms were increasingly used, their limitations became more apparent, particularly their poor detection capabilities for small objects. To address this, researchers developed various methods based on feature fusion, such as FPN, Cascade R-CNN, and M2Det, significantly enhancing small object detection performance in images.

To overcome the long computation time associated with two-stage object detection algorithms, a one-stage object detection algorithm based on the YOLO series was proposed. Over time, YOLOv2, YOLOv3, and YOLOv4 gradually addressed the issues of inaccurate bounding box localization, poor small object detection performance, and low algorithm accuracy found in YOLOv1. However, the introduction of the anchor mechanism in YOLOv2 brought new challenges, such as difficult parameter settings and severe imbalance in the positive and negative sample ratio. To tackle these issues, researchers proposed a series of anchor-free algorithms including CornerNet, CenterNet, FCOS, and FoveaBox, which further improved the accuracy and speed of these algorithms.

B. OBJECT TRACKING ALGORITHMS

Object tracking algorithms are a category of techniques used to detect and track the location and movement trajectories of specific targets (such as people, vehicles, animals, etc.)

in video sequences. These algorithms are widely used in fields such as surveillance, security, autonomous driving, and robotics.

1) SINGLE OBJECT TRACKING (SOT)

KLT (Kanade-Lucas-Tomasi) Tracking Algorithm: Based on optical flow, it utilizes feature points of the target for tracking. Suitable for small range movements and scenes with minimal lighting changes, its advantages include low computational requirements and strong real-time performance. However, it is not robust against occlusion and significant target movements.

CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability): A target tracking algorithm based on correlation filtering that enhances tracking robustness through channel and spatial reliability. Its strengths lie in handling target deformation and lighting changes, but its disadvantages include high computational complexity, making it unsuitable for real-time tracking.

2) SIAMESE NETWORK

A twin network based on deep learning, it uses similarity measurements for target tracking, such as SiamFC, SiamRPN, etc. Its advantages are its effectiveness in handling complex scenes and variable targets, but it requires a large amount of data for training and high computational resources.

3) MULTI-OBJECT TRACKING (MOT)

SORT (Simple Online and Realtime Tracking): A simple online multi-object tracking algorithm that combines a Kalman filter and the Hungarian algorithm for data association. Its advantages include simplicity in implementation and fast computation speed, making it suitable for real-time applications. However, it is not robust to changes in target appearance and occlusion.

4) DEEPSORT

Builds on SORT by adding deep learning features to utilize target appearance information for more accurate association. It is robust against target occlusion and appearance changes, suitable for high-density scenarios. However, it has the disadvantage of high computational complexity and substantial hardware requirements.

5) IOU TRACKER

A simple multi-object tracking method based on the intersection over union (IoU) of target bounding boxes. Its advantages are simplicity in implementation and fast computation speed. Its disadvantage is that it handles target occlusion and overlapping poorly.

C. PRINCIPLES OF YOLOv5 OBJECT DETECTION ALGORITHM

YOLOv5 uses deep convolutional neural networks (CNNs) as feature extractors, transforming input images into feature maps that represent various characteristics of the images. Anchor boxes, which are predefined rectangular boxes, are then used to predict the positions and classes of the objects. The algorithm predicts the offsets for each anchor box and the class probabilities of the objects. These predictions are made using the feature maps output by the network. For each anchor box, the algorithm predicts the bounding box and the class probability of the target. The final object detection results are obtained through decoding and post-processing these predictions. YOLOv5 uses a loss function called YOLO Loss to measure the disparity between the predicted results and the ground truth labels. This loss function includes aspects such as localization loss and class probability loss. The model is optimized by minimizing this loss function [18]. YOLOv5 has released four different versions of object detection networks: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x [19]. The structures of these four models are essentially identical, with the primary differences being the depth and width parameters, depth_multiple and width_multiple, respectively. In this paper, the YOLOv5s version is used to improve the object detection algorithm, and DeepSORT is used to improve the object tracking statistics.

D. PRINCIPLES OF THE DEEPSORT OBJECT TRACKING ALGORITHM

DeepSORT is a multi-object tracking algorithm based on deep learning and is an improved version of SORT (Simple Online and Realtime Tracking) [20]. DeepSORT uses an already tracked targets from the previous frame, thus achieving multi-object tracking.

In DeepSORT, a deep neural network is used to extract features of the targets. The input to this network is the image region of a target, and the output is a vector representing the target's features. DeepSORT also uses a convolutional neural network (CNN) for feature extraction of the targets and employs the Hungarian algorithm to associate the detection results in the current frame with the Kalman filter to estimate the positions and velocities of the targets, reducing errors during the tracking process.

In each frame, DeepSORT first extracts features from the detection results using a CNN and calculates the similarity between all the detection results in the current frame and the already tracked targets from the previous frame. Then, the Hungarian algorithm is used to associate the detection results in the current frame with the already tracked targets from the previous frame. Finally, new targets are created for unmatched detection results, and the existing targets are updated during the tracking process.

Using DeepSORT can achieve accurate tracking of marine benthic organisms, which is crucial for studying their behavior, interactions, and movements over time. Additionally, traditional monitoring of benthic organisms requires a significant amount of manual labor and time. DeepSORT automates this process, allowing for continuous and long-term monitoring without the need for constant human intervention. This automation is particularly important in harsh and remote

underwater environments. Moreover, using DeepSORT to track and count marine benthic organisms helps researchers understand changes in population numbers within a marine area and assess the impact of environmental changes such as pollution, climate change, and habitat destruction.

III. MATERIALS AND METHODS

The original YOLOv5s algorithm model's network structure and loss function are designed for general scenarios, resulting in suboptimal performance in certain specific applications. However, due to its clear and simple network structure and relatively low detection accuracy, there is significant potential for improvement. Therefore, this paper proposes modifications to YOLOv5s to address the challenges of detecting benthic organisms in complex underwater environments. Additionally, a counting function is added to DeepSORT to facilitate more comprehensive tracking and analysis.

A. IMPROVED YOLOv5S ALGORITHM

1) GLOBAL CONTEXT BLOCK ATTENTION MECHANISM

The Global Context (GC) block originates from the GCNet paper, and its core idea is to utilize non-local information for modeling. This allows the block to extract relevant information from the global context, thereby enhancing the model's feature extraction capability [21]. This allows the model to capture global contextual relationships (or features), thereby enhancing its feature extraction capabilities. The Global Context Block combines the benefits of a simplified non-local (SNL) block and a lightweight computational squeeze-and-excitation (SE) block. Additionally, by incorporating the structure of SENet, this method significantly reduces the model's computational load, making it a plug-and-play, lossless module.

In the simplified non-local (SNL) block, shown in Figure 1(a), the transformation module contains the most parameters, including a 1×1 convolution with $C \times C$ parameters. When this SNL block is added to higher layers, the number of parameters in this 1×1 convolution, $C \times C = 2048 \times 2048$, dominates the parameter count of this block. The SE block, shown in Figure 1(b), achieves lightweight characteristics by replacing the 1×1 convolution with a bottleneck transformation module, significantly reducing the parameter count from $C \times C$ to $2 \times C \times C/r$, where r is the bottleneck ratio, and C/r is the hidden representation dimension of the bottleneck. By setting the initial compression rate to $r = 16$, the number of parameters in the transformation module is reduced to 1/8 of the original SNL block. Due to the increased optimization difficulty caused by the two-layer bottleneck transformation, layer normalization is added inside the bottleneck transformation (before ReLU) to simplify optimization and act as a regularizer to improve generalization.

The detailed architecture of the Global Context (GC) block is shown in Figure 1(c), The formula is shown in (1):

$$\mathbf{z}_i = \mathbf{x}_i + W_{v2}\mathrm{ReLU}(\mathrm{LN}(W_{v1}\sum_{j=1}^{N_p}\frac{e^{W_k\mathbf{x}_j}}{\sum_{m=1}^{N_p}e^{W_k\mathbf{x}_m}}\mathbf{x}_j)) \quad (1)$$

where $\boldsymbol{x}_i$ represents the input of the ith element, and W_{v2} is the weight matrix applied to the transformed input. $\alpha_j = \frac{e^{W_k\boldsymbol{x}_j}}{\sum_m e^{W_k\boldsymbol{x}_m}}$ is the weight of the global attention pool, $\delta(\cdot) = W_{v2}\mathrm{ReLU}(\mathrm{LN}(W_{v1}(\cdot)))$ represents a bottleneck transformation. Specifically, GC blocks include: (a) a global attention pool for contextual modeling; (b) bottleneck shifts to capture channel dependencies; (c) Broadcast element addition for feature fusion. Feature maps are represented by their feature dimensions, e.g., $C \times H \times W$ denotes a feature map with C channels, height H, and width W. $\bigotimes$ represents matrix multiplication, $\bigoplus$ represents broadcast element-wise addition, an $\odot$ represents broadcast element-wise multiplication.

2) BOTTLENECKCSPGC MODULE

The BottleneckCSP module is a typical residual block structure. It includes a convolution layer in the middle of the residual unit to reduce the number of channels, facilitating module connections and reducing computational complexity. Each residual unit in the BottleneckCSP module consists of two standard 3×3 convolution layers and one 1×1 convolution layer. Additionally, the BottleneckCSP module introduces a cross-stage partial connection (CSP) structure, which enables information transfer between different stages, thereby improving the model's performance.

Shallow features that are easily overlooked become difficult to extract when the network structure is deep. To address this, we propose a new feature extraction structure, BottleneckCSPGC, which combines the Global Context Block with the BottleneckCSP structure. This integration, illustrated in Figure 2, allows for better extraction of useful features through global relationship modeling.

Firstly, the input feature map is processed through the global attention mechanism module (GC) in the first branch. Then, it passes through a 1×1 convolution layer, reducing the number of channels to the hidden channels c_. Subsequently, it goes through a sequence of multiple Bottleneck blocks, where each Bottleneck block's input and output channels are the hidden channels c_. This is followed by a 1×1 convolution layer for channel adjustment. Simultaneously, in the second branch, the input feature map is also processed through the global attention mechanism module (GC) and then through a 1×1 convolution layer for channel adjustment. At the end of both branches, the feature maps are concatenated, and the concatenated feature map is processed through a batch normalization layer and the SiLU (Sigmoid-Weighted Linear Unit) activation function. Finally, the processed feature map is passed through another 1×1 convolution layer for channel adjustment, resulting in the final output feature map.

3) NORMALIZED WASSERSTEIN DISTANCE (NWD) LOSS FUNCTION

The default loss function in YOLOv5 is not an optimal metric for small objects. Therefore, we replace it with a new metric that measures the similarity of bounding boxes using the Wasserstein distance. The boundary box is modeled into a two-dimensional Gaussian distribution, and the similarity of the Gaussian distribution is derived by using the standardized Wasserstein distance measure, and the similarity of the Gaussian distribution is studied on this basis [22]. NWD loss function as shown in equation (2):

$$NWD\left(\mathcal{N}_a, \mathcal{N}_b\right) = \exp\left(-\frac{\sqrt{W_2^2\left(\mathcal{N}_a, \mathcal{N}_b\right)}}{C}\right) \tag{2}$$

$\mathcal{N}_a$ and $\mathcal{N}_b$ represent two bounding boxes being compared, $W_2^2(\mathcal{N}_a, \mathcal{N}_b)$ is the second-order Wasserstein distance between these two bounding boxes, and C is a normalization constant, typically set to the diagonal length of the bounding boxes to ensure that the distance scale remains within a reasonable range.

The biggest advantage of Wasserstein distance is that it measures the similarity between distributions, even when there is no overlap or very little overlap. In addition, NWD is less sensitive to objects of different scales, and is more suitable for measuring the similarity between small objects.

4) OVERALL FRAMEWORK

YOLOv5 is a fast and flexible method for object detection consisting of a backbone network, PANet, and detection

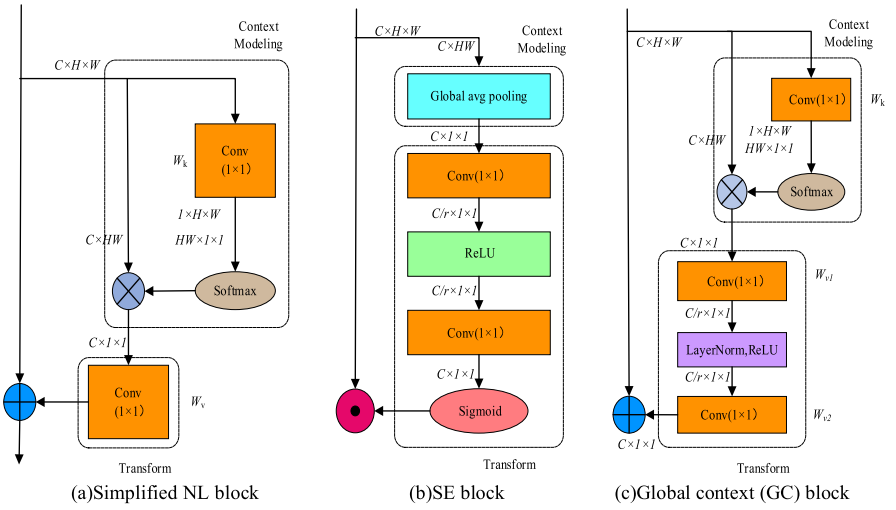

FIGURE 1. Architecture of the main modules.

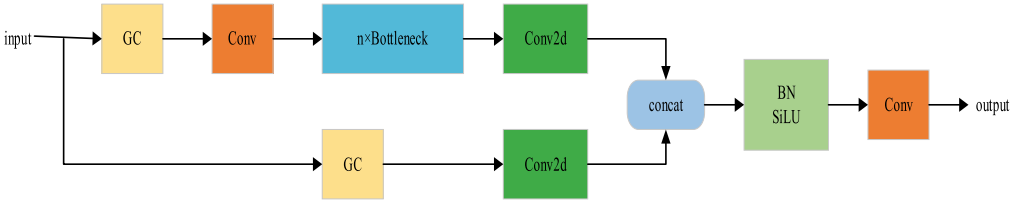

FIGURE 2. Structure of BottleneckCSPGC.

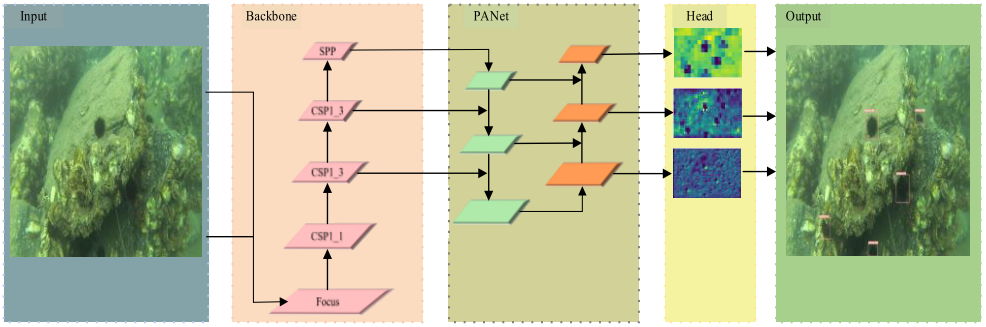

FIGURE 3. YOLOv5 network architecture.

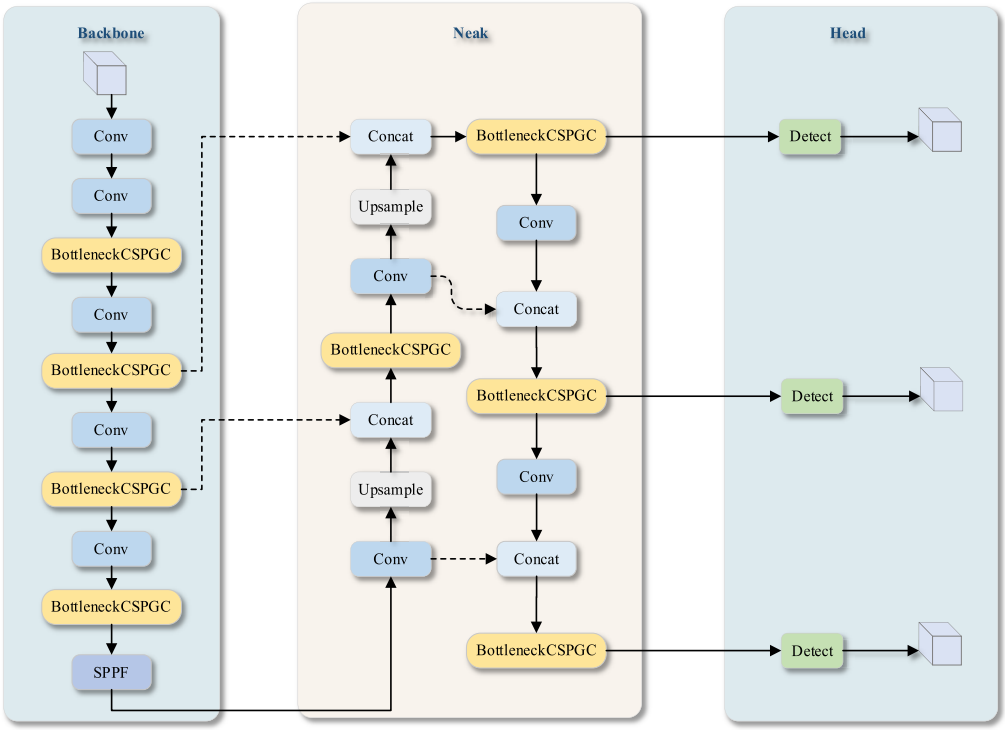

FIGURE 4. Improved network framework.

heads, as shown in Figure 3. PANet is a recurrent pyramid structure composed of convolution, upsampling, and CSP2_X modules, enabling feature fusion. The detection head is responsible for object detection and localization. By incorporating improved modules into YOLOv5, the enhanced network framework is illustrated in Figure 4.

B. STATISTICS ON THE NUMBER OF DEEPSORT TARGET TRACKS

To add a counting functionality to the existing DeepSORT tracking, we establish a dictionary to store all the IDs that have appeared. Whenever an ID appears, we check if it exists in the dictionary. If it does, the count for each category remains unchanged. If the ID does not exist, we determine which category it belongs to, increment the count for that category by one, and store the new ID in the dictionary. These steps are repeated until the end of the video. The overall flowchart for DeepSORT tracking and counting is shown in Figure 5.

IV. EXPERIMENTS AND DISCUSSIONS

A. DATASETS AND EVALUATION INDICATORS

We utilized two marine benthic organism datasets for our study. The dataset 1 is from the Underwater Robot Picking Contest, which includes images of holothurian, echinus, scallop, and starfish. Additionally, we incorporated images captured from manual observations at a Weihai aquaculture farm. These images were used for training and evaluating our object detection model. The second dataset comprises synthetic models of the aforementioned four types of organisms, which were purchased online and placed in an experimental water tank for image capture. These images were used for training and evaluating our object tracking and counting model. The dataset 1 contains 8,919 training images and 991 testing images, while the dataset 2 contains 1,078 training images and 120 testing images. We evaluated our models using the test sets and conducted ablation studies. Furthermore, we compared our results with those of other state-of-the-art models. We used the standard COCO metric, mean Average Precision (mAP), to assess the accuracy of our models.

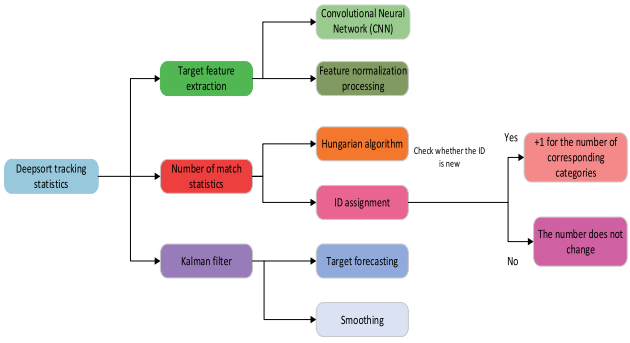

FIGURE 5. Overall flow chart of DeepSORT.

B. DETAILS OF THE EXPERIMENT

The experimental environment is shown in Table 1. Parameter Settings:

For the improved YOLOv5s network, we used the default parameter settings of YOLOv5s. We replaced the original C3 module and the loss function. The training was conducted for 200 epochs with a batch size of 8.

C. EVALUATION METRICS

In our experiments, Precision is used to evaluate the true detection capability of the model. The effectiveness of the model's detection is measured using mAP0.5:0.95 (mean Average Precision [0.5:0.95]), which represents the average precision across all detection categories at IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05. Recall is used to indicate the probability that positive samples are correctly predicted as positive. The Miss Rate (MR) is used to evaluate the extent of missed detections by the model. The F1-Score is employed to comprehensively assess the impact of Precision and Recall on the model's performance, with its value being positively correlated with the model's performance. These indicators are calculated as shown in equation (3), (4), (5) and (6) below.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{MR} = 1 - \text{Recall} = 1 - \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

$$\text{mAP} = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{5}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{6}$$

TP is the number of samples that are correctly predicted to be positive. FP is the number of samples that are marked as negative and predicted to be positive. FN is the number of samples labeled as positive and predicted as negative. APi is the average accuracy of the prediction class. N is the number of classes. In addition, FLOPs and Params, which measure the accuracy of the model, are also used to represent the amount of computation and the number of parameters of the model, respectively.

D. SET THE SPECIFIC PARAMETERS OF THE NWD LOSS FUNCTION

When we use the NWD loss function, we need to select the size of the iou_ratio, iou_ratio represents the percentage of the IoU metric that decreases the small target when it is large in the dataset, i.e., increases the NWD metric. We use the NWD loss function of different sizes and iou_ratio on the YOLOv5s source code to train dataset 1, and the training results under different parameters are shown in Table 2.

From the overall analysis in Table 2, it can be concluded that the training effect is best when iou_ratio = 0.5, so we chose it.

E. COMPARISON OF DIFFERENT LOSS FUNCTIONS

We also constructed different loss functions for comparison, and trained them on dataset 1 to test the efficiency of NWD loss [23]. A number of loss functions were also tested, including CIoU loss [24], VF loss [25], alpha-iou loss [26], and SIoU loss [27].

As can be seen from Table 3, the NWD loss function is the best fit for our model, with P, R, and mAP0.5 reaching optimal values of 86.7%, 78.2%, and 85.7%, respectively. CIoU reached 85.3% and 49.3% in mAP0.5 and mAP0.5:0.95, respectively, and SIoU reached 85.0% and 48.8% in mAP0.5 and mAP0.5:0.95, respectively. However, the results of VF and Alpha-IoU are poor, and the accuracy of mAP0.5 is only 83.4% and 83.1%, which may be due to the complex image and more information of small targets in the presence of underwater objects, resulting in poor regression convergence. This method is not affected by scale factors and is suitable for measuring the similarity between small targets. Compared to other loss functions, it is more suitable for underwater multi-small target detection.

TABLE 1. Experimental environment.

parameters	configuration
System environment	Windows10
CPU	Intel(R) Core (TM) i5-12490F
GPU	NVIDIA GeForce RTX 3060
Deep learning framework	Pytorch1.11.0
CUDA	12.5.51

TABLE 2. Training results for different parameters.

iou_ratio size	P（%）	R（%）	mAP0.5（%）	mAP0.5:0.95（%）
0.3	86.3	77.2	85.5	47.8
0.4	84.8	77.6	85.3	48.1
0.5	**86.7**	**78.2**	**85.7**	**48.5**
0.6	86.4	76.8	85.2	48.3

TABLE 3. Comparison of different loss functions.

Loss function	P（%）	R（%）	mAP0.5（%）	mAP0.5:0.95（%）
CIoU	86.6	78.0	85.3	49.3
VF	84.2	76.3	83.4	49.0
Alpha-IoU	82.8	77.0	83.1	49.1
SIoU	85.7	76.8	85.0	48.8
NWD	**86.7**	**78.2**	**85.7**	48.5

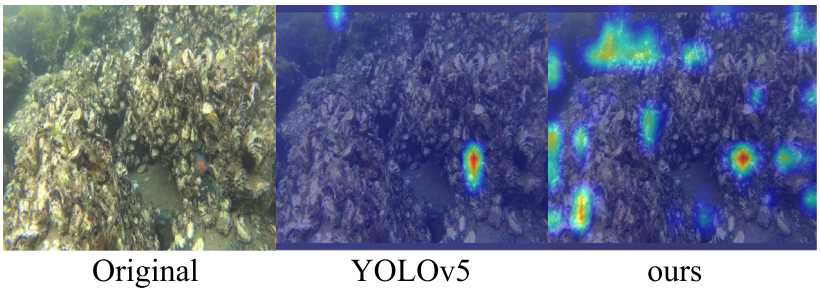

FIGURE 6. Comparison of heat map effects.

F. ABLATION EXPERIMENTS

To verify the effectiveness of the proposed algorithm's improvements based on YOLOv5, as well as the impact of the BottleneckCSPGC module and NWD loss function on the detection of small, numerous, and occluded marine benthic organisms, ablation experiments were conducted on Dataset One. The results are shown in Table 4.

Ablation experiments demonstrate that the BottleneckCSPGC module and NWD loss function, when used individually, provide varying degrees of slight improvement over the original YOLOv5 code in certain aspects. When both components are used together, the performance is even better, with increases of 1.3%, 2.9%, 1.8%, and 4% in accuracy, recall, mAP0.5, and mAP0.5:0.95, respectively, compared to the original YOLOv5. To clearly illustrate the advantages of our improved algorithm in marine benthic organism detection, Figure 6 compares the heatmap results of our improved algorithm with YOLOv5, and Figure 7 shows the test results in different scenarios selected from dataset 1.

From the comparative analysis in Figure 7, it can be seen that our network can accurately identify the missed and false detection of the YOLOv5 network caused by occlusion and low visibility, and at the same time detect some small targets, and combined with Table 4, it can be seen that the accuracy, recall rate, and average accuracy of our improved network are improved compared with the original YOLOv5.

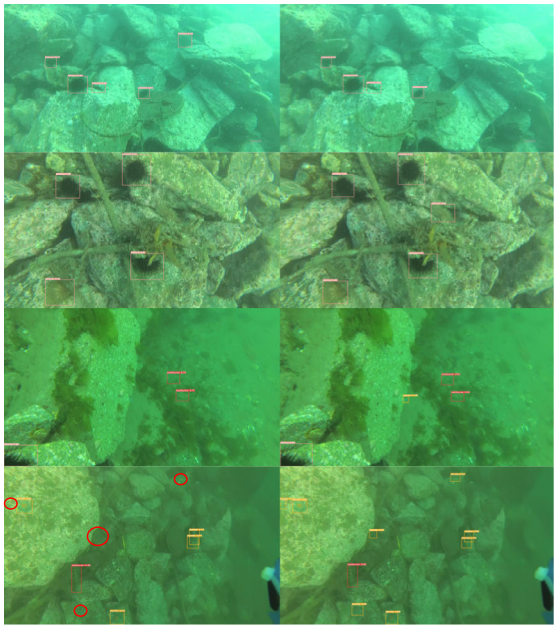

FIGURE 7. Comparison of the detection effects of the two networks.

G. COMPARATIVE EXPERIMENTS

In order to evaluate the effectiveness of the algorithm designed in this paper, it is compared with other advanced algorithms. YOLOv4 [28], YOLOv7 [29], YOLOx [30] and YOLOv5 were selected for training on dataset one, respectively.

As shown in Table 5, our proposed model outperforms other models, achieving the best performance. Compared to YOLOv4, our model's mAP0.5 increased by 29.9% and mAP0.5:0.95 increased by 24.7%. Compared to YOLOv7, mAP0.5 increased by 15% and mAP0.5:0.95 increased by 13.8%, with our model having only a quarter of the parameters of YOLOv7. Compared to YOLOx, mAP0.5 increased by 8.5% and mAP0.5:0.95 increased by 9.1%. Finally, compared to YOLOv5, mAP0.5 increased by 1.8% and

TABLE 4. Ablation results of different models.

BottleneckCSPGC	NWD	P（%）	R（%）	mAP0.5（%）	mAP0.5:0.95（%）
×	×	86.6	78.0	85.3	49.3
√	×	85.8	78.6	86.3	50.1
×	√	86.7	78.2	85.7	48.5
√	√	**87.9**	**80.9**	**87.1**	**53.3**

TABLE 5. Comparison between different networks.

model	P(%)	R(%)	mAP0.5(%)	mAP0.5:0.95(%)	Parm(M)	GFLOPs(G)
YOLOv4	80.9	33.2	57.2	28.6	64	142
YOLOv7	83.0	66.1	72.1	39.5	36.5	103.2
YOLOx	90.8	64.1	78.6	44.2	8.939	26.763
YOLOv5	86.6	78.0	85.3	49.3	7.03	16
ours	87.9	**80.9**	**87.1**	**53.3**	8.237	18

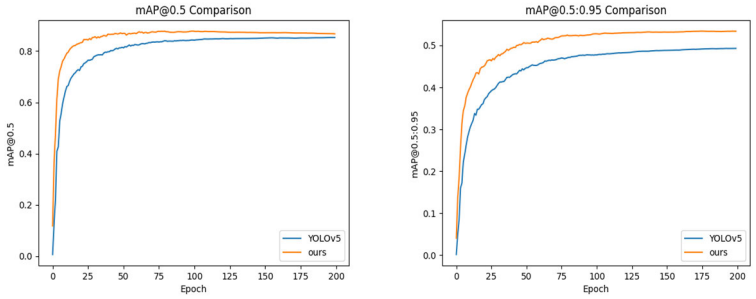

FIGURE 8. Comparison of the accuracy of YOLOv5 and ours.

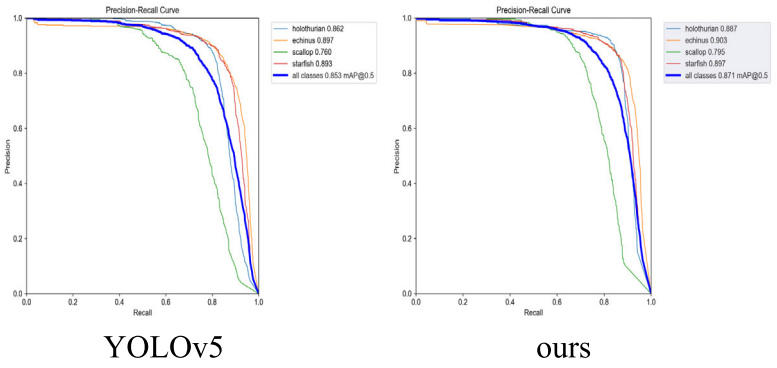

FIGURE 9. mAPs of different classes of YOLOv5 and ours.

mAP0.5:0.95 increased by 4%. A comparison of precision between our model and the second-best YOLOv5 is shown in Figure 8, and Figure 9 compares the accuracy of different categories. These results demonstrate that the proposed model has a comprehensive ability for underwater target detection.

H. DEEPSORT TRACKING STATISTICS

To better verify the accuracy of the tracking statistics, we randomly arranged simulated models of holothurian, echinus, scallop, and starfish in an experimental water tank (Figure 10). We used Dataset 2 for training the detection and tracking weights. The tracking module achieved an accuracy of over 90%. The tracking and statistical results are depicted in Figure 11 (the picture is a frame captured from the video, and the red font in the upper left corner is the number of statistical results), with the detailed statistics presented in Table 6.

Figure 11 is a frame extracted from the video, with real-time tracking statistics displayed in red text in the top left

TABLE 6. Statistical results.

category	True value	Statistical value
holothurian	13	13
echinus	11	11
scallop	40	38
starfish	10	10

FIGURE 10. Experimental water tank.

FIGURE 11. Tracking statistics results.

corner. Each detection frames has a unique ID that remains consistent and does not change as the camera moves. The count is entirely accurate. According to the overall tracking and statistical results presented in Table 6, the count of holothurian, echinus, and starfish in our experimental setup matches the actual numbers exactly. Only the count of scallops is off by two. Overall, the experimental results demonstrate good accuracy and strong tracking and statistical capabilities.

V. CONCLUSION

This paper proposes an improved YOLOv5-DeepSORT-based algorithm for target detection and tracking, designed specifically for marine benthic organism detection and counting in complex underwater environments. To achieve more accurate underwater target detection, we introduce an innovative BottleneckCSPGC module to enhance feature extraction for underwater targets. Additionally, we incorporate the Normalized Wasserstein Distance (NWD) loss function to improve detection performance for small targets. Experiments conducted on Dataset One demonstrate that our improved network outperforms other state-of-the-art detection algorithms. Furthermore, we added a counting functionality to DeepSORT and validated it using Dataset Two. The experimental results indicate that the counting accuracy is high.

We have currently completed the collection and creation of the dataset and have innovated the network architecture for training. On this basis, we integrated DeepSORT and added a counting feature. In the future, we will also consider deploying to mobile devices. However, in complex underwater target detection, image blurring remains a significant challenge affecting detection accuracy. Future research should focus on collecting high-quality underwater datasets and developing new data augmentation algorithms specifically for underwater images to improve image quality. Addressing these challenges is crucial for advancing underwater target detection and the accurate monitoring and counting of marine benthic organisms.

REFERENCES

[1] L. Wenhai, L. Xiao, and C. Meng, "Constructing a taxonomic and zoning system for marine ecology to promote ecosystem-based ocean management," *Bull. Chin. Acad. Sci.*, vol. 39, no. 5, pp. 872–880, 2024.

[2] D. Crespo and M. Ă. Pardal, "Ecological and economic importance of benthic communities," in *Life Below Water*. Cham, Switzerland: Springer, 2020, pp. 1–11.

[3] J. Yu, S.-H. Kong, and Y. Meng, "Underwater vision environment perception methods and technologies," *Robotics*, vol. 44, no. 2, pp. 224–235, 2022.

[4] D. Ma, H. Fang, N. Wang, H. Lu, J. Matthews, and C. Zhang, "Transformer-optimized generation, detection, and tracking network for images with drainage pipeline defects," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 38, no. 15, pp. 2109–2127, Oct. 2023.

[5] Z. Wang, S. Wang, X. Wang, and X. Luo, "Underwater moving object detection using superficial electromagnetic flow velometer array-based artificial lateral line system," *IEEE Sensors J.*, vol. 24, no. 8, pp. 12104–12121, Apr. 2024.

[6] Q. Xiao, Q. Li, and L. Zhao, "Lightweight sea cucumber recognition network using improved YOLOv5," *IEEE Access*, vol. 11, pp. 44787–44797, 2023.

[7] P. Liu, W. Qian, and Y. Wang, "YWnet: A convolutional block attention-based fusion deep learning method for complex underwater small target detection," *Ecol. Informat.*, vol. 79, Mar. 2024, Art. no. 102401.

[8] M. I. H. Azhar, F. H. K. Zaman, N. Md. Tahir, and H. Hashim, "People tracking system using DeepSORT," in *Proc. 10th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Aug. 2020, pp. 137–141.

[9] Q. Xiaofeng, S. Xiangrui, C. Yongchang, and W. Xinyan, "Pedestrian detection and counting method based on YOLOv5+DeepSORT," *Proc. SPIE*, vol. 12080, pp. 177–181, Nov. 2021.

[10] Z. Duan, S. Li, J. Hu, J. Yang, and Z. Wang, "Review of deep learning object detection methods and their main frameworks," *Laser Optoelectron. Prog.*, vol. 57, no. 12, 2020, Art. no. 120005.

[11] T. Wang, J. Huang, H. Zhang, and Q. Sun, "Visual commonsense R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10757–10767.
[12] N. Arora, Y. Kumar, R. Karkra, and M. Kumar, "Automatic vehicle detection system in different environment conditions using fast R-CNN," *Multimedia Tools Appl.*, vol. 81, no. 13, pp. 18715–18735, May 2022.
[13] H. Qin, J. Wang, X. Mao, Z. Zhao, X. Gao, and W. Lu, "An improved faster R-CNN method for landslide detection in remote sensing images," *J. Geovisualization Spatial Anal.*, vol. 8, no. 1, p. 2, Jun. 2024.
[14] H. F. Xu, D. M. Huang, Q. He, Y. L. Du, and X. B. Qin, "Ocean front detection method based on improved mask R-CNN," *J. Image Graph.*, vol. 26, no. 12, pp. 2981–2990, 2021.
[15] A. Kumar and S. Srivastava, "Object detection system based on convolution neural networks using single shot multi-box detector," *Proc. Comput. Sci.*, vol. 171, pp. 2610–2617, Jan. 2020.
[16] Y. Shao, D. Zhang, H. Chu, X. Zhang, and Y. Rao, "A review of YOLO object detection based on deep learning," *J. Electron. Inf. Sci.*, vol. 44, no. 10, pp. 3697–3708, 2022.
[17] M. Zand, A. Etemad, and M. Greenspan, "ObjectBox: From centers to boxes for anchor-free object detection," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2022, pp. 390–406.
[18] Y. Juan and L. Shun, "Detection method of illegal building based on YOLOv5," *J. Comput. Eng. Appl.*, vol. 57, no. 20, pp. 236–244, 2021.
[19] L. Yuan, H. Tang, Y. Chen, R. Gao, and W. Wu, "Improving the detection method of road targets in complex environments with YOLOv5," *J. Comput. Eng. Appl.*, vol. 59, no. 16, pp. 212–222, 2023.
[20] Y. Zhang, H. Z. Lu, L. P. Zhang, and M. Hu, "A review of visual multi-target tracking algorithms based on deep learning," *J. Comput. Eng. Appl.*, vol. 57, no. 13, pp. 55–66, 2021.
[21] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.
[22] F. U. Jinyi, Z. Zijia, S. U. N. Wei, and Z. O. U. Kaixin, "Improved YOLOv8 small target detection algorithm in aerial images," *J. Comput. Eng. Appl.*, vol. 60, no. 6, p. 100, 2024.
[23] J. Wang, C. Xu, W. Yang, and L. Yu, "A normalized Gaussian Wasserstein distance for tiny object detection," 2021, *arXiv:2110.13389*.
[24] S. Du, B. Zhang, P. Zhang, and P. Xiang, "An improved bounding box regression loss function based on CIOU loss for multi-scale object detection," in *Proc. IEEE 2nd Int. Conf. Pattern Recognit. Mach. Learn. (PRML)*, Jul. 2021, pp. 92–98.
[25] L. Yang, G. Yuan, H. Zhou, H. Liu, J. Chen, and H. Wu, "RS-YOLOX: A high-precision detector for object detection in satellite remote sensing images," *Appl. Sci.*, vol. 12, no. 17, p. 8707, Aug. 2022.
[26] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X. S. Hua, "α-IoU: A family of power intersection over union losses for bounding box regression," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20230–20242.
[27] Z. Gevorgyan, "SIoU loss: More powerful learning for bounding box regression," 2022, *arXiv:2205.12740*.
[28] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[29] X. Xu and X. Li, "Research on surface defect detection algorithm of pipeline weld based on YOLOv7," *Sci. Rep.*, vol. 14, no. 1, p. 1881, Jan. 2024.
[30] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

JIAN LIU is currently pursuing the master's degree with the Department of Electronics, Electrical and Control, Qilu University of Technology. His research interests include underwater object recognition and tracking.

QIAN LI has been engaged in the research and development of robotics with the Institute of Automation, Shandong Academy of Sciences, since April 2007. He is currently the Director of the Underwater Robot and Intelligent System Research Team and a Master's Tutor with the Qilu University of Technology. He is also a Professor with the Institute of Automation, Shandong Academy of Sciences.

SHANTAO SONG is currently pursuing the master's degree with the Department of Electronics, Electrical and Control, Qilu University of Technology. His research interests include image stitching and correction.

KALIYEVA KULYASH received the Ph.D. degree in physics and mathematics.

She is currently a Professor with the Faculty of Mechanics-Mathematics, Al-Farabi Kazakh National University.