IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 January 2024, accepted 16 February 2024, date of publication 22 February 2024, date of current version 1 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368878

RESEARCH ARTICLE

An Efficient YOLO Network With CSPCBAM, Ghost, and Cluster-NMS for Underwater Target Detection

ZHENG ZHANG, QINGSHAN TONG, AND XIAOFEI HUANG
School of Mathematics and Statistics, Changsha University of Science and Technology, Changsha 410114, China

Corresponding author: Qingshan Tong (tongqs@csust.edu.cn)

This work was supported by the School of Mathematics and Statistics, Changsha University of Science and Technology.

ABSTRACT In recent years, owing to the rapid advancements in deep learning, advanced object detection methods, such as You Only Look Once (YOLO) and Efficient Detector (EfficientDet), have been frequently used to detect underwater organisms. However, due to the complexity of underwater scenarios and deployment limitations, these models often encounter various challenges, such as blurred targets, occlusions, and high model computing costs. On this basis, we propose a YOLO network (CGC-YOLO) based on Cross-Stage Partial Convolutional Block Attention Module (CSPCBAM), Ghost module, and cluster non-maximum suppression (Cluster-NMS). Firstly, CSPCBAM enhances the model's ability to extract intricate features by amplifying pertinent feature information across both channel and spatial dimensions. This augmentation contributes to an improved detection performance of the model, especially when dealing with fuzzy targets. Secondly, the Ghost module is employed to optimize the model's efficiency by decreasing its parameters and reducing the computational load in terms of floating-point operations per second (FLOPs). Finally, by introducing Cluster-NMS and Score Penalty Mechanism (SPM) to reweight the confidence of bounding boxes, the model can retain the real object with occlusion. The experimental results show that on the Underwater Robot Picking Competition 2020 (URPC 2020) and brackish water dataset, the mAP@0.5 of our proposed CGC-YOLO reaches 87.2% and 98.6% respectively, which is at least 1 percentage point higher than all other models. The CGC-YOLO has 14.8 FLOPs and speeds of 7.1ms and 6.3ms, respectively, which is also better than all other models. Ablation experiments and qualitative analysis show that CGC-YOLO can deal with fuzzy and obscured objects well, with lower computational cost and faster inference speed.

INDEX TERMS Underwater detection, attention module, non-maximum suppression, lightweight model.

I. INTRODUCTION

The rising demand for marine resources and advancements in ocean detection technology have resulted in increased usage of intelligent underwater robots for identifying and monitoring marine organisms, such as seaurchins, seacucumbers, starfish, and scallops. This development is significantly important for the sustainable exploitation of marine resources [1] and military applications [2] in the marine domain.

Despite the increasing utilization of intelligent underwater robots for marine organism identification and monitoring, the task of underwater object detection still encounters several challenges. First, low underwater visibility can cause captured objects to appear blurred, compromising the accuracy and reliability of detection algorithms [3]. Second, underwater creatures are frequently densely packed, resulting in occlusion between targets and causing the detection algorithm to miss the real targets [4]. Finally, current target detection algorithms are computationally intensive with complex structures, resulting in the difficulty of achieving real-time detection by underwater robots. Thus, more lightweight models must be proposed [5].

Existing methods have adopted various improvement strategies based on the detection model framework to address

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

the challenges of blurred objects, severe occlusion, and large computational expenses in underwater scenes. A feasible method involves adding an attention module [6] to the network, such as Bi-Level routing attention [7], Convolution Block Attention Module (CBAM) [8], and Graph Attention Module (GAM) [9], to improve its expressive ability so that the network can concentrate on relevant features. However, this method is prone to overfitting, wherein the model may excessively focus on unimportant object edge features and ignore other critical background features.

When traditional convolution is used for convolution operation, the spatial relationship between input channels and the weight between output channels must be studied, resulting in unnecessary redundant information and increased calculation cost. Some studies are devoted to designing lightweight neural networks instead of traditional convolutional neural networks to extract features. These methods, such as MobileNet [10], [11], [12] and ShuffleNet [13], [14], [15], construct small and efficient convolutional filters through depth-wise separable convolutions and channel shuffling operations. However, the remaining convolutional layers still consume a significant amount of memory and computation. Accordingly, the Ghost module [16] has been proposed to address this issue. It reduces the number of convolution filters used to generate intrinsic feature maps. These feature maps are then transformed into more feature maps through simple linear operations. This approach effectively minimizes the computational complexity and parameters compared with the original convolution layers.

The majority of the target detection models based on anchors adopt the Non-Maximum Suppression (NMS) algorithm to eliminate redundant prediction boxes. However, in the case of highly overlapping objects caused by occlusion, the traditional NMS algorithm selects only the bounding box with the highest score and ignores the other overlapping bounding boxes, resulting in missed detections. Numerous approaches have been suggested to address the problems of the NMS algorithm, such as Soft-NMS [17], Fast-NMS [18], and Cluster-NMS [19]. Although Soft-NMS significantly improves the detection performance of occluded objects, its computational complexity increases due to the introduction of the Score Penalty Mechanism (SPM), which reduces the detection efficiency. Fast-NMS is highly efficient in suppressing through the Intersection over Union (IoU) matrix, but it easily causes over-suppression. Cluster-NMS divides bounding boxes into different clusters and performs row transformation for each cluster. The iteration only occurs on the cluster with the most bounding boxes, minimizing the number of iterations and computations.

This work proposes a You Only Look Once (YOLO) network (CGC-YOLO) based on Cross-Stage Partial Convolution Block Attention Module (CSPCBAM), the Ghost module, and Cluster Non-Maximum Suppression (Cluster-NMS) to address certain problems, such as ambiguity, occlusion, and deployment difficulties in underwater detection. CSPCBAM seamlessly integrates the benefits of the attention mechanism and Cross-Stage Partial (CSP) structure. This synergy enhances the model's perceptual acuity for targets in underwater scenes, facilitating selective attention to crucial and distinct features. Furthermore, it bolsters the detection model's robustness across diverse target scales, particularly small targets. Consequently, this integration leads to a notable enhancement in the overall accuracy of the model. The Ghost module operates by employing a limited number of convolution kernels to generate intrinsic feature maps and enhance the nonlinear representation capability of the network while minimizing the number of parameters through depth-wise separable convolution and splicing operations. Cluster-NMS introduces the SPM to update the scores of highly overlapping detection boxes, thereby preserving the real object boxes that overlap each other due to occlusion. The main contributions of our work are as follows:

- A CGC-YOLO model is proposed, which introduces CSPCBAM, Ghost module, and Cluster-NMS based on YOLOv5. This model primarily addresses the problems of ambiguity, occlusion, and high computational cost in underwater target detection.
- CSPCBAM is designed to extract the channel and space feature information of fuzzy objects. The ghost module is applied to reduce parameters and improve the characteristic expression ability of the model through deep separable convolution and stitching operations. By integrating SPM, Cluster-NMS can effectively retain obscured targets while ensuring predictive efficiency.
- Our CGC-YOLO achieves the best performance on the Underwater Robot Picking Competition (URPC) 2020 and Brackish datasets. In addition, we verify the effectiveness of CSPCBAM, Ghost module, and Cluster-NMS in solving the problems of fuzzy target, occlusion, and large model parameters through ablation study and qualitative analysis.

The rest of the paper is arranged as follows: Section II briefly reviews the relevant work and research. Section III introduces the overall structure of CGC-YOLO and the details of the improved module. Section IV mainly conducts experiments and analyzes the results. Section V is the discussion and conclusion.

II. RELATED WORK

Herein, we review existing methods for improving the target detection model, which is mainly categorized into three: modules based on attention mechanism, lightweight object detection methods, and improved NMS algorithms.

A. MODULES BASED ON ATTENTION MECHANISM

The attention mechanism is a method utilized to dynamically allocate attention weight to improve model performance. This method is widely used in image processing [20], speech recognition [21], and natural language processing [22] because of its advantages of few parameters, strong

adaptability, and high speed. Various approaches have embedded the attention module into detection networks to address the issue of underwater image blurring. Jia et al. [23] proposed the EfficientDet-Revised (EDR) to tackle the issue of poor differentiation between marine organisms and their environment in underwater settings marked by ambiguity. By designing a Channel Shuffle module to reconstruct the MBConvBlock [24], the exchange of information between feature layer channels was achieved, thereby enhancing the feature extraction capacity of the network for differentiating targets in the underwater environment. Zhao et al. [25] fused the selective kernel network with YOLOv5's Backbone, weighting feature information of different scale channels to enhance the recognition ability of fuzzy fish. Additionally, Wen et al. [26] introduced an enhanced version of the YOLOv5s network, denoted as YOLOv5s-CA. This model incorporates the Coordinate Attention (CA) [27] module and the Squeeze-and-Excitation (SE) [28] module to boost performance in shallow feature extraction and augment attention within the model. Consequently, the accuracy of underwater target detection is improved.

B. LIGHTWEIGHT OBJECT DETECTION METHODS

A series of lightweight object detection methods have been proposed in recent years to meet the needs of deploying neural networks on embedded devices. Yu et al. [29] introduced YOLOv4-Tinier, a rapid and accurate underwater biological detection method, built upon enhancements to the YOLOv4-Tiny [30] algorithm. The reduction of computational complexity is achieved through the integration of Ghost modules into the backbone network, CSPDarknet53-Tiny. This results in an effective decrease in model size while preserving a high level of detection accuracy. Separately, Zhang et al. [31] introduced CSL-YOLO, a novel lightweight object detector that centers on an innovative Cross-Stage Lightweight (CSL) module design. This unique architecture efficiently merges deep convolutional layers with pointwise convolutions to generate feature maps with significantly lower computational requirements. Sun et al. [32] introduced a novel underwater target detection model, amalgamating MobileViT [33] and YOLOX [34]. MobileViT serves as the backbone network to streamline parameter count, while a dual-coordinate attention (DCA) mechanism is designed to amplify small target detection. Experimental results validate the model's suitability for lightweight algorithms on underwater unmanned platforms, ensuring both efficiency and high detection accuracy. Unfortunately, these methods do not fully utilize the correlation within feature maps.

C. IMPROVED NMS ALGORITHMS

The NMS algorithms can only preserve sparse real targets in occluded scenes but cannot preserve overlapping real targets. Accordingly, some researchers have dedicated their efforts to improving NMS algorithms. Li et al. [35] used Soft-NMS to avoid overlapping regions of the same category of defects in the fabric dataset, improving the detection accuracy of fabric defects. Wang et al. [36] realized the detection of sweet potato leaves in natural scenes through DIoU-NMS [37], which considers not only the overlapping regions but also the distance between the center points of two boxes, thus improving the error suppression of the IoU measure in the case of occlusion. Chen et al. [38] used the K-Means clustering method to determine the aspect ratio range, which effectively improved the recall rate of object detection. However, these improved methods increase the time complexity, which is not conducive to the real-time detection of underwater robots. By contrast, Cluster-NMS [39] can be parallelized on the implicit cluster of detected boxes because it has fewer iterations and higher efficiency than other methods.

III. METHODS

A. OVERVIEW OF CGC-YOLO

We propose CGC-YOLO to enhance the accuracy and speed of underwater object detection and minimize computational expense. The network structure of CGC-YOLO is shown in the Figure. 1, where the parameters in brackets represent the number of input channels, output channels, convolution kernel size, and stride. We introduce the CSPCBAM modules (shown in light yellow) and the Ghost modules (shown in light orange) into the YOLOv5s architecture to achieve better performance. CSPCBAM aims to enhance the ability of a network to extract features across channels and spaces. Cluster-NMS aims to prevent the true positives with partial overlap from being suppressed during post-processing. The ghost module aims to minimize the model's calculation and maximize the inference speed.

B. CSPCBAM

The CBAM attention weights are harnessed in this study to augment the network's focus on the distinct characteristics of the target. Commonly, prevailing methodologies involve the direct integration of CBAM modules between convolutional modules within the backbone network. Nevertheless, the model's attention to both the extraction of target features and the diverse visibility characteristics of the targets should be concurrently addressed. As such, this paper introduces the CBAM module within the CSP structure of the convolution module, resulting in the CBAMBottleneck moudel. The objective is to enhance the detection model's resilience across targets of varying scales and concurrently amplify attention to the lucid features of targets, thereby improving the model's accuracy in discerning fuzzy targets. As shown in Figure. 2. Given an intermediate feature map $I \in R^{C\times H\times W}$ of height H, width W, and channel C as an input, the CBAMBottleneck module divides the input feature map I into two branches. The first branch passes through a Convolution-Batch Normalization-SiLU (CBS) module M_{CBS} with a kernel size of 1×1 and step length of one to generate a feature map $F \in R^{C\times H\times W}$ and infers a 1D channel

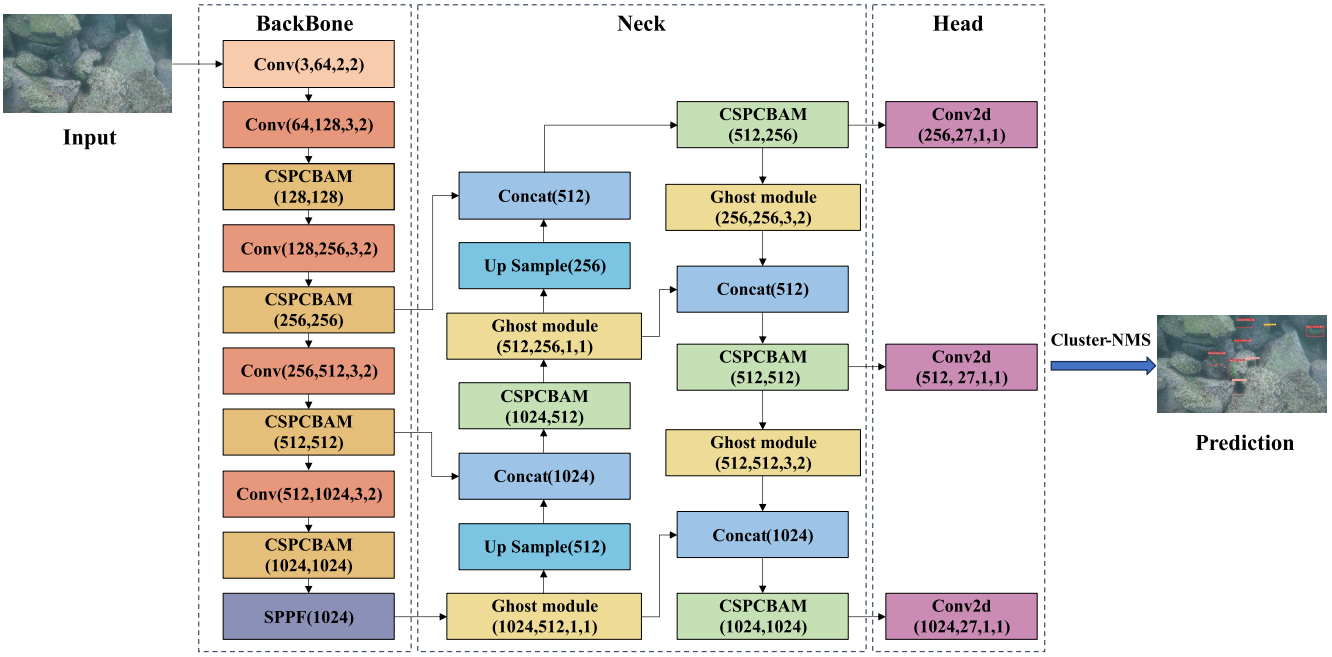


FIGURE 1. The proposed CGC-YOLO network. The digits in brackets for each module denote input channel, output channel, convolution kernel size, and stride. CSPCBAM is used to extract target features, Ghost module is used for feature fusion, and Cluster-NMS is used as a post-processing algorithm to remove redundant bounding boxes.

attention map $M_c \in R^{C\times1\times1}$ and a 2D spatial attention map $M_s \in R^{1\times H\times W}$. The calculation of the CBS module can be expressed as follows:

$$\begin{aligned} F &= CBS(I) = SiLU(BN(Conv(I))),\\ Conv(x) &= f^{3\times3}(x),\\ BN(x) &= \gamma\cdot(\frac{x-\mu(x)}{\sqrt{\sigma(x)^2+\epsilon}})+\beta,\\ SiLU(x) &= x * sigmoid(x) = \frac{x}{1+e^{-x}}, \end{aligned} \tag{1}$$

where x is the vector formed by the samples of the current batch. $f^{3\times3}$ is a convolution layer with kernel size of 3, step size of 1, and padding of 1. μ and σ are the mean and standard deviation of the vector. ϵ is introduced with a value of 10^{-5} to prevent division by zero, as suggested in the referenced paper [40]. γ and β are the scale and shift parameters to be learned. The output $F'' \in R^{C\times H\times W}$ refined by channel and spatial attention is processed with the input feature map I in the second branch after passing through another CBS module. The skip connection between the two branches results in the final output feature map $F_O \in R^{C\times H\times W}$. This design enhances the feature representation ability of the network while minimizing computational load. Moreover, this connection helps in improving the accuracy of detection on blurry underwater images. The calculation of the CBAM module can be expressed as follows:

$$\begin{aligned} F' &= M_c(F)\otimes F,\\ F'' &= M_s(F')\otimes F',\\ F_O &= I\oplus CBS(F''), \end{aligned} \tag{2}$$

where $\otimes$ represents the element-by-element multiplication, and $\oplus$ represents the element-by-element addition. In addition, the channel and spatial attention values are broadcasted along with their respective dimensions. F' is the result achieved by element-wise multiplication of $Mc(F)$ with F. Meanwhile, F'' is the result acquired by element-wise multiplication of $Ms(F')$ with F' and F_O is the final result obtained by element-wise addition of F'' with I. Channel attention enhances the importance of feature channels by learning the weights associated with each channel and multiplying them with the corresponding channels. Feature map F first undergoes average and maximum pooling operations to obtain two different feature maps F_{avg} and F_{max} that contain spatial context information. To minimize computational overhead, this work uses a shared network consisting of two 1×1 convolutional layers to process these two feature maps and generates the channel attention map M_c through element-wise addition and sigmoid activation function. The calculation process of the channel attention is as follows:

$$\begin{aligned} M_c(F) &= \sigma(f^{1\times1}(AvgPool(F)) + f^{1\times1}(MaxPool(F)))\\ &= \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))), \end{aligned} \tag{3}$$

where F^c_{avg} and F^c_{max} represent the feature maps generated after average and maximum pooling, respectively; σ represents the sigmoid activation function; $f^{1\times1}$ denotes the convolution layer with a kernel size of 1×1; $W_0 \in R^{C/r\times C}$ and $W_1 \in R^{C\times C/r}$ are weights of each channel; and r is the reduction ratio. Both inputs share the same weights. Meanwhile, the ReLU activation function follows

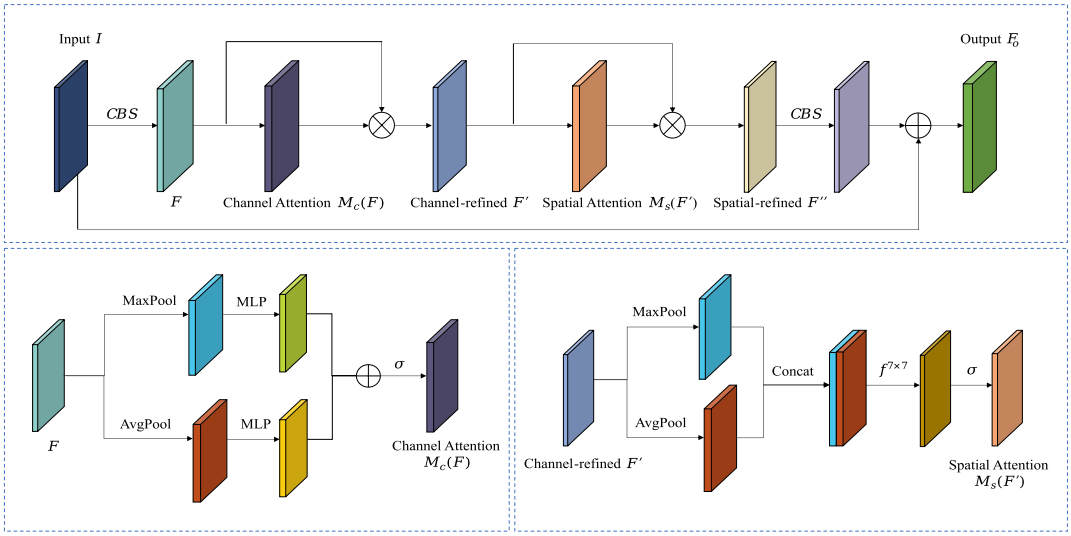


FIGURE 2. The architecture of the proposed CBAMBottleneck, where ⊕ signifies element-wise addition, ⊗ represents element-wise multiplication, and σ represents sigmoid activation function. The input feature map I is operated by CBS to generate feature map F, which is refined by channel and spatial attention to generate F''. The feature map generated by F'' after passing through CBS is residually connected with I to obtain the output feature map F_o.

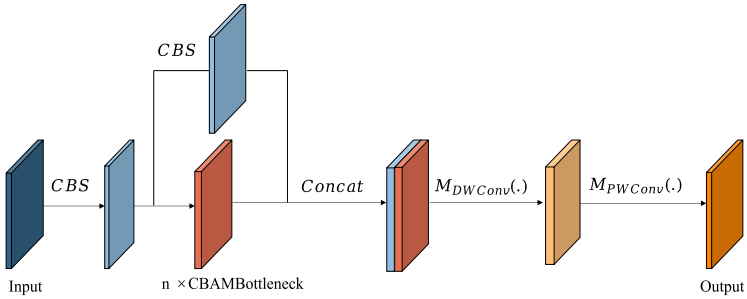


FIGURE 3. The structure of proposed CSPCBAM. It divides the feature map into two parts, one part is fused with the other through cross-stage connection, and then depth-wise convolution and point-wise convolution are performed. Finally, a multi-scale feature representation is formed. This design preserves low-level details while enhancing high-level semantic information and reducing the number of parameters.

W_0. Spatial attention enhances the model's attention to target location information by selectively highlighting information-rich regions of the feature map and encoding the spatial relationships of features. First, maximum and average pooling operations are performed along the channel of the feature map F', resulting in two feature maps, $F_{avg} \in R^{1\times H\times W}$ and $F_{max} \in R^{1\times H\times W}$. After concatenation, the combined feature maps undergo a convolutional layer with a filter size of 7 × 7. Subsequently, a sigmoid activation function is applied to generate the spatial attention map M_s. The calculation of the spatial attention is as follows:

$$\begin{aligned} M_s(F) &= \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) \\ &= \sigma(f^{7\times7}([F^s_{avg}; F^s_{max}])), \end{aligned} \tag{4}$$

where F^s_{avg} corresponds to the feature map obtained through average pooling. F^s_{max} corresponds to the feature map generated through maximum pooling.

As shown in Figure 3, the CSPCBAM module includes CBAMBottleneck, skip connection, splicing operation, and depth separable convolution. Similar to the CSP module, the CSPCBAM module uses CBAMBottleneck instead of the original bottleneck structure, and replaces the traditional convolution with deep separable convolution, thus reducing the parameters and computation. The calculation process of the whole depth separable convolution is as follows:

$$\begin{aligned} F_{DWConv} &= M_{DWConv}(F) = f^{3\times3\times1}(F_{concat}), \\ F_{PWConv} &= M_{PWConv}(F) = f^{1\times1\times c}(F_{DWConv}), \end{aligned} \tag{5}$$

where F_{DWConv} represents the feature map generated by the depth-wise convolution of M_{DWConv}. F_{PWConv} represents the feature map generated by point-wise convolution of M_{PWConv}. $f^{3\times3\times1}$ represents a convolution layer with a kernel size of 3 and channels of 1. $f^{1\times1\times c}$ represents a convolution

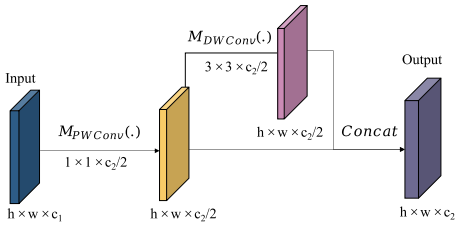


FIGURE 4. The structure of the Ghost module. *h*, *w*, *c*1, and *c*2 denote the height, width, input channel, and output channel. The input feature map is divided into two parts after $1 \times 1 \times c_2$ point-wise convolution. One part after $3 \times 3 \times c_2$ depth-wise convolution is combined with the other part along the channel to generate the final output feature map.

layer with a kernel size of 1 and channels of c. The incorporation of cross-stage connections in the network's mid-layer facilitates direct communication between feature maps from various stages. This communication is instrumental in synthesizing information across different stages, thereby amplifying the network's memory and contextual information perception capabilities. The attention weighting mechanism employed by CBAM effectively assigns weights to key channels and spatial positions, directing the model to prioritize clear features of the target while mitigating attention towards ambiguous features. The objective behind designing CSPCBAM in this paper is to synergistically leverage the benefits of both cross-stage connections and CBAM, aiming to enhance the accuracy of the detection model, especially in the identification of fuzzy targets.

C. GHOST MODULE

The advantage of the Ghost module over other lightweight modules is that it enhances the nonlinear expressiveness of the network by adding depth-wise separable convolutional layers and concat operations. Accordingly, this work uses the Ghost to substitute the CBS of the neck of YOLOv5s. The CBS module in the entire network is not replaced because if the Ghost is used in all stages of the model, then the number of network layers will become deeper. These deep structures can lead to increased data flow resistance and substantially longer inference time. However, by the time the feature maps reach the neck, they have already become slim (with maximum channels and minimum width and height). As a result, the feature maps produced by the Ghost module contain fewer redundant details, eliminating the need for compression [41]. Hence, using the Ghost only on the neck is a better choice. Figure 4 depicts the structure of the Ghost module. The input feature map is partitioned into two branches along the channels after passing through the point-wise convolution layer. One branch produces a feature map through depth-wise convolution, and it is then concatenated with the other branch along the channels to fuse local feature information, thereby achieving the goal of reducing parameters and computational complexity. The calculation process of the whole module is as follows:

$$\begin{aligned} F'_O &= [M_{PWConv}(I); M_{DWConv}(M_{PWConv}(I))] \\ &= [F'_{PWConv}; F'_{DWConv}], \end{aligned} \tag{6}$$

Algorithm 1 Cluster-NMS Pseudo Algorithm

Input: N bounding boxes $B = [B_1, B_2, \ldots, B_N]^T$, suppression matrix $b = \{b_i\}_{1\times N}, b_i \in \{0, 1\}$ represents decision outcome, where 1 represents reservation and 0 represents suppression, and classification scores $S = \{s_1, s_2, \ldots, s_N\}$ of the boxes, where $s_1 \geq s_2 \geq \cdots \geq s_N$.

Output: the classification scores S of the bounding boxes, the bounding boxes B.

Initialize $T = N$, $t = 1$, and $b^0 = 1$, calculate IoU matrix $X = \{x_{ij}\}_{N\times N}$ with $x_{ij} = IoU(B_i, B_j)$, $i, j = 1, \ldots, n$.
$X = triu(X)$ ▷ Upper triangular matrix with $x_{ii} = 0, \forall i$
while $t \leq T$ **do**
  $A^t = diag(b^{t-1})$
  $C^t = A^t \times X$
  $g \leftarrow \max_j C^t$ ▷ Find maximum for each column j
  $b^t \leftarrow find(g) = \begin{cases} b_j = 1, if\ g_j < \varepsilon \\ b_j = 0, if\ g_j \geq \varepsilon \end{cases}$
  if $b^t == b^{t-1}$ **then**
    break
  end if
  $t = t + 1$
end while
$C^t = \begin{cases} 0, if\ C^t_{ij} < \varepsilon \\ C^t_{ij}, if\ C^t_{ij} > \varepsilon \end{cases}$
$s_j = s_j \prod_i e^{-\frac{C^t_{ij}}{\sigma}}$
return S, B

where $I \in R^{h\times w\times c_1}$ denotes the input feature map with height h, width w, and channels of c_1, $F'_O \in R^{h\times w\times c_2}$ denotes the output feature map with channels of c_2. $F'_{PWConv} = f^{1\times1\times c_2/2}(I)$, $F'_{DWConv} = f^{3\times3\times c_2/2}(F'_{PWConv})$. This structure diversifies the network to improve accuracy while minimizing network computation and parameters, thereby enhancing model inference speed and reducing latency.

D. CLUSTER-NMS

Cluster-NMS accelerates non-maximum suppression by parallelizing IoU calculations. As illustrated in Algorithm 1, the bounding boxes are sorted in descending order based on their scores $S = s_1, s_2, \ldots, s_N$, where $s_1 \geq s_2 \geq \cdots \geq s_N$, resulting in the reordered set $B = [B_1, B_2, \ldots, B_N]^T$. The IoU matrix $X = \{x_{ij}\}_{N\times N}$ is first computed, where $x_{ij} = IoU(B_i, B_j) = x_{ji}$. Since X is a symmetric matrix, we only need the upper triangular matrix of X:

$$X = triu(X) = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ 0 & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_{nn} \end{pmatrix},$$
$$x_{ij} = IoU(B_i, B_j), \tag{7}$$

Initiate a binary vector, denoted as $b = b_{i1\times N}$, representing the outcome of non-maximum suppression, where $b_i \in 0, 1$. Set $b^0 = 1$ as the initial state. Specifically, at iteration t, denote the bounding box decision result as b^{t-1}. Construct a diagonal matrix A^t from the elements of the vector b^t in their original order. Subsequently, perform a left multiplication of the IoU matrix X by the diagonal matrix A^t, yielding the suppression matrix C.

$$\begin{aligned} A^t &= diag(b^{t-1}), \\ C^t &= A^t \times X, \end{aligned} \tag{8}$$

The binary vector g is derived by selecting the largest element from each column in matrix C. Subsequently, g is binarized using the threshold ε which is set to 0.5 [42]. Specifically, elements in g that are less than ε are set to 0, while those greater than ε are set to 1, yielding the updated vector b^t. The process is iterated by left-multiplying the IoU matrix X continuously using the diagonal matrix A until convergence, i.e., $b^t = b^{t-1}$, resulting in the final suppression matrix C. In addressing the limitations of traditional NMS, which merely removes adjacent bounding boxes surpassing the threshold ε and is incapable of resolving occlusion issues between neighboring objects, this paper incorporates the SPM of Soft-NMS [43] into Cluster-NMS. With the introduction of penalty terms, the objective is to penalize bounding boxes with scores exceeding the NMS threshold, while those below the threshold remain unaffected. The entire process of scoring penalty for bounding boxes is as follows:

$$C^t = \begin{cases} 0, \text{if } C_{ij}^t < \varepsilon \\ C_{ij}^t, \text{if } C_{ij}^t > \varepsilon, \end{cases} \tag{9}$$

$$s_j' = s_j \prod_i e^{-\frac{C_{ij}^{t\,2}}{\sigma}}, \tag{10}$$

where s_j' is the weighted score of the detection box. σ is the penalty factor, which is set to 0.2 [19]. Given that C^t is an upper triangular matrix, the penalty function will only impose penalties on the detection boxes whose scores are above the IoU threshold ε. After the score penalty is applied, a category score S corresponding to bounding box B is obtained, and bounding boxes with scores greater than the confidence threshold are extracted as the final output prediction box.

In summary, Cluster-NMS demonstrates an enhanced capacity to elevate detection accuracy for occluded targets, all while preserving computational efficiency. This is achieved through the seamless integration of a score penalty function and parallelized calculations. The synergistic application of these techniques not only refines the precision of object detection but also addresses the challenge of occlusion, thereby contributing to the overall efficacy of the model.

IV. EXPERIMENTS

In this section, we evaluate the performance of our CGC-YOLO model on the URPC 2020 [44] and Brackish [45] datasets. We also performed ablation studies to verify the efficiency of CSPCBAM, Cluster-NMS, and Ghost module. Moreover, we apply the Gradient-weighted Class Activation Mapping (Grad-CAM) [46] and results visualization for qualitative analysis to verify whether CGC-YOLO can better handle blurred and occluded targets than YOLOv5. We select the mean average precision (mAP), parameters, floating-point operations (FLOPs), and Speed as evaluation metrics.

A. DATASETS AND METRICS

The URPC 2020 dataset is supplied by China URPC for target recognition, which contains four seafood species, including seaurchin, seacucumber, scallop, and starfish. This dataset contains 5543 images, which are partitioned into training, validation, and test sets at a ratio of 7:2:1 [23].

The Brackish dataset is the first publicly available European underwater image dataset that contains 11,205 images annotated with bounding boxes for fish, shrimp, crab, and other organisms. We conducted a random split of the dataset into training, validation, and test sets, maintaining a ratio of 8:1:1 [47], consistent with our approach for the previous dataset.

In this work, the metrics used for evaluation are mAP@0.5, mAP@0.5:0.95, Parameters, FLOPs, and Speed. Parameters represent the parameter quantity of the network, which is important for the deployment of underwater robots. mAP consists of two evaluation metrics, namely, precision and recall. Precision calculates the fraction of accurate predictions among the total predictions. Meanwhile, recall calculates the fraction of correctly positioned and recognized objects in the ground truth. The calculation formulas of precision and recall are shown as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{11}$$

$$Recall = \frac{TP}{TP + FN}, \tag{12}$$

where TP, FP, and FN represent the number of true positives, false positives, and false negatives, respectively, in a prediction. The calculation of AP and mAP for measuring the detector performance are shown as follows:

$$AP = \int_0^1 Precision(Recall) dRecall, \tag{13}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{14}$$

where n denotes the number of classes. mAP@0.5 is the average AP of total classes when the IoU threshold is 0.5. The mAP@0.5:0.95 signifies the average mAP at various IoU thresholds, ranging from 0.5 to 0.95 with increments of 0.05. When employing a low IoU threshold, such as 0.5, the model tends to accept partially overlapping prediction boxes. While this tendency can improve the recall rate, it may also result in an increase in false positives (FP), subsequently diminishing the precision. On the other hand, using a higher IoU threshold, like 0.95, makes the model more

stringent in handling partially overlapping targets, thereby enhancing precision. However, this stringency may lead to the overlooking of some true positives (TP), thereby reducing the recall.

To conduct a comprehensive assessment of model performance, we incorporate the evaluation of mAP@0.5:0.95. This metric considers a spectrum of IoU thresholds, offering a well-rounded evaluation that accounts for both recall and precision aspects.

B. EXPERIMENTAL ENVIRONMENT AND TRAINING

To ensure the model's comparability, all the models are trained under the Windows 10 operating system and Pytorch1.10 framework. The software environments are CUDA11.6, CUDNN8.3, and Python 3.8. The CPU is Intel Core i7-12700F, and the GPU is Nvidia GeForce RTX 3060.

This work adopts the same loss function as YOLOv5, which combines the complete intersection over union (CIoU) loss [37] and the binary cross-entropy (BCE) loss. The CIoU loss is employed to calculate the bounding box loss $\mathcal{L}_{\text{box}}$:

$$IoU(\beta, \beta^{gt}) = \frac{|\beta \cap \beta^{gt}|}{|\beta \cup \beta^{gt}|}, \tag{15}$$

$$\mathcal{L}_{\text{box}} = 1 - IoU(\beta, \beta^{gt}) + \frac{\rho^2(\beta, \beta^{gt})}{c^2} + \alpha v, \tag{16}$$

where β represents the prediction box, β^{gt} represents the actual box, ρ denotes the Euclidean distance between the two center points of β and β^{gt}, and c signifies the diagonal distance between the minimal enclosing area of β and β^{gt}. $\alpha = \frac{v}{1-IoU(\beta,\beta^{gt})+v}$, $v = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2$ is used to measure the consistency of height and width of β and β^{gt}. while the BCE loss is used to calculate confidence loss $\mathcal{L}_{\text{obj}}$ and classification loss $\mathcal{L}_{\text{cls}}$:

$$\mathcal{L}_{\text{obj}} = -p_{gt} \log p - (1 - p_{gt}) \log(1 - p), \tag{17}$$

$$\mathcal{L}_{\text{cls}} = -q_{gt} \log q - (1 - q_{gt}) \log(1 - q), \tag{18}$$

where p is the probability that the model predicts that the input sample is positive, p_{gt} is the label corresponding to the input sample that the positive sample is denoted as 1 and the negative sample denoted as 0, and q is the probability that the model predicts that the input sample is the current category. q_{gt} is the category corresponding to the input sample (current category is 1, not current category is 0). The entire loss function $\mathcal{L}$ is of a sample shown as follows:

$$\mathcal{L} = \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{obj}} + \mathcal{L}_{\text{cls}}, \tag{19}$$

The final loss is obtained by summing up the losses of all training samples. The epoch is set to 100 and the images in the training set are uniformly resized to 640 × 640×3. The gradient accumulation strategy is adopted to increase the batch size to 8 under the condition of limited memory.

C. ABLATION STUDIES

In this work, we conduct ablation studies with the following research objectives: (a) Whether the CSPCBAM module can enhance the model's capability to extract intricate features, consequently leading to an overall performance improvement. (b) Whether the Ghost module can effectively reduce model parameters and computation while enhancing feature expression. (c) Whether Cluster-NMS can enhance the model's detection performance for occluded targets while maintaining detection speed. To achieve these research goals, we designed eight states through four components within the CGC-YOLO: (1)Only YOLOv5 (2)YOLOv5 with CSPCBAM (3)YOLOv5 with Ghost (4)YOLOv5 with Cluster-NMS (5)YOLOv5 with CSPCBAM and Ghost (6)YOLOv5 with CSPCBAM and Cluster-NMS (7)YOLOv5 with Ghost and Cluster-NMS (8)YOLOv5 with CSPCBAM, Ghost, and Cluster-NMS. The results of these comprehensive experiments are presented in Tables 1, 2, and 3.

1) VALIDATION OF CSPCBAM

In this work, we assess the performance of CGC under various states on the URPC 2020 and Brackish test sets, as detailed in Tables 1, 2 and 3. For the URPC 2020 dataset (as shown in Table 1), the mAP of state (2) is 86.2% and 50.9%, surpassing state (1) by 0.6% and 1.2%. State (5) achieves an mAP of 86.4% and 51.4%, exceeding state (3) by 0.2% and 0.6%. State (6) reaches an mAP of 86.3% and 54.7%, outperforming state (4) by 0.6% and 0.9%. Finally, state (8) records an mAP of 86.8% and 55.1%, outpacing state (7) by 0.2% and 0.7%. Likewise, for the Brackish dataset (as seen in Table 2), state (2) achieves an mAP of 96.9% and 72.7%, surpassing state (1) by 1.1% and 1.3%. State (5) attains an mAP of 96.9% and 74.2%, outperforming state (3) by 0.6% and 1.7%. State (6) reaches an mAP of 97.8% and 74.6%, exceeding state (4) by 1.9% and 0.8%. Finally, state (8) records an mAP of 98.6% and 75.2%, which is 0.9% and 0.8% higher than state (7). These results demonstrate that the integration of the CSPCBAM module has led to improvements in accuracy, model complexity, and inference speed. This improvement can be ascribed to the synergistic impact of the attention mechanism and the CSP structure within CSPCBAM. These components enhance the model's capacity to discern targets in underwater scenes, enabling selective focus on pivotal features. Moreover, they augment the detection model's resilience to targets of diverse scales, particularly small ones, thereby further elevating the overall model accuracy.

2) VALIDATION OF GHOST MODULE

To assess the effectiveness of integrating the Ghost module, we analyzed the performance disparities between models with and without the Ghost module, as detailed in Tables 1, 2 and 3. In state (3), the model comprises approximately 6.5 million parameters and FLOPs of 15.2, representing 0.5 and 0.7 increases compared to state (1). In state (5), the model encompasses around 6.2 million parameters and FLOPs 14.8, which is 0.5 and 0.9 higher than those of state (2). For state (7), the model's parameters amount to roughly 6.5 million, with FLOPs of 15.2, indicating 0.5 and

TABLE 1. Performance results of different states on the URPC 2020 test set. Speed-GPU indicates the speed averaged over URPC 2020 test set using a RTX3060 instance.

NO.	YOLOv5	CSPCBAM	Ghost	Cluster-NMS	mAP@0.5 ↑	mAP@0.5:0.95 ↑	Speed-GPU(ms) ↓
1	✓				85.6%	49.7%	7.8
2	✓	✓			86.2%	50.9%	7.4
3	✓		✓		86.2%	50.8%	7.1
4	✓			✓	85.7%	53.8%	7.8
5	✓	✓	✓		86.4%	51.4%	**6.9**
6	✓	✓		✓	86.3%	54.7%	7.4
7	✓		✓	✓	86.6%	54.4%	7.1
8	✓	✓	✓	✓	**87.2%**	**55.1%**	**6.9**

TABLE 2. Performance results of different states on the Brackish test set. Speed-GPU indicates the speed averaged over URPC 2020 test set using a RTX3060 instance.

NO.	YOLOv5	CSPCBAM	Ghost	Cluster-NMS	mAP@0.5 ↑	mAP@0.5:0.95↑	Speed-GPU(ms) ↓
1	✓				95.8%	71.4%	6.8
2	✓	✓			96.9%	72.7%	6.6
3	✓		✓		96.3%	72.5%	6.3
4	✓			✓	95.9%	73.8%	6.8
5	✓	✓	✓		96.9%	74.2%	**6.1**
6	✓	✓		✓	97.8%	74.6%	6.6
7	✓		✓	✓	97.7%	74.4%	6.3
8	✓	✓	✓	✓	**98.6%**	**75.2%**	**6.1**

0.7 increases compared to state (4). Lastly, in state (8), the model is composed of approximately 6.2 million parameters and FLOPs of 14.8, which is 0.5 and 0.9 higher than those of state (6). After observing the experimental outcomes across both datasets, it becomes evident that the inclusion of the Ghost module within the model's feature fusion network leads to a notable reduction in model parameters and computational complexity. Consequently, this enhancement significantly improves the model's inference speed while also contributing to a certain degree of accuracy improvement. These findings demonstrate the capacity of the Ghost module to effectively diminish model parameters and enhance the model's ability to represent intricate features when integrated into the model's architecture.

3) VALIDATION OF CLUSTER-NMS

Tables 1 and 2 present the performance of CGC under various configurations on the URPC 2020 and Brackish test sets. As depicted in Table 1, the mAP of state (4) is 85.7% and 53.8%, respectively. This represents a marginal increase of 0.1% and a more substantial 4.1% improvement compared to state (1). State (6) achieves an mAP of 86.3% and 54.7%, showing a slight 0.1% rise and a notable 3.8% increase over state (2). State (7) records an mAP of 86.6% and 54.4%, indicating a 0.4% and 3.6% boost compared to state (3). Lastly, state (8) attains an mAP of 87.2% and 55.1%, marking an increase of 0.8% and 3.7% relative to state (5). Similarly, in Table 2, state (4) exhibits an mAP of 95.9% and 73.8%, representing a slight 0.1% rise and a 2.4% improvement over state (1). State (6) achieves an mAP of 97.8% and 74.6%, indicating a 0.9% and 1.9% increase compared to state (2). State (7) records an mAP of 97.7% and 74.4%, marking a 1.4% and 1.9% improvement over state (3). Lastly, state (8) attains an mAP of 98.6% and 75.2%, showing a 1.7% and 1.0% increase relative to state (5). From the experimental results across both datasets, it is evident that Cluster-NMS, as a post-processing algorithm, maintains consistent parameters and computational complexity when integrated into the model. Notably, the accuracy of the Cluster-NMS model significantly outperforms that of the traditional NMS model, while inference speed remains largely unchanged. In summary, Cluster-NMS substantially enhances detection accuracy under high IoU thresholds while preserving the model's inference speed. This improvement is attributed to the fractional penalty mechanism introduced by Cluster-NMS. Furthermore, Cluster-NMS's ability to process different clusters in parallel mitigates the speed impact resulting from increased computational complexity, resulting in efficient processing.

TABLE 3. Parameters and FLOPs of different states. As Cluster-NMS is a post-processing algorithm, it does not impact the model parameters and FLOPs.

NO.	YOLOv5	CSPCBAM	Ghost	Parameters(M)↓	FLOPs(G)↓
1, 4	✓			7.0	15.9
2, 6	✓	✓		6.5	15.2
3, 7	✓		✓	6.7	15.7
5, 8	✓	✓	✓	**6.2**	**14.8**

D. QUALITATIVE ANALYSIS

Grad-CAM is a technique employed for comprehending and visualizing specific image regions crucial for particular predictions made by CNNs or similar deep-learning models. This technique plays a pivotal role in providing profound insights

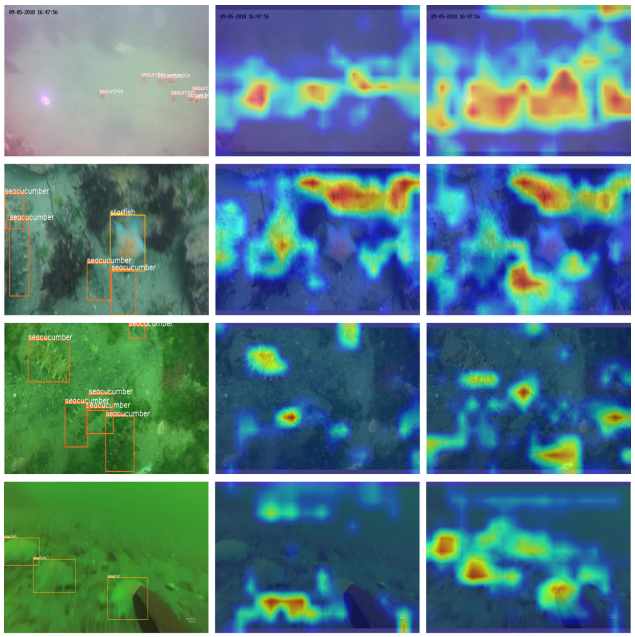


FIGURE 5. Real labels of four sample pictures from URPC 2020 dataset and Grad-CAM maps of different states. The redder the color, the more attention the model pays to this area. Column (a) is the real positions and categories of the target, column (b) is the Grad-CAM maps of YOLOv5, and column (c) is the Grad-CAM maps of YOLOv5 with CSPCBAM.

into why a model reaches specific decisions. However, it's worth noting that the primary function of the Ghost module is to reduce network parameters and computational load, while Cluster-NMS, serving as a post-processing algorithm, predominantly enhances the detection of occluded targets. Neither of these two components contributes directly to the model's understanding of target features. CSPCBAM enhances the comprehensive comprehension of fuzzy targets by dynamically weighting attention across both channel and spatial dimensions. Consequently, in this study, we focus solely on generating Grad-CAM maps for YOLOv5 and YOLOv5 with CSPCBAM to assess the effectiveness of the CSPCBAM module.

With the aid of Grad-CAM, we conduct an in-depth analysis of the role played by CSPCBAM in enhancing underwater target detection performance. Figure 5 showcases the Grad-CAM maps generated when four distinct sample images from the URPC 2020 dataset were processed by different models. In the diagram, the first column reveals the actual location and category of the target, while the second column displays the Grad-CAM map generated by YOLOv5. The third column displays the Grad-CAM map generated by YOLOv5 with CSPCBAM. As depicted in the first row of Figure 5, during the detection of the indistinct seaurchin target on the right, both YOLOv5 and YOLOv5 with CSPCBAM focus their attention on the region housing seaurchin features. However, YOLOv5 with CSPCBAM exhibits a broader scope of attention and assigns higher importance weights. In the second and third rows of Figure 5, while detecting seacucumber targets with similar backgrounds, YOLOv5 tends to treat them as part of the background. In contrast, YOLOv5 with CSPCBAM pays considerable attention to nearly all areas containing seacucumber targets. In the fourth row of Figure 5, when dealing with features affected by motion blur, YOLOv5 tends to lose focus, whereas YOLOv5 with CSPCBAM manages to concentrate on the region housing the starfish target. In summary, these cases demonstrate the adaptability of our CSPCBAM within dynamic underwater scenes featuring fuzzy and small targets and its robust performance in addressing the challenges presented by such dynamic environments.

Next, we validate the efficacy of each component by visualizing the model's output results. To enable a visual comparison of how each component impacts the original model's detection performance before and after integration, we have selected and visualized the detection results for four different states: YOLOv5, YOLOv5 with CSPCBAM, YOLOv5 with CSPCBAM and Cluster-NMS, and YOLOv5 with CSPCBAM, Cluster-NMS, and Ghost. We selected four sample images from the test set, which can well represent four situations that may be encountered in a low-visibility underwater environment: (1) distant blurred target (2) near blurred target (3) near sheltered target (4) far sheltered target. Figure 6 presents the visualization results for the four sample images previously discussed, once they have been processed by models in different states. Column (a) shows the actual bounding box and category of the target. Column (b) displays the test results of YOLOv5. Column (c) shows the detection results of YOLOv5 with CSPCBAM. Column (d) illustrates the detection results of YOLOv5 with CSPCBAM and Cluster-NMS. Finally, Column (e) exhibits the detection results of YOLOv5 with CSPCBAM, Cluster-NMS, and Ghost. In Columns (b) and (c) of Figure 6, it is evident that YOLOv5 with CSPCBAM can detect more fuzzy targets that the original YOLOv5 model cannot. When we examine the experimental results presented in Columns (c) and (d) in the comparison Figure 6, we observe that the addition of Cluster-NMS enhances the model's ability to accurately detect partially occluded seacucumber targets and starfish targets compared to YOLOv5 with CSPCBAM. Columns (d) and (e) of Figure 6 demonstrate that in comparison to YOLOv5 with CSPCBAM and Cluster-NMS, the addition of Ghost enables the model to detect more seaurchin targets and seacucumber targets that closely resemble the background.

Upon analyzing Figure 6, we observed a notable increase in the number of false positives (FP) in certain instances attributable to our proposed approach. We posit that this rise in FP may be linked to a concurrent increase in true positives (TP). The escalation of FP tends to elevate the recall rate, prompting the model to lean towards classifying more samples as positive. In specific cases, this inclination might result in the misjudgment of negative samples as positive, consequently contributing to the overall increase in FP. For our future endeavors, we plan to delve into discussions surrounding the adjustment of classification thresholds and

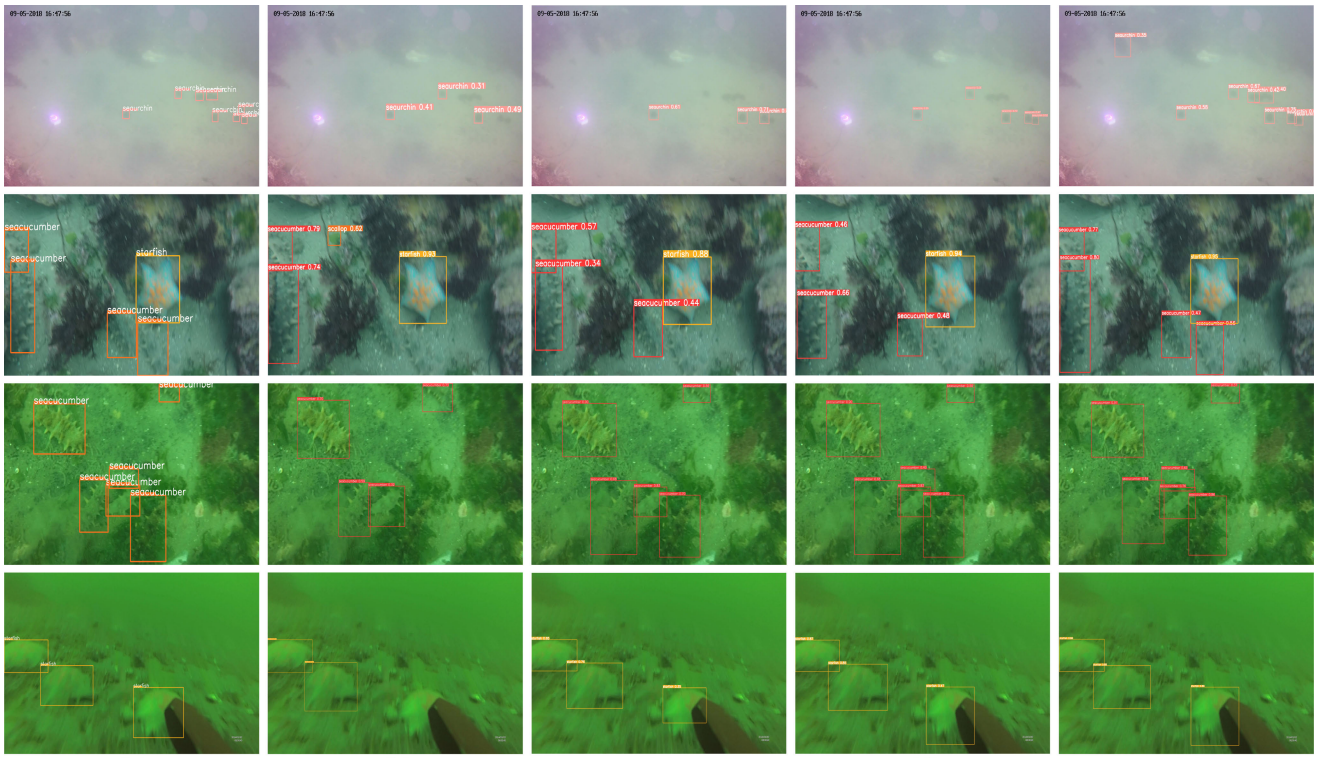


FIGURE 6. The visual detection results of CGC-YOLO and different states on four sample images from URPC 2020 dataset. The number represents the probability of prediction categories. Column (a) shows the real position and category of the sample picture. Column (b) shows the detection results of YOLOv5. Column (c) shows the detection results of YOLOv5 with CSPCBAM. Column (d) shows the detection results of YOLOv5 with CSPCBAM and Cluster-NMS. Column (e) shows the detection results of YOLOv5 with CSPCBAM, Cluster-NMS and Ghost(CGC-YOLO).

weights. This strategic adjustment aims to strike a balance between Precision and Recall, intending to mitigate the rise in false positives and enhance accuracy without compromising other pivotal indicators.

In summary, the visualization results corroborate that CSPCBAM improves the performance of detecting fuzzy targets, Cluster-NMS enhances the detection of occluded targets, and Ghost augments the model's capacity to represent complex features, thereby elevating overall performance.

E. COMPARISON WITH THE STATE-OF-THE-ART

After conducting ablation studies and qualitative analysis, we compared the performance of CGC-YOLO with models from the DETR series [48], [49], [50], [51] and some advanced models ([23], [26], [29], [32]) mentioned in related work on URPC 2020 and Brackish datasets. The author of [32] did not provide a specific name for the proposed model; therefore, in this paper, we have temporarily named it YOLOX-MobileViT-DCA for the sake of discussion and reference. The experimental results are presented in Table 4, Table 5, and Table 6, demonstrating CGC-YOLO's superiority over other models. On the URPC 2020 dataset, CGC-YOLO exhibits outstanding performance across all metrics. Notably, compared with the best-performing EDR-D0 model, CGC-YOLO achieves a 0.4% and 2.1% increase in overall category mAP, reaching 87.2% and 55.1%, respectively. Moreover, it reduces parameters and FLOPs by 0.8M and 1.1G, respectively, while improving processing speed by 1.6ms per image. For the Brackish dataset, CGC-YOLO outperforms all other models. It achieves mAP@0.5 and mAP@0.5:0.95 scores of 98.6% and 77.4%, respectively, surpassing YOLOX-MobileViT-DCA by 1.1 and 0.9 percentage points, which are the best results among other models. In terms of model size and speed, CGC-YOLO possesses approximately 6.2M parameters, 14.8 GFLOPs, and a processing speed of 6.1ms per image, surpassing the DETR series and other related models for underwater target detection.

However, it's worth noting that, based on the experimental results from the two datasets, CGC-YOLO's mAP@0.5 is not as significantly improved compared to other models as its mAP@0.5:0.95. This observation led us to carefully examine the datasets, where we discovered a highly imbalanced class distribution. In the URPC training set, each category accounts for approximately 14.2%, 56.0%, 15.0%, and 14.5% of the total number. In contrast, the Brackish training set exhibits even greater class imbalance, with each category comprising around 9.5%, 31.4%, 21.5%, 1.7%, 1.6%, and 34.3%, respectively. This class imbalance poses challenges for the model to effectively learn the characteristics of certain classes. Therefore, it becomes imperative to employ

TABLE 4. Performance results of CGC-YOLO and other related models on the URPC 2020 test set. Speed-GPU indicates the speed averaged over URPC 2020 test set using a RTX3060 instance.

Methods	mAP@0.5↑	mAP@0.5:0.95↑	Speed-GPU(ms) ↓
DETR [49]	60.7%	27.1%	36
Deformable-DETR [50]	67.2%	31.4%	35
DN-DETR [51]	77.6%	37.8%	67
DINO [52]	78.5%	39.6%	72
EDR-D0 [23]	86.8%	53.0%	8.7
YOLOv4-Tinier [29]	85.7%	49.2%	11.4
YOLOv5s-CA [26]	86.2%	51.4%	7.8
YOLOX-MobileViT-DCA [32]	85.8%	50.5%	18.2
CGC-YOLO	**87.2%**	**55.1%**	**6.9**

TABLE 5. Performance results of CGC-YOLO and other related models on the Brackish test set. Speed-GPU indicates the speed averaged over Brackish test set using a RTX3060 instance.

Methods	mAP@0.5↑	mAP@0.5:0.95↑	Speed-GPU(ms) ↓
DETR [49]	87.4%	50.6%	34.5
Deformable-DETR [50]	95.7%	55.8%	32
DN-DETR [51]	96.6%	67.8%	60
DINO [52]	95.8%	70.4%	65
EDR-D0 [23]	97.2%	74.1%	8.0
YOLOv4-Tinier [29]	96.4%	72.9%	7.2
YOLOv5s-CA [26]	96.8%	73.5%	6.8
YOLOX-MobileViT-DCA [32]	97.5%	74.3%	14.8
CGC-YOLO	**98.6%**	**75.2%**	**6.1**

TABLE 6. Parameters and FLOPs of CGC-YOLO and other related models.

Methods	Parameters(M)↓	FLOPs(G)↓
DETR [49]	41.3	86
Deformable-DETR [50]	34.4	78
DN-DETR [51]	44.2	91
DINO [52]	47.3	98
EDR-D0 [23]	7.0	15.9
YOLOv4-Tinier [29]	7.8	18.7
YOLOv5s-CA [26]	7.8	18.6
YOLOX-MobileViT-DCA [32]	10.6	25.4
CGC-YOLO	**6.2**	**14.8**

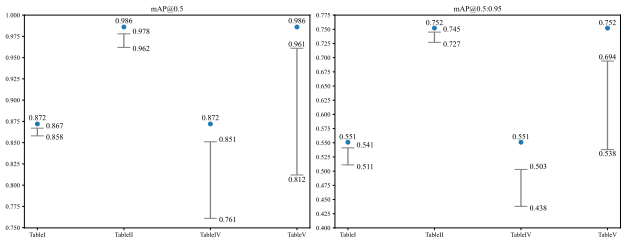


FIGURE 7. Confidence intervals for mAP@0.5 and mAP@0.5:0.95 in Tables 1, 2, 4, 5 and at a 95% confidence level. The blue points denote the mAP value of CGC-YOLO.

TABLE 7. Results of independent sample T-test for mAP@0.5 and mAP@0.5:0.95 in Table 1, Table 2, Table 4, Table 5 and at a 95% confidence level. The null hypothesis (H_0) assumes that CGC-YOLO's mAP is equivalent to the mean of other models. A p-value below 0.05 indicates rejection of the null hypothesis.

Test sample(n=4)	t	p-value
mAP@0.5 in Table 1	-5.14	0.001
mAP@0.5 in Table 2	-4.61	0.002
mAP@0.5 in Table 4	-3.39	0.009
mAP@0.5 in Table 5	-3.11	0.014
mAP@0.5:0.95 in Table 1	-3.34	0.012
mAP@0.5:0.95 in Table 2	-3.54	0.009
mAP@0.5:0.95 in Table 4 mAP@0.5:0.95	-4.9	0.001
mAP@0.5:0.95 in Table 5 mAP@0.5:0.95	-3.41	0.009

data augmentation techniques to balance the number of samples across different categories. In conclusion, the CGC-YOLO model proposed in this paper demonstrates versatility and effectiveness, making it highly suitable for real-time inference and deployment in underwater vehicles.

F. SIGNIFICANCE ANALYSIS

To assess whether CGC-YOLO significantly differs from the ablation model and other existing models, we examined the confidence intervals of all samples containing mAP in the tables. Figure 7 displays the confidence intervals for mAP@0.5 and mAP@0.5:0.95 in Table 1, Table 2, Table 4, and Table 5 at a 95% confidence level, alongside the mAP values of CGC-YOLO. As illustrated in Figure 7, the mAP value of CGC-YOLO noticeably deviates from the sample mean and falls outside the obtained confidence interval, suggesting potential significant differences between our proposed model, the ablation model, and other existing models at a 95% confidence level. To further substantiate the significance of these differences, we conducted an independent sample t-test. The experimental results are

presented in Table 7. Observing the data in Table 7 reveals that the p-values of the T-test for mAP in Tables 1, Table 2, Table 4, and Table 5 are all below 0.05. This statistical evidence supports the assertion that there are significant differences between the mAP values of our proposed CGC-YOLO on two datasets, ablation models, and other existing models.

V. DISCUSSION AND CONCLUSION

When an intelligent robot performs object detection in the underwater environment, its camera is typically affected by light absorption when shooting images. Furthermore, the light scattering of impurities in the water further contributes to the image blurring, obscuring critical details. In addition, some marine creatures that like to live in groups (such as seaurchins, starfish, and scallops) sometimes block each other. The key to improving the model prediction performance is to understand the characteristics of fuzzy objects in fine granularity and avoid errors to suppress occluded objects. The images in the URPC 2020 and Brackish datasets cover several common marine organisms and a variety of complex underwater scenes. Moreover, target blurring and occlusion problems are also most common in these datasets. CGC-YOLO has shown good performance on both datasets compared with some improved methods in previous studies.

In underwater images, the presence of large targets, even when blurred, typically entails a discernible overall outline, facilitating the model's identification. Consequently, the accuracy of underwater target detection is often more susceptible to the challenge posed by small, fuzzy targets. To address this issue, conventional approaches introduce an attention module as an independent entity between convolution modules in the backbone network. In order to harmonize the feature extraction process of the convolution module with the weighting function of the attention module, we introduce the CSPCBAM module. This module incorporates CBAM within the CSP structure of the convolution module, aimed at enhancing the model's accuracy in detecting fuzzy targets and fortifying its resilience to small targets. This integration ultimately contributes to an improvement in overall detection accuracy. The majority of the previous studies have reduced the accuracy of the model detection when using depth-separable convolution to minimize the size and parameters. The Ghost module generates more intrinsic feature maps with inherent feature information through simple linear transformation, minimizing the parameters and improving the feature expression performance of the network. When the SPM is added to the NMS algorithm to detect the blocked target, most previous studies inevitably obtain increased time complexity and decreased model inference speed. Cluster-NMS can be executed concurrently on the inherent cluster of the detection boxes, thus improving the detection speed. By optimizing the model size and processing speed, our model excels in achieving enhanced real-time detection capabilities. This enables more effective deployment in complex dynamic scenarios, including situations with motion blur and occlusion.

In this study, we attempted to use Cluster-NMS to address the problem of missing occluded targets. However, the Cluster-NMS algorithm has a certain sensitivity to the threshold value. If the threshold is considerably low, then the detection will be missed. If the threshold is particularly high, then it will bring false positive examples. Therefore, we urgently need a loss function that can require the prediction boxes to be close to the target boxes and away from other real boxes that do not belong to the target to enhance the performance of the detection model and decrease the sensitivity of Cluster-NMS to the threshold.

This work proposes an improved model based on YOLOv5. Firstly, the CSPCBAM module is designed and integrated into the entire network, resulting in improved accuracy for detecting fuzzy and small targets. The Ghost module is then incorporated into the YOLOv5's neck section to decrease calculations while maintaining accuracy. Finally, the Cluster-NMS algorithm is introduced as a post-processing method to improve the detection performance and detection speed of occluded objects. The ablation study shows that this method has higher detection accuracy than YOLOv5s and fewer calculations and parameters. Comparisons with other enhanced methods illustrate the effectiveness of CGC-YOLO in improving model accuracy and detection speed. The improved model offers more real-time detection capabilities and is easier to deploy in underwater vehicles. In the future, we intend to develop a loss function that ensures the prediction boxes are distant from other unrelated real boxes, which will enhance the model's performance in scenarios with multiple occlusions. In addition, addressing the current class imbalance issue in underwater target detection datasets is of paramount importance. To this end, we propose to investigate data enhancement techniques centered around data synthesis. Techniques such as Poisson fusion [52] can be employed to augment the number of samples for minority classes by combining existing ones while preserving the inherent diversity of the data. We will also discuss adjusting the classification threshold and weight to balance precision and recall, so as to reduce the increase of false positives and improve the accuracy without damaging other key indicators.

ACKNOWLEDGMENT

In the process of writing this article, the authors were carefully guided and helped by Prof. Tong Qingshan. At the same time, Changsha University of Science and Technology provided them with the necessary experimental conditions, enabling them to complete their research. Here, they would like to express their most sincere thanks to them and all the members of the research group.

REFERENCES

[1] J. Zhang, X. Xiang, and W. Li, "Advances in marine intelligent electromagnetic detection system, technology, and applications: A review," *IEEE Sensors J.*, vol. 23, no. 5, pp. 4312–4326, Mar. 2023.

[2] C. V. Mahamuni and Z. M. Jalauddin, "Intrusion monitoring in military surveillance applications using wireless sensor networks (WSNs) with deep learning for multiple object detection and tracking," in *Proc. Int. Conf. Control, Autom., Power Signal Process. (CAPS)*, Dec. 2021, pp. 1–6.

[3] T. Li, S. Rong, B. He, and L. Chen, "Underwater image deblurring framework using a generative adversarial network," in *Proc. OCEANS*, Feb. 2022, pp. 1–4.

[4] J. Zhou, T. Xu, W. Guo, W. Zhao, and L. Cai, "Underwater occlusion object recognition with fusion of significant environmental features," *J. Electron. Imag.*, vol. 31, no. 2, Mar. 2022, Art. no. 023016.

[5] C.-H. Yeh, C.-H. Lin, L.-W. Kang, C.-H. Huang, M.-H. Lin, C.-Y. Chang, and C.-C. Wang, "Lightweight deep neural network for joint learning of underwater object detection and color conversion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6129–6143, Nov. 2022.

[6] X. Wei, L. Yu, S. Tian, P. Feng, and X. Ning, "Underwater target detection with an attention mechanism and improved scale," *Multimedia Tools Appl.*, vol. 80, no. 25, pp. 33747–33761, Oct. 2021.

[7] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. Lau, "BiFormer: Vision transformer with bi-level routing attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 10323–10333.

[8] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[9] Y. Liu, Z. Shao, and N. Hoffmann, "Global attention mechanism: Retain information to enhance channel-spatial interactions," 2021, *arXiv:2112.05561*.

[10] R. Chen, H. Huang, Y. Yu, J. Ren, P. Wang, H. Zhao, and X. Lu, "Rapid detection of multi-QR codes based on multistage stepwise discrimination and a compressed MobileNet," *IEEE Internet Things J.*, vol. 10, no. 18, pp. 15966–15979, Jun. 2023.

[11] Y. Liu, Z. Wang, R. Wang, J. Chen, and H. Gao, "Flooding-based MobileNet to identify cucumber diseases from leaf images in natural scenes," *Comput. Electron. Agricult.*, vol. 213, Oct. 2023, Art. no. 108166.

[12] S. G. Sambandam, R. Purushothaman, R. U. Baig, S. Javed, V. T. Hoang, and K. Tran-Trung, "Intelligent surface defect detection for submersible pump impeller using MobileNet v2 architecture," *Int. J. Adv. Manuf. Technol.*, vol. 124, no. 10, pp. 3519–3532, Feb. 2023.

[13] H. Yang, J. Liu, G. Mei, D. Yang, X. Deng, and C. Duan, "Research on real-time detection method of rail corrugation based on improved ShuffleNet v2," *Eng. Appl. Artif. Intell.*, vol. 126, Nov. 2023, Art. no. 106825.

[14] Y. Wang, X. Xu, Z. Wang, R. Li, Z. Hua, and H. Song, "ShuffleNet-triplet: A lightweight RE-identification network for dairy cows in natural scenes," *Comput. Electron. Agricult.*, vol. 205, Feb. 2023, Art. no. 107632.

[15] L. Zheng, M. Zhao, J. Zhu, L. Huang, J. Zhao, D. Liang, and D. Zhang, "Fusion of hyperspectral imaging (HSI) and RGB for identification of soybean kernel damages using ShuffleNet with convolutional optimization and cross stage partial architecture," *Frontiers Plant Sci.*, vol. 13, pp. 1–10, Jan. 2023.

[16] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[17] F. Chen, L. Zhang, S. Kang, L. Chen, H. Dong, D. Li, and X. Wu, "Soft-NMS-Enabled YOLOv5 with SIOU for small water surface floater detection in UAV-captured images," *Sustainability*, vol. 15, no. 14, p. 10751, Jul. 2023.

[18] L. Zhang, C. Lu, H. Xu, A. Chen, L. Li, and G. Zhou, "MMFNet: Forest fire smoke detection using multiscale convergence coordinated pyramid network with mixed attention and fast-robust NMS," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18168–18180, Aug. 2023.

[19] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing geometric factors in model learning and inference for object detection and instance segmentation," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022.

[20] Z. Fan, G. Hu, X. Sun, G. Wang, J. Dong, and C. Su, "Self-attention neural architecture search for semantic image segmentation," *Knowl.-Based Syst.*, vol. 239, Mar. 2022, Art. no. 107968.

[21] A. Pandey and D. Wang, "Self-attending RNN for speech enhancement to improve cross-corpus generalization," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 1374–1385, 2022.

[22] J. Stacey, Y. Belinkov, and M. Rei, "Supervising model attention with human explanations for robust natural language inference," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 10, pp. 11349–11357.

[23] J. Jia, M. Fu, X. Liu, and B. Zheng, "Underwater object detection based on improved EfficientDet," *Remote Sens.*, vol. 14, no. 18, p. 4487, Sep. 2022.

[24] M. L. Mekhalfi, C. Nicolò, Y. Bazi, M. M. A. Rahhal, N. A. Alsharif, and E. A. Maghayreh, "Contrasting YOLOv5, transformer, and EfficientDet detectors for crop circle detection in desert," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[25] Z. Meng, Y. Hong, L. Haiqing, X. Jingwen, C. Siqi, G. Lishuai, Z. Peng, W. Sixue, and Z. Guowei, "Farmed fish detection by combining sknet and YOLOv5 deep learning," *J. Ocean Univ. China*, vol. 37, no. 2, pp. 312–319, 2022.

[26] G. Wen, S. Li, F. Liu, X. Luo, M.-J. Er, M. Mahmud, and T. Wu, "YOLOv5s-CA: A modified YOLOv5s network with coordinate attention for underwater target detection," *Sensors*, vol. 23, no. 7, p. 3367, Mar. 2023.

[27] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2021, pp. 13713–13722.

[28] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[29] K. Yu, Y. Cheng, Z. Tian, and K. Zhang, "High speed and precision underwater biological detection based on the improved YOLOV4-tiny algorithm," *J. Mar. Sci. Eng.*, vol. 10, no. 12, p. 1821, Nov. 2022.

[30] L. Wang, K. Zhou, A. Chu, G. Wang, and L. Wang, "An improved light-weight traffic sign recognition algorithm based on YOLOv4-tiny," *IEEE Access*, vol. 9, pp. 124963–124971, 2021.

[31] Y.-M. Zhang, C.-C. Lee, J.-W. Hsieh, and K.-C. Fan, "CSL-YOLO: A new lightweight object detection system for edge computing," 2021, *arXiv:2107.04829*.

[32] Y. Sun, W. Zheng, X. Du, and Z. Yan, "Underwater small target detection based on YOLOX combined with MobileViT and double coordinate attention," *J. Mar. Sci. Eng.*, vol. 11, no. 6, p. 1178, Jun. 2023.

[33] S. Mehta and M. Rastegari, "MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer," 2021, *arXiv:2110.02178*.

[34] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[35] F. Li and F. Li, "Bag of tricks for fabric defect detection based on cascade R-CNN," *Textile Res. J.*, vol. 91, nos. 5–6, pp. 599–612, Mar. 2021.

[36] M. Wang, B. Fu, J. Fan, Y. Wang, L. Zhang, and C. Xia, "Sweet potato leaf detection in a natural scene based on faster R-CNN with a visual attention mechanism and DIoU-NMS," *Ecolog. Informat.*, vol. 73, Mar. 2023, Art. no. 101931.

[37] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[38] Y. Chen, H. Xu, X. Zhang, P. Gao, Z. Xu, and X. Huang, "An object detection method for bayberry trees based on an improved YOLO algorithm," *Int. J. Digit. Earth*, vol. 16, no. 1, pp. 781–805, Oct. 2023.

[39] Y. Gao, Y. Ding, W. Xiao, Z. Yao, X. Zhou, X. Sui, Y. Zhao, and Y. Zheng, "A semi-supervised learning framework for micropapillary adenocarcinoma detection," *Int. J. Comput. Assist. Radiol. Surgery*, vol. 17, no. 4, pp. 639–648, Apr. 2022.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[41] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan, and Q. Ren, "Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles," 2022, *arXiv:2206.02424*.

[42] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9156–9165.

[43] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.

[44] C. Liu, H. Li, S. Wang, M. Zhu, D. Wang, X. Fan, and Z. Wang, "A dataset and benchmark of underwater object detection for robot picking," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2021, pp. 1–6.

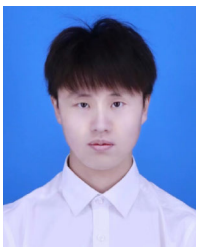
[45] M. Pedersen, J. B. Haurum, R. Gade, T. B. Moeslund, and N. Madsen, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 18–26.

[46] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[47] S. Zhao, J. Zheng, S. Sun, and L. Zhang, "An improved YOLO algorithm for fast and accurate underwater object detection," *Symmetry*, vol. 14, no. 8, p. 1669, Aug. 2022.
[48] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
[49] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.
[50] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "DN-DETR: Accelerate DETR training by introducing query DeNoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13609–13617.
[51] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.
[52] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.

ZHENG ZHANG received the B.S. degree in information and computing science from Shanghai Ocean University, Shanghai, China, in 2021. He is currently pursuing the master's degree with the School of Mathematics and Statistics, Changsha University of Science and Technology, China. His current research interests include intelligent underwater robots, computer vision, deep learning, and machine learning.

QINGSHAN TONG received the Graduate degree majoring in mathematics from Changsha University of Electric Power, in 1994. In October 2005, he was an Associate Professor with the School of Mathematics and Computational Science, Changsha University of Science and Technology. His research interests include artificial intelligence, big data analysis, and the optimal allocation of human resources.



XIAOFEI HUANG received the B.S. degree in mathematics and statistics from Changsha University of Science and Technology, Changsha, China, in 2020, where he is currently pursuing the M.S. degree with the School of Mathematics and Statistics. His current research interests include computerized medical diagnosis, pattern recognition, deep learning, and machine learning.