AB-YOLOv8: Attention-based Feature Extraction model for Underwater Object Detection

Pratima Sarkar[1,2], Bitan Misra[2], Sourav De[3], Sandeep Gurung[1]
[1]Sikkim Manipal Institute of Technology, Department of Computer Science and Engineering, Majhitar, Sikkim-737132, India
[2]Department of Computer Science and Engineering, Techno International New Town, Kolkata-900156, India
[3]Department of Computer Science and Engineering, Government College of Engineering and Textile Technology, Serampore 12, William Carey Road, Serampore, Hooghly, Pin-712201, India
E-mail: pratima.sarkar@tint.edu.in, bitan.misra@tint.edu.in, dr.sourav.de79@gmail.com, sandeep.gu@smit.smu.edu.in

Keywords: Channel attention, data augmentation, spatial attention, R-CNN, underwater object detection

Received: December 23, 2024

Accurate and timely underwater object detection is crucial in the field of marine environmental engineering. The detection of such targets has been improved recently using techniques based on Convolutional Neural Networks (CNN). However, the processing performance of deep neural networks is typically inadequate due to their high parameter requirements. Accurate detection is difficult with current techniques when dealing with small, close-packed underwater targets. In order to overcome these problems, the proposed work combined YOLOv8 with different attention modules and proposed a novel neural network model to enhance underwater object detection capabilities. In this research, AB-YOLOv8 is proposed, which adds the attention mechanism to the original YOLOv8 design. To be more precise, the proposed work introduced four attention modules, Convolutional Block Efficient Channel Attention (ECA), Shuffle Attention (SA), Global Attention Mechanism (GAM), and Attention Module (CBAM), to create the enhanced models and train them in the aquarium dataset. Each of the attention blocks is combined with YOLOv8 to improve the performance of the entire object detection. The residual block is introduced into the CBAM to optimize the performance of the CBAM. The detailed experiments are conducted on the aquarium dataset, and various performance assessment parameters are used, like mAP, FLOPS, Params, inference time, etc. After performing the experiment, it was found that ECA gives the best result out of all attention blocks and improved mAP value by 8%, also reduced the number of parameters generated during training. To validate the work, we also performed the experiment on the Brackish dataset, and we found that ECA outperforms other attention mechanisms with YOLOv8.

Povzetek: Zasnovan je nov model AB-YOLOv8 z mehanizmi pozornosti (ECA, CBAM, SA, GAM) za izboljšanje zaznavanja podvodnih objektov. Model ECA-YOLOv8 je izkazal najboljše rezultate: izboljšal je metriko mAP v primerjavi z osnovnim YOLOv8 in zmanjšal število parametrov.

1 Introduction

Underwater object recognition is a crucial stage in image processing that is important for a number of applications, including marine sciences and the upkeep and repair of subaquatic infrastructure. One of the most difficult study areas in modern computer vision technologies is the detection of underwater objects [1]. Specifically, the widespread deployment of digital cameras on Autonomous Underwater Vehicles (AUVs) and Unmanned Underwater Vehicles (UUVs) has led to an exponential increase in the availability of underwater imagery in recent years [2]. The primary obstacles to underwater vision are the increased expense of the devices, their intricate configuration, and the distortion of light and signal propagation caused by the water medium [3]. The propagation of light in underwater environments is particularly affected by phenomena such as absorption and scattering, which have a significant impact on visual perception [4, 5]. In recent years, generic object detection algorithms have demonstrated their exceptional performance. In digital image processing for object recognition and classification, deep learning, also referred to as deep machine learning or deep structured learning-based techniques, has recently seen significant success [6]. Thus, they are attracting the interest and popularity of the computer vision research community rather quickly [7]. However, these approaches are not sufficiently capable of handling underwater object detection due to the following challenges: (1) Real-world applications typically feature small objects with hazy photos [8], and (2) real-world applications and underwater datasets have images with heterogeneous noise [9]. When taking into account underwater variables like sufficient light, reasonable current intensity, and clear underwater eyesight, simple underwater target-detection tech-

niques can be used more effectively. The primary features extracted by early conventional detection techniques were color, texture, and geometry. As the deep learning technique continues to advance, neural networks have emerged as underwater target-detection frameworks that enable target detection by identifying and locating objects in photos [10]. However, underwater image quality deteriorates due to less-than-ideal conditions in practice, which consequently impairs the accuracy of detection. Convolutional Neural Networks (CNNs) [11] have made significant strides in object detection in recent years due to their potent feature learning and transfer learning capabilities, which have drawn increasing attention from the discipline of computer vision. The application of CNN to object detection for improved performance is therefore a significant domain of research work [12]. YOLOv8 differs from previous YOLO models in several significant ways. Its transformer-based architecture, which improves accuracy and performance, especially for small and difficult-to-detect objects, is one of the biggest upgrades.

In order to effectively address the challenges associated with underwater object detection, the proposed research integrates the YOLOv8 [13] architecture with various attention modules, culminating in the development of a novel neural network model designed to significantly enhance detection capabilities in underwater environments. The unique combination of YOLOv8 with sophisticated attention mechanisms and the calculated improvements made to the CBAM constitute the work's originality. The following are the primary contributions of this paper:

1 This innovative approach is encapsulated in the newly introduced model, termed AB-YOLOv8, which incorporates an attention mechanism into the foundational design of YOLOv8. This study introduces four distinct attention modules: Convolutional Block Efficient Channel Attention (ECA) [14], Shuffle Attention (SA) [15], Global Attention Mechanism (GAM) [16], and Convolutional Block Attention Module (CBAM) [17]. Each of these modules is strategically combined with the YOLOv8 framework to create enhanced models that are specifically trained on the aquarium dataset. The integration of these attention blocks is aimed at improving the overall performance of object detection tasks, particularly in the challenging underwater context, where visibility and clarity are often compromised.

2 Additionally, the study improves the CBAM by adding a residual block, which helps to maximize its efficiency. This innovation makes better feature extraction and representation possible, which enhances the model's capacity to identify items in intricate underwater environments.

3 The success of the suggested models is evaluated using a range of performance assessment metrics, such as mean Average Precision (mAP), FLOPS (Floating Point Operations Per Second), number of parameters (Params), and inference time [18]. In real-time underwater detection applications, these measures are crucial for understanding the trade-offs between accuracy and processing efficiency.

The structure of the paper is as follows. In Section 2, the relevant literature is discussed. The network architecture and adopted approach are presented in Section 3. The dataset description is given in Section 4, and the experimental evaluation parameters are shown in Section 5. Experimental results and discussions are included in Section 7 and 8 respectively. Future work, our findings, and research outlook are summed up in Section 9.

2 Literature review

Underwater object detection can be accomplished by different two-stage and single-stage object detectors. The most popular two-stage detectors are R-CNN, Fast R-CNN, and Faster R-CNN. R-CNN [19] is performing better for small object detection, but it is not suitable for real-time object detection. So, many researchers have selected the single-stage object detectors, i.e., YOLO series, as the foundation for future development in order to accomplish real-time underwater object identification. The YOLO-UOD [20] optimization algorithm, a unique underwater object identification technique based on YOLOv4-tiny research, is presented in the article [21]. The suggested approach, which combines the symmetric FPN-Attention module and the symmetric dilated convolutional module, may efficiently collect important characteristics and contextual information while maintaining deep features, according to experimental results on the Brackish undersea dataset. Its underwater object detection mAP score of 87.88% is superior to YOLOv5s and YOLOv5m and higher than YOLOv4-Tiny's score of 77.38%. In [22], the Transformer encoder and a coordinate attention module were integrated into YOLOv5 to create a new detection network called TC-YOLO. Underwater picture enhancement was done using the CLAHE [23] algorithm, while label assignment in training was done using the optimal transport assignment approach. By combining these methods, our suggested strategy maintained computational efficiency for real-time underwater detection tasks while achieving state-of-the-art performance on the RUIE2020 [24] dataset. The attachment of the coordinate attention module to the end of the neck was found to be a very successful and efficient method of enhancing detection networks' performance in the ablation experiments. Article [25] includes the plug-and-play mDFLAM with YOLO detectors to satisfy the high-precision and real-time demands for underwater object detection. By enhancing the quality of feature fusion between scales, the full-port embedding significantly reinforces the expression of semantic information. Using a lightweight backbone network built on deformable convolution YOLOv3, article [26] proposes a dynamic YOLO de-

tector with certain specialized designs for small item identification. Experimental findings on the Pascal VOC and MS COCO datasets further support the superiority of the suggested model. Article [27] proposes a high detection accuracy cascade model based on the UGC-YOLO network structure. Additionally, PPM pooling is added to the top layer network for the purpose of aggregating semantic data, and deformable convolution is utilized to capture long-range semantic dependencies. Lastly, a multi-scale weighted fusion method for learning semantic data at various scales is introduced. The suggested approach has been shown through experiments on an underwater test dataset to be able to identify aquatic targets in intricately deteriorated underwater images. In order to decrease feature interference and increase detection accuracy, an enhanced YOLO detection technique without anchor points is presented [28], in which the detection and recognition features are kept apart. Additionally, a technique for improving underwater photos based on Retinex is also suggested. To confirm the efficacy of the suggested improved YOLO detection technique, pertinent tests based on underwater datasets are carried out.In order to create a quick, precise, and compact neural network model that can identify goldfish breeds in real time, the authors of the research [29] examine the impact of shrinking the size of the pre-trained MobileNetV2, which serves as the foundation of the YOLOv2 object detection framework. Paper [30] proposes the YOLO-SC algorithm as a solution to the problem of finding the submarine cable's position and feature information using the YOLOv3 [31, 32] prototype network because of the blurry and blue-green underwater images. Three enhanced modules work together to address the aforementioned issues. The multi-structured multi-size feature fusion module improves the efficiency of feature information extraction; the light-weighted module streamlines the prediction network and reduces identification duration; and the skip connection module, which is included in the residual network, enhances the extraction of position information. Another modified

3 Methodology

Recently, the attention mechanism has achieved outstanding outcomes in the domain of object detection. Attention blocks are capable of selecting most significant features and discarding irrelevant features. This study integrates the attention module into the neck and head component of YOLOv8 in order to improve the detection of important characteristics and reduce the impact of irrelevant information. We have chosen four attention mechanism like Efficient Channel Attention (ECA), Convolutional Block Attention Module (CBAM), Shuffle Attention (SA) and Global Attention Mechanism (GAM) for feature aggregation. ECA was selected because of its lightweight design and capacity to enhance channel-wise feature recalibration without appreciably raising model complexity. CBAM combines both channel and spatial attention, making it well-suited to capture complex underwater textures and cluttered scenes. SA helps in capturing long-range dependencies, which is beneficial when objects are partially occluded or dispersed. GAM enhances global context aggregation, helping to better differentiate between background and foreground in low-visibility underwater conditions.

YOLOv8 Architecture consists of different key components like backbone, neck, head and loss function as shown in figure 1. CSPDarknet used as backbone which contains CSP connections to increase information exchange. The neck work as a feature extractor, neck uses C2f architecture which integrates C3 modules. Neck aggregate features for detecting three different size of objects. YOLOv8 makes use of a number of detection modules to predict class probabilities, bounding boxes, and objectness scores for every grid cell in the feature map. The final detection are then obtained by averaging these forecasts. There are three types of loss funtion used during object prediction in YOLOv8 to optimize object detection those are: Binary Cross-Entropy (BCE), Distribute Focal Loss (DFL) and Complete Intersection over Union (CIoU) Loss. The classification component of YOLOv8 utilizes the Binary Cross-Entropy (BCE) Loss as its loss function, which is represented by the following equation:

$$BCE = -wt[x_n.logy_n + (1 - x_n).log(1 - y_n)] \quad (1)$$

wt represents weight, x_n is labeled and y_n is predicted value. A DFL function is specifically developed to highlight the amplification of probability values about p. The equation is given as follows:

$$DFL = P_A + P_B \quad (2)$$

Where P_A is shown in eq(3) ans P_B Shown in eq(4)

$$P_A = -[(p_{n+1} - p)log(\frac{p_{n+1} - p_n}{p_{n+1} - p_n}) \quad (3)$$

$$P_B = (p - p_n)log(\frac{p - p_n}{p_{n+1} - p_n}) \quad (4)$$

Incorporating the dimensions between the predicted bounding box and the ground truth bounding box, the CIoU Loss adds an influence factor to the Distance Intersection over Union (DIoU) Loss. The equation is as specified below:

$$CIoU = 1 - IoU + \frac{l^2}{c^2} + \frac{v^2}{1 - IoU + v} \quad (5)$$

IoU is intersection over union, d is Euclidean distance between predicted value and ground truth, l is diagonal length of predicted box, v is aspect ration of bounding box. In Figure 1 BBox-loss is combination of DFL and CIoU whereas Cls-loss represents BEC loss.

This work made modification on existing YOLOv8 architecture by adding attention module in neck and head of

YOLOv8 as illustrated in Figure 1. We have added one attention block in neck and rest all are added in head of YOLOv8. In proposed work used four different attention blocks i.e. Efficient Channel Attention (ECA), Convolutional Block Attention Module (CBAM), Shuffle Attention (SA) and Global Attention Mechanism (GAM). After incorporating these four different attention module into YOLOv8 analysed the performance of YOLOv8 with Attention block or AB-YOLOv8.

3.1 Attention modules

3.1.1 Efficient channel attention (ECA)

ECA mainly involves cross-channels and the use of 1D convolution with an adaptive single-dimensional convolution kernel as shown in Figure 2. Cross-channel interaction is an innovative method of merging characteristics to improve the representation of certain meanings. The input feature map I, which has dimensions $R^{C\times H\times W}$, is transformed into the aggregated feature F through the processes of Global Average Pooling (GAP) and cross-channel interaction. For the following equation, C refers to the cross-channel interaction.

$$F = C(GAP(I)) \tag{6}$$

ECA captures the local cross-channel interaction in aggregated data by examining the interaction between the features of each channel and their nearby k channels. The ECA method avoids utilizing 1D convolution for reducing dimensionality and effectively achieves multi-channel interaction. where the weights of the features F_i can be calculated as [14]:

$$w_i = \sigma(W) \tag{7}$$

where, W is a weight matrix and σ. is sigmoid function.

3.1.2 Convolutional block attention module (CBAM)

The CBAM [17] module has two attention sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM) as presented in Figure 3. The Channel Attention Module (CAM) is designed to enhance informative elements in the channel dimension, while the Spatial Attention Module (SAM) is designed to emphasize important features along the spatial axes. CBAM successfully captures the channel and spatial dependence in the input feature map by integrating these two attention processes. Input of CBAM is a feature map I $\in R^{C\times H\times W}$ then it is converted into 1D channel attention map $F_C \in R^{C\times 1\times 1}$ and 2D spatial map $F_S \in R^{1\times W\times H}$. So CBAM is a combination of following equations:

$$F = F_C \odot I \tag{8}$$

$$F' = F_S \odot F \tag{9}$$

where $\odot$ is element wise multiplication.

In order to efficiently calculate the channel attention, compress the spatial dimension of the input feature map. CBAM employed both Global Average Pooled (GAP) and Global Max Pooled (GMP) features concurrently. Empirical findings have demonstrated that the utilization of both features significantly enhances the representational capacity of networks, as opposed to using each feature independently. Then element wise sum (+) and sigmoid (σ) function is used to find channel attention(F_C). Equation for channel attention is as follows:

$$F_C(I) = \sigma(MLP(GAP(I)) + MLP(GMP(I))) \tag{10}$$

In this equation I is input feature matrix and MLP is Multi Layer Perception. CBAM utilizes GAP and GMP along the channel axis for spatial attention, and subsequently combines them by concatenation ($\oplus$). The concatenation output is passed through a convolutional layer, and the resulting output is then used as the input for the sigmoid (σ) function. The spatial attention (F_S) is calculated using the following method.

$$F_S(I) = \sigma[CONV(GAP(I) \oplus GMP(I))] \tag{11}$$

3.1.3 Global attention mechanism (GAM)

GAM [16] adopts similar architecture as CBAM. GAM added additional shortcut connections between channel attention and spatial attention as depicted in Figure 4. The following equation represents GAM:

$$Fout = I + [F_S(F_C(I)) \times I) \times (F_C(I)) \times I)] \tag{12}$$

where, I is input feature, F_C channel attention block and F_S is spatial attention block.

To focus on specific channels, the GAM technique utilizes a 3D permutation from the beginning to preserve three-dimensional information. Afterwards, it utilizes a MLP to enhance the channel-spatial interdependence across dimensions. Following expression shows channel attention block representation:

$$F_C(I) = \sigma[RevPermutate(MLP(Permutate(I)))] \tag{13}$$

GAM utilizes two 7×7 convolution layers to combine spatial information for spatial attention as hown in eq. (14).

$$F_S(I) = \sigma[BN(f^{7\times 7}(BN + ReLU(f^{7\times 7}(I))))]. \tag{14}$$

where, σ is sigmoid function, BN is batch normalization.

3.1.4 Shuffle attention (SA)

SA [15] divides the input feature maps into different groups, employing the Shuffle Unit to integrate both channel attention and spatial attention into one block for each group as shown in 5. Then these features are aggregated using spatial and channel attention. The channel attention mechanism utilizes the Global Average Pooling (GAP) technique

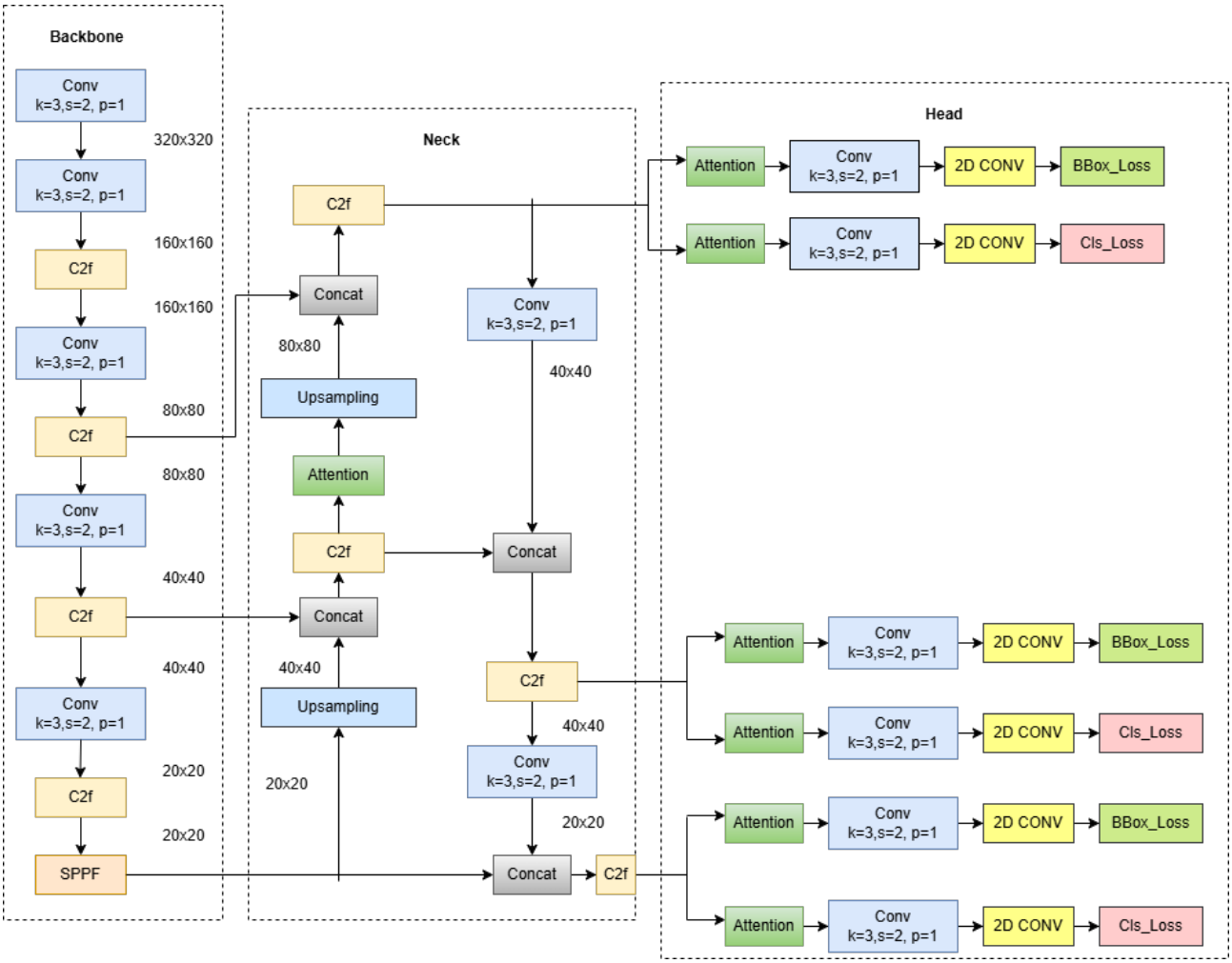

Figure 1: AB-YOLOv8 model architecture

Figure 2: Efficient channel attention

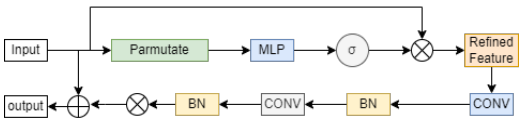

Figure 4: Global attention mechanism

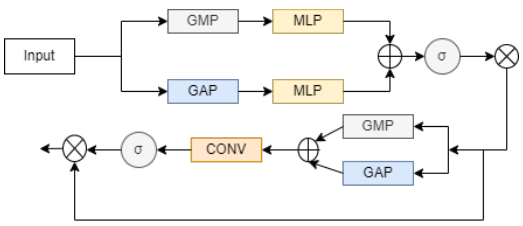

Figure 3: Convolutional block attention module

to acquire and incorporate global information for the specific sub-feature sb_1. Furthermore, a straightforward gating mechanism employing sigmoid functions is utilized to generate a concise function that enables accurate and adaptable selection. Final output of channel attention is as follows:

$$CA = \sigma[fc(GAP(sb_1))] \odot sb_1 \tag{15}$$

In spatial attention first step involves applying Group Normalization (GN) to the sub-feature sb_2 in order to calculate spatial-wise statistics. Afterwards, the output sub-feature sb_2 is improved through fully connected layer fc, as demonstrated in the following equation.

$$SPA = \sigma[fc(GN(sb_2))] \odot sb_2 \tag{16}$$

Algorithm 1 Attention-Integrated YOLOv8 for Object Detection

Require: Input image $I \in \mathbb{R}^{H \times W \times 3}$, Ground truth labels (for training)
Ensure: Predicted bounding boxes and class labels

Backbone Feature Extraction:

$F_1 \leftarrow \text{Conv}(I, k = 3, s = 2, p = 1)$
$F_1 \leftarrow \text{Conv}(F_1, k = 3, s = 2, p = 1)$
$F_1 \leftarrow \text{C2f}(F_1)$
$F_2 \leftarrow \text{Conv}(F_1, k = 3, s = 2, p = 1)$
$F_2 \leftarrow \text{C2f}(F_2)$
$F_3 \leftarrow \text{Conv}(F_2, k = 3, s = 2, p = 1)$
$F_3 \leftarrow \text{C2f}(F_3)$
$F_4 \leftarrow \text{Conv}(F_3, k = 3, s = 2, p = 1)$
$F_4 \leftarrow \text{C2f}(F_4)$
$F_4 \leftarrow \text{SPPF}(F_4)$

Neck with Attention:

$U_1 \leftarrow \text{Upsample}(F_4)$
$A_1 \leftarrow \text{Attention}(U_1)$
$M_1 \leftarrow \text{Concat}(A_1, F_3)$
$M_1 \leftarrow \text{C2f}(M_1)$
$U_2 \leftarrow \text{Upsample}(M_1)$
$A_2 \leftarrow \text{Attention}(U_2)$
$M_2 \leftarrow \text{Concat}(A_2, F_2)$
$M_2 \leftarrow \text{C2f}(M_2)$
$D_1 \leftarrow \text{Conv}(M_2, k = 3, s = 2, p = 1)$
$D_1 \leftarrow \text{Concat}(D_1, M_1)$
$D_1 \leftarrow \text{C2f}(D_1)$
$D_2 \leftarrow \text{Conv}(D_1, k = 3, s = 2, p = 1)$
$D_2 \leftarrow \text{Concat}(D_2, F_4)$
$D_2 \leftarrow \text{C2f}(D_2)$

Detection Head with Attention:

for $H \in \{M_2, D_1, D_2\}$ **do**
  $H' \leftarrow \text{Attention}(H)$
  $H' \leftarrow \text{Conv}(H', k = 3, s = 2, p = 1)$
  $\text{BBox} \leftarrow \text{Conv2D}(H')$ {Bounding box regression}
  $\text{Cls} \leftarrow \text{Conv2D}(H')$ {Classification}
end for
if training **then**
  $\text{Loss}_{\text{bbox}} \leftarrow \text{ComputeLoss}(\text{BBox})$
  $\text{Loss}_{\text{cls}} \leftarrow \text{ComputeLoss}(\text{Cls})$
  $\text{TotalLoss} \leftarrow \text{Loss}_{\text{bbox}} + \text{Loss}_{\text{cls}}$
else
  $\text{Predictions} \leftarrow \text{NMS}(\text{BBox}, \text{Cls})$
  return Predictions
end if

After concatenating these features the final output of Shuffle attention is:

$$SA = CA \oplus SPA \tag{17}$$

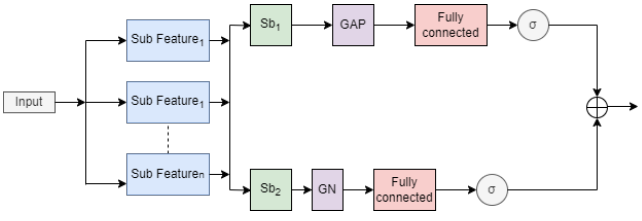

Figure 5: Shuffle attention block

4 Pre-processing and data augmentation of dataset

aquarium Dataset is used for performing experiment with different attention mechanism using YOLOv8. The aquarium Dataset, provided by Roboflow, consists of underwater images captured in controlled environments with limited variation in brightness. This homogeneity in image characteristics poses a challenge for the generalization of the trained model to other underwater images with different lighting conditions. To deal with this issue data augmentation technique is used to improve training dataset. The proposed work used fine-tuning of contrast and brightness so that different lightening levels are present with varying environment during training. The balance of the class of the aquarium dataset is shown in Table 1. To validate the work,

Table 1: Class balance for aquarium dataset

Class	Annotation
Fish	2669
Jellyfish	694
Penguin	516
Shark	354
Puffin	284
Stingray	184
Starfish	116

an experiment was also performed on the Brackish dataset and the class balance is shown in Table 2.

Table 2: Class balance for brackish dataset

Class name	Annotations
Crab	12,348
Smallfish	10,768
Starfish	7,912
Fish	3,352
Jellyfish	637
Shrimp	548

A popular data augmentation method in computer vision, HSV Augmentation modifies an image's Hue (H), Saturation (S), and Value (V) components to replicate different lighting and color conditions is used for augmentation [33]. Because underwater images frequently include uneven lighting and color distortion from light absorption and dispersion in water, this approach works especially well for underwater item detection. HSV augmentation improves the resilience and generalization of models to real-world underwater environments by randomly adjusting hue, saturation, and brightness during training. This helps models learn to distinguish objects under diverse visual appearances.

Since the dataset publisher did not give any predetermined training, validation, and test sets, we randomly divide the aquarium Dataset. More precisely, we assign 70% of the dataset to the training set, 20% to the test set, and 10% to the validation set.

5 Assessment parameters

Evaluation of proposed work is performed based on precision, recall, F1 score, mAP, Params(parameters), inference time, floating point operations (FLOPs) and frames per second (FPS). Precision, recall, F1 score and mAP are calculated based on True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN). Equation (18), eq (19), eq (20) presents formula for precision, recall, F1-score respectively.

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{20}$$

Category-wise Average Precision is calculated as equation (21)

$$AP(C_i) = (1/n) \times (\sum_{i=1}^{n} P_i) \tag{21}$$

where P_i is i^{th} the image of the C_i category and n is number of iterations.

Mean Average Precision is computed as equation (22)

$$mAP = (1/N) \times \sum_{i=1}^{n} AP(C_i) \tag{22}$$

where N is number of classes.

Params are the numbers of parameters involved during training and in this work parameters are calculated using Millions. The number of layers, neurons per layer, architectural complexity, and other variables all affect how many parameters a model has. A larger model size is typically associated with more parameters. In most cases, the larger the model, the better the performance of the model, but it also requires the use of additional data and processing power for training. The connection between computing cost and model complexity must be balanced in real-world applications.

The computational complexity of neural network models is frequently assessed using floating-point operations, which are a metric to evaluate computer or computing system performance. FLOPs show the number of floating-point operations per second of floating-point calculations, offering a vital measure of the model's speed and computational efficiency.

Frames Per Second (FPS) is an important statistic for object recognition, especially for real-time processing applications like interactive gaming, surveillance, and driver less cars. The responsiveness and efficacy of the detection system are strongly impacted by the frame rate (FPS), which shows how many frames a model can process in a second. The inference time of a trained object identification model is the amount of time it takes to process an input image and provide predictions.

6 Experimental setup

The experiment is conducted on PyTorch 2.1.2, utilizing CUDA 11.7 framework. The training is carried out on a single NVIDIA Tesla T4 GPU, which provides a balance between computational power and accessibility. The models are validated after training using the best checkpoint saved during the training process. Validation includes metrics such as precision, recall, mean Average Precision(mAP), and inference speed. Training hyper-parameters are shown in Table 3.

Table 3: Hyper-parameters model training

Parameter	Value
Image Size	640×640
Epochs	100
Optimizer	Stochastic Gradient Descent
Weight Decay	5×10^{-4}
Momentum	0.937
Initial Learning Rate	1×10^{-2}
Batch Size	16
Warmup Epochs	3
Warmup Momentum	0.8
Warmup Bias Learning Rate	0.1

Different software's are used during implementation of AB-YOLOv8 with Python 3.9. Pytorch and Tensor Board used to train the model and for visualization. Numpy and pandas are used for data pre-processing. The base YOLOv8 model is taken from Ultralytics and with it different attention module are used for proposed AB-YOLOv8.

7 Experimental results

In this section, detailed experimental results of the proposed work are reported. We train the AB-YOLOv8 model using training sets with input image size 1024 , to compare the impact of varying input image sizes on the model's performance in the underwater item detection task. Table 4 shows the performance of different attention models combined with YOLOv8. YOLOv8 combined with ResCBAM, GAM, SA and ECA attention block and results are incorporated in this section. Table 4 presents the experimental results with respect to precision, recall, F1 score, and mAP. From Table 4 it is clear that ECA performs better than GAM, ResCBAM, SA when combined with YOLOv8. ECA performs 8% better than YOLOv8 and 6% better than GAM, SA, and ResCBAM.

Table 5 presents another set of AB-YOLOv8 experiment results showing evaluation of different metrics such as parameters, GLOPs, inference time and FPS. It is found from Table 5 in proposed AB-YOLOv8 ECA with YOLOv8 performs better than other techniques. ECA also achieved lowest inference time i.e. 7.7 ms where as other models attains 12.8ms, 8.7ms, 8.0ms inference time. AB-YOLOv8 when based om ResCBAM increased number of parameters almost 10M but when YOLOv8 is based on ECA its not increasing number of parameters as pooling operations are used to optimized the number of parameters. It is also clear that in all the models of AB-YOLOv8 have achieved similar FPS as original YOLOv8 but ECA based YOLOv8 attains 59FPS which is better than SA, GAM and ResCBAM. The aquarium dataset consists of seven categories species like fish, jellyfish, penguin, puffin, shark, starfish, stingray. The Table 6 presents class wise precision achieved by using AB-YOLOv8 models and YOLOv8. Bold results are showing best result achieved during experiments. Out of all AB-YOLOv8 models, ECA based model attains best result in most of the cases. In jellyfish class ResCBAM attains maximum mAP@50.

A small number of images are chosen at random for this paper's evaluation of the attention module's impact on the YOLOv8 model's accuracy in detecting fractures in a real-world marine environment exploration scenario. Figure 10 shows the prediction results of several AB-YOLOv8 models. As an object detection model, the AB-YOLOv8 model is essential to monitor and investigate the marine environment during research. It's crucial to remember, though, that every AB-YOLOv8 model worked flawlessly with tiny, tightly spaced items as well.

The **ablation experiment** shown in Table 7 indicates that the application of different attention mechanisms to the YOLOv8 model can result in considerable gains in mAP@50, recall, and precision; the most striking effect was shown by ECA (Efficient Channel Attention). The precision, recall, and mAP@50 of the base YOLOv8 model are 0.464, 0.305, and 0.328, respectively. The best overall results are obtained when ECA is applied at both the neck and the head (D+H), boosting precision to 0.561, recall to 0.387, and mAP@50 to 0.400. ECA consistently performs better than the other attention mechanisms, especially in terms of recollection and mAP, while SA, GAM, and ResCBAM show only modest gains, especially at the neck. D+H (YOLOv8 with ECA at both the neck and the head) is the best-performing configuration overall, suggesting that using ECA at both phases achieved substantial gains.

Among the evaluated models, ECA and the SA model achieve the highest overall F1-score of 0.29, shown in Figure 9 and Figure 8 respectively, while GAM lags slightly

Table 4: Experiment results of different attention models for aquarium dataset

Model	Precision (P)	Recall (R)	F1 Score	mAP@50	mAP@50-95
YOLOv8	0.464	0.305	0.367	0.328	0.150
YOLOv8+GAM	0.473	0.293	0.363	0.337	0.154
YOLOv8+ResCBAM	0.477	0.301	0.370	0.346	0.152
YOLOv8+SA	0.481	0.321	0.341	0.334	0.151
YOLOv8+ECA	0.561	0.387	0.458	0.400	0.194

Table 5: Experiment results of different attention models for aquarium Dataset

Model	Params(M)	GFLOPs	Inference(ms)	FPS
YOLOv8	43.67	164.37	7.7	60
YOLOv8+GAM	49.89	183.54	12.8	57
YOLOv8+ResCBAM	53.46	196.29	8.7	55
YOLOv8+SA	43.76	165.20	8.0	58
YOLOv8+ECA	43.54	165.34	7.7	59

Table 6: Category wise mAP@50 for different models for aquarium dataset

Category	YOLOv8	YOLOv8+SA	YOLOv8+GAM	YOLOv8+ResCBAM	YOLOv8+ECA
All	0.328	0.336	0.317	0.326	**0.400**
Fish	0.356	0.317	0.289	0.312	**0.378**
Jellyfish	0.614	0.561	0.566	**0.682**	0.656
Penguin	0.227	0.215	0.336	0.245	**0.336**
Puffin	0.114	0.183	0.105	0.182	**0.249**
Shark	0.283	0.281	0.207	0.291	**0.295**
Starfish	0.333	0.410	0.439	0.420	**0.512**
Stingray	0.372	0.152	0.274	0.150	**0.381**

Table 7: Ablation experiment for AB-YOLOv8 using aquarium dataset

Model	Precision	Recall	mAP@50
YOLOv8	0.464	0.305	0.328
A: YOLOv8+ SA at neck of YOLOv8	0.469	0.289	0.330
B: YOLOv8+ GAM at neck of YOLOv8	0.470	0.300	0.338
C: YOLOv8+ResCBAM at neck of YOLOv8	0.469	0.318	0.333
D: YOLOv8+ ECA at neck of YOLOv8	0.521	0.367	0.347
E: YOLOv8+ SA at head of YOLOv8	0.462	0.283	0.334
F: YOLOv8+ GAM at head of YOLOv8	0.476	0.291	0.342
G: YOLOv8+ResCBAM at head of YOLOv8	0.471	0.311	0.332
H: YOLOv8+ ECA at head of YOLOv8	0.541	0.367	0.381
A+E : YOLOv8 +SA	0.473	0.293	0.337
B+F: YOLOv8+ GAM	0.477	0.301	0.346
C+G: YOLOv8+ResCBAM	0.481	0.321	0.334
D+H: YOLOv8+ECA	**0.561**	**0.387**	**0.400**

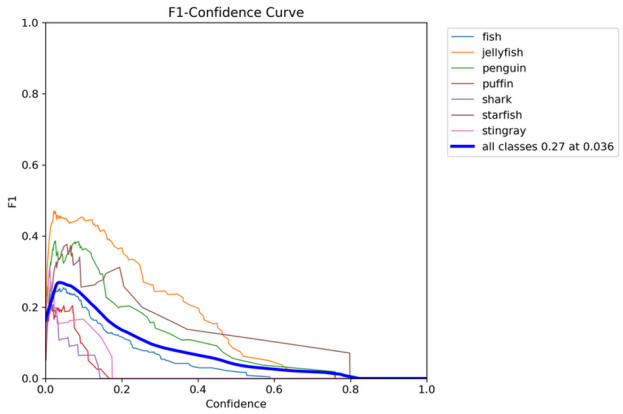

Figure 6: F1-confidence curve for YOLOv8 with GAM attention mechanism using aquarium dataset

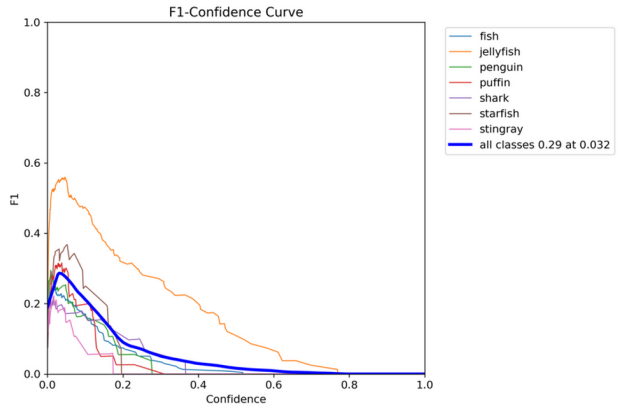

Figure 7: F1-confidence curve for YOLOv8 with CBAM attention mechanism using aquarium dataset

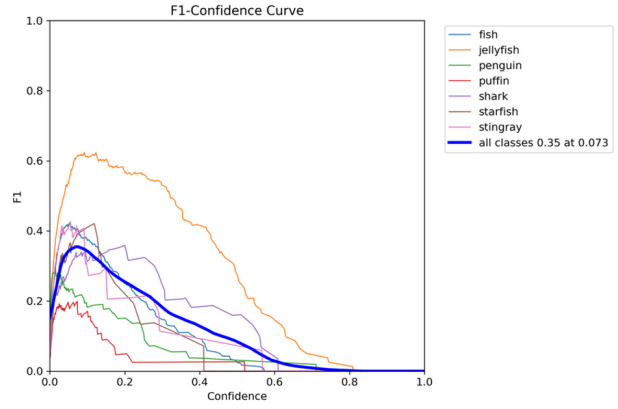

Figure 8: F1-confidence curve for YOLOv8 with SA attention mechanism using aquarium dataset

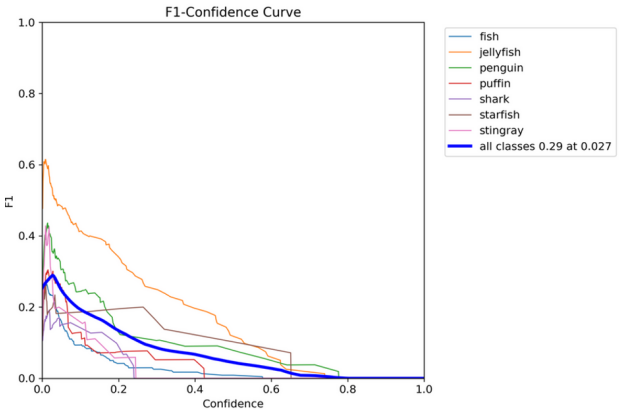

Figure 9: F1-confidence curve for YOLOv8 with ECA attention mechanism using aquarium dataset

at 0.27 as depicted in Figure 6. ECA stands out with the lowest optimal confidence threshold (0.027), offering superior early-stage detection and the smoothest confidence-F1 curve, making it ideal for robust predictions. CBAM and GAM contribute more toward improving per-class balance, with CBAM enhancing spatially diverse classes like puffin and starfish, and GAM excelling in classes with complex contextual dependencies like penguin, as shown in Figure 7. Although GAM does not reach peak F1 performance, it demonstrates the best inter-class balance. Overall, ECA provides the best trade-off between accuracy, stability, and efficiency, making it the most effective enhancement in this setting.

Statistical analysis The proposed work used an ANOVA test for performing statistical analysis. We have performed the same experiment 4 times and calculated mean, standard deviation, standard error and found YOLOv8+ECA performing better than others as shown in Table 8. Also assumed significance level as 5%. Table 9 shows that the p-value is 0.0004, which is much less than 0.05, so the result is significantly good. Moreover, the mean of YOLOv8 + ECA is maximum, so the performance of the ECA attention mechanism is performing well for the aquarium dataset.

8 Discussion

The AB-YOLOv8 compared with SSD and Faster R-CNN and results are shown in Figure 11, Figure 12 and Figure 13. With respect to precision, recall, and mAP, Faster R-CNN gives better results than YOLOv8, but after using the attention mechanism with YOLOv8, it is possible to outperform Faster R-CNN. Although Faster R-CNN is well-known for its high accuracy in object identification tasks, it has a number of drawbacks that limit its usefulness in real-time applications. Due to its two-stage detection architecture, which consists of a Region Proposal Network (RPN) followed by a classification and bounding box regression step, its main disadvantage is its lengthy inference time as shown in Figure 13. Because of this, it is computationally demanding

Figure 10: Sample images of object detection on aquarium dataset, (a-c) object detection by YOLOv8, (d-f) object detection by GAM, (g-i) object detection by ResCBAM, (j-l) object detection by SA, (m-o) object detection by ECA

Table 8: Statistical analysis based on aquarium dataset to calculate mean, standard deviation, standard error

Models	N	Mean	Std. Dev.	Std. Error
YOLOv8	4	43.88	1.781	0.7965
YOLOv8+SA	4	46.3333	1.2111	0.4944
YOLOv8+GAM	4	45.7833	1.3862	0.5659
YOLOv8+ResCBAM	4	46.85	1.1895	0.4856
YOLOv8+ECA	4	52.0167	5.1148	2.0881

Table 9: Statistical analysis based on aquarium dataset to calculate p-value

Source	Degrees of Freedom (DF)	Sum of Squares (SS)	Mean Square (MS)	F-Statistic	P-Value
Between Groups	4	211.1774	52.7943	7.5641	0.0004
Within Groups	24	167.5099	6.9796		
Total	28	378.6872			

and inappropriate for situations requiring quick decisions, such as autonomous driving or real-time video processing. SSD has significant limits even if it provides a decent balance between speed and accuracy. Its inability to detect small objects is a significant disadvantage, mainly due to the fact that it employs numerous feature maps with varying resolutions, which may result in the loss of fine features that are essential for localizing small objects. Furthermore, situations with dense backgrounds or complicated backdrops, where object boundaries are less clear, can be difficult for SSD to handle. Compared to two-stage detectors like Faster R-CNN, its accuracy is typically lower, but it has improved inference time to 25ms, depicted in Figure 13.

Figure 11 and Figure 12 clearly shows that GAM's performance on the AB-YOLOv8 model's on the aquarium dataset is poorer than other attention blocks. The one reason behind the poor performance of GAM is that it has an abundance of pooling layers. The ECA module can be deployed on devices with limited resources because it is computationally efficient and does not require dimensionality reduction or completely connected layers, which makes ECA more efficient and involves fewer parameters. Also, it is visible from Figure 12 ResCBAM and SA performed well with the YOLOv8 model. In order to improve feature representation and performance on a range of tasks, ResCBAM adds both channel and spatial attention, which enables the model to preferentially focus on the most informative channels and spatial regions of the feature maps. Another important issue is the result found on the aquarium Dataset, which consists only of 638 images, including validation, training, and testing images.

Based on how long it typically takes each object detection model to process a single image (measured in milliseconds), the inference time graph comparison shown in Figure 13. As can be seen from the graphic, Faster RCNN has the longest inference time—nearly 80 ms—which suggests that while it may attain competitive accuracy, its computational overhead renders it less appropriate for real-time applications. Even while SSD is faster than Faster RCNN, it still takes about 25 ms, which is more than the YOLO variations. The YOLOv8 and YOLOv8+ECA show noticeably higher inference efficiency than any of the other models that were assessed. Because YOLOv8 has the shortest inference time (around 7 ms), it is ideal for real-time systems.

The proposed work tested on another dataset to validate the performance of the proposed work. Brackish dataset used for the purpose of the experiment is shown in Table 10. It is found that for Brackish datset ECA and ResCBAM achieved 74% mAP@50. ECA does not perform dimensionality reduction so channel-wise features are intact and attains better result. ResCBAM efficiently determines the location and class of the objects by channel attention and spatial attention block. After inclusion of GAM and SA also achieved 2-4% improvement on mAP.

9 Conclusion

After the release of the YOLOv8 model by Ultralytics in 2023, researchers commenced utilizing it for object recognition in underwater images. Although the almost recent generation of the YOLO model, the YOLOv8 model, despite the fact that models performed admirably on the aquarium dataset, were unable to meet the good performance. We added four attention modules GAM, ResCBAM, SA and ECA to the YOLOv8 architecture, respectively, to improve the model's performance in order to overcome this constraint. Furthermore, we integrate ResBlock with CBAM to enhance the overall performance of the model. The proposed work with aquarium dataset achieved 40% maximum mAP@50 for ECA and ECA achieved 7.7 ms inference time with 59 FPS which is better than all other attention blocks. It is also notable that number of parameters not increased for ECA so finally, out of all attention block ECA performed better. Validation of the proposed work is checked on Brackish Dataset also. The results for Brackish dataset that shows for ResCBAM and ECA attention blocked achieved 74% mAP. YOLOv8 with ResCBAM and ECA achieved 8% better mAP than base YOLOv8 model.

Funding and conflicts of interest

Conflict of Interest: The authors did not receive funding and do not have any conflict of interest.

Data availability statements

The article used publicly the available aquarium dataset and link is as follows: aquarium dataset: `https://public.roboflow.com/object-detection/aquarium`
Brackish dataset: `https://public.roboflow.com/object-detection/brackish-underwater`

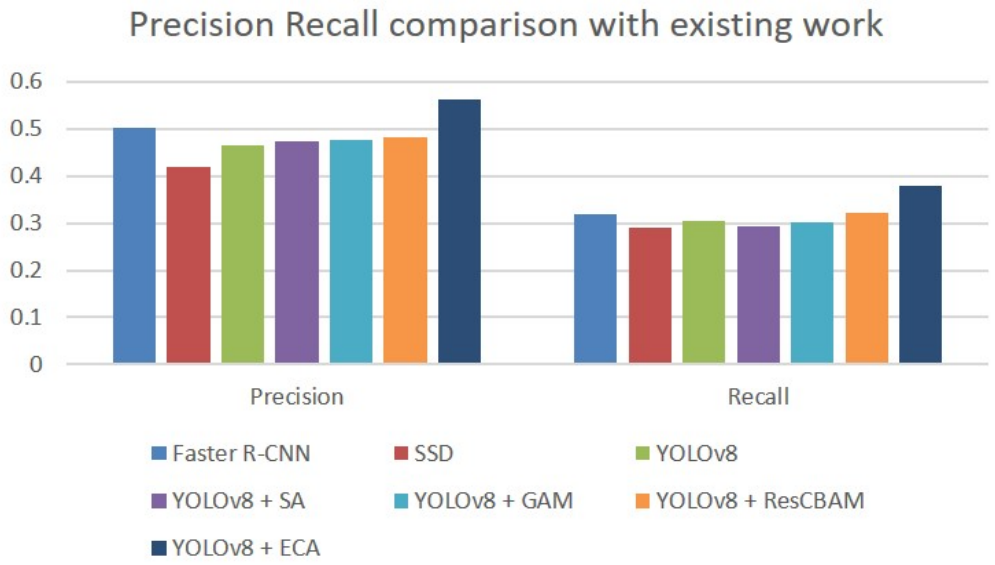

Figure 11: Precision and recall comparison for different models for aquarium dataset

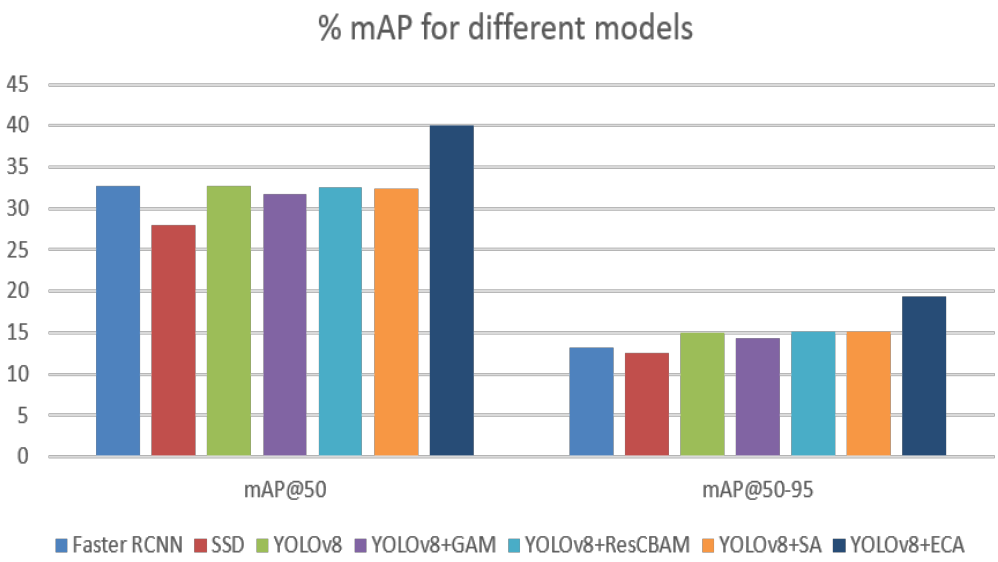

Figure 12: mAP comparison with different models for aquarium dataset

Table 10: Evaluation comparison between different models for Brackish Dataset

Network	Precision	Recall	mAP@50:95
SSD	41.19	35.02	30.71
Faster-RCNN	69.23	65.02	61.45
YOLOv8	92.29	91.04	68.21
YOLOv8+GAM	91.10	92.8	69.44
YOLOv8+SA	92.49	90.28	72.30
YOLOv8+ResCBAM	**94.80**	**91.90**	**74.20**
YOLOv8+ECA	**95.01**	**90.90**	**74.31**

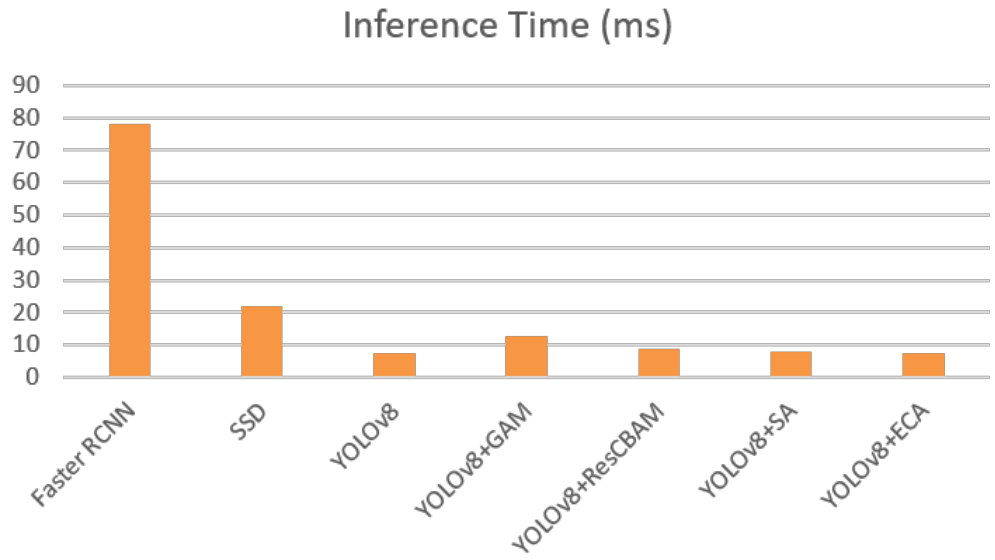

Figure 13: Inference Time comparison with different models

References

[1] Pratima Sarkar, Sourav De, and Sandeep Gurung. A survey on underwater object detection. In *Intelligence Enabled Research: DoSIER 2021*, pages 91–104. Springer, 2022 https://doi.org/10.1007/978-981-19-0489-9.

[2] Huimin Lu, Yujie Li, Yudong Zhang, Min Chen, Seiichi Serikawa, and Hyoungseop Kim. Underwater optical image processing: a comprehensive review. *Mobile networks and applications*, 22:1204–1211, 2017 https://doi.org/10.1007/s11036-017-0863-4.

[3] Pratima Sarkar, Sandeep Gurung, and Sourav De. Underwater image segmentation using fuzzy-based contrast improvement and partition-based thresholding technique. In *Evolution in Computational Intelligence: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021)*, pages 473–482. Springer, 2022 https://doi.org/10.1007/978-981-16-6616-2.

[4] Dario Lodi Rizzini, Fabjan Kallasi, Fabio Oleari, and Stefano Caselli. Investigation of vision-based underwater object detection with multiple datasets. *International Journal of Advanced Robotic Systems*, 12(6):77, 2015 https://doi.org/10.5772/60526.

[5] Pratima Sarkar, Sourav De, Sandeep Gurung, and Prasenjit Dey. Uice-mirnet guided image enhancement for underwater object detection. *Scientific Reports*, 14(1):22448, 2024 https://doi.org/10.1038/s41598-024-73243-9.

[6] Yan Zhai. River ship monitoring based on improved deep-sort algorithm. *Informatica*, 48(9), 2024 https://doi.org/10.31449/inf.v48i9.5886.

[7] Md Moniruzzaman, Syed Mohammed Shamsul Islam, Mohammed Bennamoun, and Paul Lavery. Deep learning on underwater marine object detection: A survey. In *Advanced Concepts for Intelligent Vision Systems: 18th International Conference, ACIVS 2017, Antwerp, Belgium, September 18-21, 2017, Proceedings 18*, pages 150–160. Springer, 2017 https://doi.org/10.1007/978-3-319-70353-4.

[8] Pratima Sarkar, Sourav De, and Sandeep Gurung. Fish detection from underwater images using yolo and its challenges. In *Doctoral Symposium on intelligence enabled research*, pages 149–159. Springer, 2022 https://doi.org/10.1007/978-981-99-1472-2.

[9] Long Chen, Zhihua Liu, Lei Tong, Zheheng Jiang, Shengke Wang, Junyu Dong, and Huiyu Zhou. Underwater object detection using invert multi-class adaboost with deep learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020 https://doi.org/10.1109/IJCNN48605.2020.9207506.

[10] Jian Zhang, Jinshuai Zhang, Kexin Zhou, Yonghui Zhang, Hongda Chen, and Xinyue Yan. An improved yolov5-based underwater object-detection framework. *Sensors*, 23(7):3693, 2023 https://doi.org/10.3390/s23073693.

[11] Pratick Gupta, Pratima Sarkar, Bijoyeta Roy, and Shivam Kumar. Fish classification using cnn and logistic regression from underwater images. In

International Conference on Advanced Computational and Communication Paradigms, pages 415–424. Springer, 2023 https://doi.org/10.1007/978-981-99-4284-8.

[12] Wang Zhiqiang and Liu Jun. A review of object detection based on convolutional neural network. In *2017 36th Chinese control conference (CCC)*, pages 11104–11109. IEEE, 2017 https://doi.org/10.23919/ChiCC.2017.8029130.

[13] Mupparaju Sohan, Thotakura Sai Ram, Rami Reddy, and Ch Venkata. A review on yolov8 and its advancements. In *International Conference on Data Intelligence and Cognitive Informatics*, pages 529–545. Springer, 2024 https://doi.org/10.1007/978-981-99-7962-2.

[14] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11534–11542, 2020 https://arxiv.org/pdf/1910.03151v3.

[15] Qing-Long Zhang and Yu-Bin Yang. Sa-net: Shuffle attention for deep convolutional neural networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2235–2239. IEEE, 2021 Doi: 10.1109/ICASSP39728.2021.9414568/.

[16] Yichao Liu, Zongru Shao, and Nico Hoffmann. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv preprint arXiv:2112.05561*, 2021 https://arxiv.org/abs/2112.05561.

[17] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018 https://doi.org/10.48550/arXiv.1807.06521.

[18] Chun-Tse Chien, Rui-Yang Ju, Kuang-Yi Chou, Chien-Sheng Lin, and Jen-Shiun Chiang. Yolov8-am: Yolov8 with attention mechanisms for pediatric wrist fracture detection. *arXiv preprint arXiv:2402.09329*, 2, 2024 https://doi.org/10.1109/ACCESS.2025.3549839.

[19] Arindam Chaudhuri. Hierarchical modified fast r-cnn for object detection. *Informatica*, 45(7), 2021 https://doi.org/10.31449/inf.v45i7.3732.

[20] Weiwen Chen, Tingting Zhuang, Yuanfang Zhang, Teng Mei, and Xiaoyu Tang. Yolo-uod: An underwater small object detector via improved efficient layer aggregation network. *IET Image Processing*, 2024 https://doi.org/10.1049/ipr2.13112.

[21] Shijia Zhao, Jiachun Zheng, Shidan Sun, and Lei Zhang. An improved yolo algorithm for fast and accurate underwater object detection. *Symmetry*, 14(8):1669, 2022 https://doi.org/10.3390/s23073693.

[22] Kun Liu, Lei Peng, and Shanran Tang. Underwater object detection using tc-yolo with attention mechanisms. *Sensors*, 23(5):2567, 2023 https://doi.org/10.3390/s23052567.

[23] Ali M Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38:35–44, 2004 https://doi.org/10.1023/B:VLSI.0000028532.53893.82.

[24] Risheng Liu, Xin Fan, Ming Zhu, Minjun Hou, and Zhongxuan Luo. Real-world underwater enhancement: challenges, benchmarks, and solutions. *arXiv preprint arXiv:1901.05320*, 2019 https://doi.org/10.48550/arXiv.1901.05320.

[25] Xin Shen, Xudong Sun, Huibing Wang, and Xianping Fu. Multi-dimensional, multi-functional and multi-level attention in yolo for underwater object detection. *Neural computing and applications*, 35(27):19935–19960, 2023 https://doi.org/10.1007/s00521-023-08781-w.

[26] Jie Chen and Meng Joo Er. Dynamic yolo for small underwater object detection. *Artificial Intelligence Review*, 57(7):1–23, 2024 https://doi.org/10.1007/s10462-024-10788-1.

[27] Yuyi Yang, Liang Chen, Jian Zhang, Lingchun Long, and Zhenfei Wang. Ugc-yolo: underwater environment object detection based on yolo with a global context block. *Journal of Ocean University of China*, 22(3):665–674, 2023 https://doi.org/10.1007/s11802-023-5296-z.

[28] Xiaohan Wang, Xiaoyue Jiang, Zhaoqiang Xia, and Xiaoyi Feng. Underwater object detection based on enhanced yolo. In *2022 International Conference on Image Processing and Media Computing (ICIPMC)*, pages 17–21. IEEE, 2022 https://doi.org/10.1109/ICIPMC55686.2022.00012.

[29] AF Ayob, K Khairuddin, YM Mustafah, AR Salisa, and K Kadir. Analysis of pruned neural networks (mobilenetv2-yolo v2) for underwater object detection. In *Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS'19*, pages 87–98. Springer, 2021 https://doi.org/10.1007/978-981-15-5281-6.

[30] Yue Li, Xueting Zhang, and Zhangyi Shen. Yolo-submarine cable: an improved yolo-v3 network for

object detection on submarine cable images. *Journal of Marine Science and Engineering*, 10(8):1143, 2022 https://doi.org/10.3390/jmse10081143.

[31] Ali Farhadi and Joseph Redmon. Yolov3: An incremental improvement. In *Computer vision and pattern recognition*, volume 1804, pages 1–6. Springer Berlin/Heidelberg, Germany, 2018 https://doi.org/10.48550/arXiv.1804.02767.

[32] Pratima Sarkar, Sourav De, and Sandeep Gurung. U-yolov3: A model focused on underwater object detection. *Informatica*, 49(6), 2025 https://doi.org/10.31449/inf.v49i6.6642.

[33] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020 https://doi.org/10.48550/arXiv.2004.10934.