



# A novel finetuned YOLOv8 model for real-time underwater trash detection

Chhaya Gupta<sup>1</sup> · Nasib Singh Gill<sup>1</sup> · Preeti Gulia<sup>1</sup> · Sangeeta Yadav<sup>1</sup> · Jyotir Moy Chatterjee<sup>2</sup>

Received: 17 July 2023 / Accepted: 14 February 2024 / Published online: 8 March 2024  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

## Abstract

When recognizing underwater images, problems, including poor image quality and complicated backdrops, are significant. The main problem of underwater images is the blurriness and invisibility of objects present in an image. This study presents a unique object identification design built on a YOLOv8 (You Only Look Once) framework upgraded to address these problems and further improve the models' accuracy. The study also helps in identifying underwater trash. The model is a two-phase detector model. The first phase has an Underwater Image Enhancer (UIE) data augmentation technique that works with Laplacian pyramids and gamma correctness methods to enhance the underwater images. The second phase, the proposed refined, innovative YOLOv8 model for classification purposes, takes the output from the first stage as its input. The YOLOv8 model's existing feature extractor is replaced in this study with a new feature extractor technique, HEFA, that yields superior results and better detection accuracy. The introduction of the UIE and HEFA feature extractor method represents the significant novelty of this paper. The proposed model is pruned simultaneously to eliminate unnecessary parameters and further condense the model. Pruning causes the model's accuracy to decline. Thus, the transfer learning procedure is employed to raise it. The trials' findings show that the technique can detect objects with an accuracy of 98.5% and a mAP@50 of 98.1% and that its real-time detection speed on the GPU is double that of the YOLOv8m model's baseline performance.

**Keywords** Feature extractor · Pruning · Transfer learning · Underwater image enhancer · Underwater trash · YOLOv8

## 1 Introduction

Trash detection in underwater settings has gained attention due to the ongoing advancements in computer vision and the commercialization of marine resources. It is now employed in a variety of sectors, including marine research

[1], underwater robotics [2], and underwater detection [3]. Long-term sewage in the water breaks down into minute particles that are difficult to perceive with the naked eye and harm human well-being and the ecosystem. Only a few researchers have thought about the enormous problem of improving trash identification accuracy in complicated and shifting underwater environments. Object detection (OD) identifies the region of an object in the input image and the type of object present in the frame. Techniques based on deep learning (DL) have replaced conventional object recognition approaches because of their weak robustness when features are manually chosen. Today, users can observe the detection method based on DL everywhere, such as the real-time vehicle monitoring in the intelligent transportation system [4], the fall detection system for older people with smart cameras [5], and a real-time detection model for the visually impaired [6]. The YOLO algorithm [7], the Single Shot MultiBox Detector (SSD) algorithm [8], R-CNN [9], and RetinaNet [10] techniques, which are traditional OD methods based on DL, demonstrate the high detection accuracy of the DL method.

---

✉ Jyotir Moy Chatterjee  
jyotirmoy.chatterjee.cse@gmail.com

Chhaya Gupta  
chhaya.rs.dcsa@mdurohtak.ac.in

Nasib Singh Gill  
nasib.gill@mdurohtak.ac.in

Preeti Gulia  
preeti@mdurohtak.ac.in

Sangeeta Yadav  
sangeeta.rs.dcsa@mdurohtak.ac.in

<sup>1</sup> Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, Haryana, India

<sup>2</sup> Department of CSE, Graphic Era University, Dehradun, India

Traditional data augmentation methods are ineffective for underwater environments because of the poor contrast and glaring colour differences in the datasets currently available. Based on a significant amount of research studies, image-enhancement methods can enhance the overall quality of images by using various versions of the original images that result from these operations but are insufficient in improving model performance. Consequently, using image-restoration techniques for data enhancement, the researchers in [11] also noted that including data-augmentation activities can increase a model's detection accuracy.

The complexity of underwater ecosystems makes it difficult to detect and mitigate underwater trash. The current trash detection techniques have to face different challenges like limited visibility factors in the underwater environment, diverse shapes of underwater trash, noise due to marine life and human activities, less life of sensors because of salty sea water, processing large volumes of underwater data is an open challenge and many more. Research done till now has achieved great results in underwater trash detection but still, many open threads need to be worked on.

This work aims to achieve high-precision detection using a newly optimized YOLOv8 object-detection model. The Trash\_ICRA19 dataset is processed for data enhancement using the proposed UIE image-enhancing approach, and the augmented images are then used as the input to the finetuned network. The model is further enhanced with a HEFA feature extractor for improved detection accuracy.

The main contribution of this work is as follows:

- A novel 11-step data augmentation technique, Underwater Image Enhancer (UIE), has been proposed for enhancing underwater images. The method employs the Laplacian pyramid and Gaussian corrections to enhance the underwater images.
- A novel feature extractor, HEFA (Hybrid Extraction of Features Algorithm), has been proposed that replaces the feature extractor of YOLOv8, improving the model's overall performance. This algorithm helps in extracting appropriate features from the enhanced images.
- The baseline YOLOv8m model has been used for the experiment. The proposed feature extractor replaces its extractor and hence contributes to better performance. The model is also pruned with a pruning algorithm which was proposed by the authors in a previous work with YOLOv6 [6]. While pruning, the model's accuracy deprecates; hence, a transfer learning algorithm has been used to improve the proposed model's deprecated accuracy.

The rest of the document is structured as follows: Sect. 2 presents the literature review. Section 3 provides a comprehensive explanation of the approach, while Sect. 4 presents

the experimental results and related discussions. Section 5 concludes the report and discusses the next research.

## 2 Related literature survey

This section discusses the related literature survey in the state-of-the-art domain.

### 2.1 Underwater object recognition

OD, which has various uses in various scenarios, is one of computer vision's most elementary and challenging problems [18]. Kun et al. [12] profounded a TC-YOLO model based on the YOLOv5 OD model. The model was trained on the RUIE2020 dataset. They also used an image enhancement technique, CLAHE, combined with the YOLOv5 model. The research is done to detect underwater marine life detection. Xu et al. [3] presented a systematic, in-depth review related to underwater OD. The authors discussed the challenges, future work, and OD applications in the aquatic field. Hao et al. [13] profounded a reinforcement learning approach for visual improvement in underwater sceneries. Pinhao et al. [14] proposed a two-stage underwater detector, namely, boosting RCNN having three components. The experiment is performed on the UTDAC2020 dataset. Ricky et al. [15] proposed a Siamese region proposal network (SiamRPN) for underwater OD on the VOT2018 dataset. Yu et al. [16] profounded a novel multiple-attentional path aggregation network APAN for better underwater detection. Young et al. [17] proposed a model based on Monocular Depth Estimation (MDE) and the YOLOv3 OD model for underwater OD. The authors used MiDAS v3 for calculating the distance between different reference points. Wu et al. [18] proposed an improved YOLOv5s-based model embedded with MobileNetv3 and Convolutional Block Attention Module for underwater garbage detection. The method achieved an accuracy of 97.5%. Although this method's accuracy is 97%, it cannot match the real-time requirements for underwater object recognition since it takes a long time to respond and requires a lot of storage. This research is also compared in this study with the proposed model and results show that the proposed model achieved better accuracy. Enhancing the underwater object identification models' computational efficiency and storage capacity is essential for their practical implementation, particularly in settings with limited resources. The computational efficiency and storage can be enhanced by different techniques such as quantization, pruning, lightweight architectures, transfer learning, batch normalization, and many more. Quantization helps in reducing the precision of model weights and activation functions. Pruning helps in removing redundant parameters from model and thus helps in reducing the size of the model.

Lightweight architectures help in improving efficiency by maintaining good performance. Transfer learning helps in reducing the amount of required training data and thus helps in increasing the accuracy. Batch normalization helps to stabilize training data. These techniques can be combined together based on specific requirements. In this work, pruning is used to improve the model's efficacy. Zhang et al. [19] suggested a unique object identification framework embedded with an image enhancement module for underwater target detection. But the detection model used in this study is quite outdated, and the image enhancer module needs to be trained independently which causes an excessive calculation.

## 2.2 Review of YOLO and YOLOv8

YOLO algorithms have been reviewed in this section. Chin et al. [20] employed YOLOv3 for underwater OD, and the network achieved good results. Jun et al. [21] proposed MCS-YOLOv4 for detecting small objects. Zhang et al. [22] suggested an enhanced YOLOv5 setup for underwater identification of things. The framework uses the locally adaptive contrast enhancement (MLLE) technique and minimal colour loss to improve underwater photos. Gupta et al. [6] proposed a novel finetuned YOLOv6 framework for real-time OD. The study proposed algorithms for pruning as well as transfer learning which are utilized for finetuning the baseline YOLOv6 framework. Liu et al. [23] proposed an improved YOLOv7 for underwater OD. The model used a Global Attention Mechanism along with the K-means + + algorithm.

YOLOv8 is the most recent model in the YOLO family. The enhancements make YOLOv8 one of the best-performing OD algorithms available, enabling faster and more accurate object recognition. Lou et al. [24] profounded a DC-YOLOv8 model for identifying small-size objects. The study also proposes a down-sampling method; the overall network is 0.5% more accurate than the baseline YOLOv8 model. Li et al. [25] introduced a Bi-PAN-FPN in the neck of the YOLOv8s model. The algorithm is evaluated on the VisDrone2019 dataset. Hwa-Kim et al. [26] modified the existing YOLOv8m model for fast drone detection. The model achieved an FPS of 45.7. The study revealed that the YOLOv8 model achieved the highest accuracy when compared to others. Table 1 presents a summary of the YOLO-related work with research gaps and results.

Although much work has been done in underwater target detection, there are still open challenges, such as illumination conditions, contrast, different colours in images, haziness, and others. Some of the limitations that need to be addressed are:

- Mostly the work is carried out on pre-trained versions of OD models without any modification or improvement in

the available OD algorithm. Innovating and modifying the existing models is a tedious task and requires much effort.

- The study so far revealed that YOLOv8 has outperformed all the existing OD models regarding accuracy and mAP.
- The dataset consists of underwater images; most images are Hazy, blurred, and have dark backgrounds, and the research done till now is done on them only without enhancing the images.

A fine-tuned YOLOv8 with a transfer learning model is proposed to overcome the above-stated limitations. The model comprises an underwater image enhancer (UIE) and a novel feature extractor module that further improves the model. In this paper:

- The YOLOv8m model is used as a baseline model. The baseline YOLOv8m is pruned to get a lightweight network, further improved by the transfer learning algorithm.
- To improve the efficiency of images in the dataset, a novel UIE algorithm is proposed. UIE is an eleven-step procedure that works on the input images' RGB channel using Laplacian pyramids and Gamma correctors.
- A new feature extractor HEFA has been proposed for further enhancing the computational speed of the overall model and increasing the detection speed as well as efficiency of the proposed model.

## 3 Methodology

### 3.1 Overall architecture of the proposed model

An enhanced and refined YOLOv8 OD model is suggested in this section. Figure 1 shows the main layout of the suggested model. The suggested model is a two-phase detector model, with the underwater image enhancer algorithm (UIE) acting as the first module of the first phase. This experiment processes underwater images before using them as inputs for the object-detection model. In the second phase, the improved OD framework based on finetuned YOLOv8 is used to classify the augmented photos.

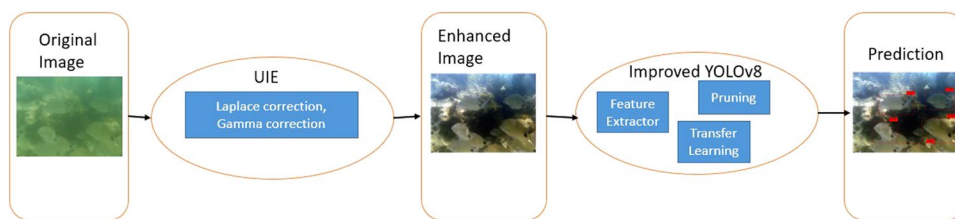
### 3.2 Data collection

This study uses the Trash\_ICRA19 dataset [36] to train the model. The videos have been used for collecting data for underwater OD. A video sampling algorithm has been proposed for sampling the frames of the videos for forming the dataset. This algorithm receives the number of frames  $F$  as input. A sample rate is determined for each video group using

**Table 1** Comparative analysis

Author	Dataset	Model	Results	Research gap
Chia Chin et al. [20]	ImageNet Dataset	YOLOv3	The model provided good results	The model is not suitable for real-time applications and is not compared with other SOTA models
Shu Jun et al. [21]	UCAS-AOD dataset	MCS-YOLOv4 with soft CIoU loss function	The model achieved an average precision of 84%	The model in its current state is not suitable for real-time applications and is a time-consuming model
Jian Zhang et al. [22]	URPC2019 and URPC2020 dataset	YOLOv5 with RTMDet and MHSA mechanism	The model achieved a mAP of 79%	The model can be further improved in terms of mAP by pruning and transfer learning
Chhaya Gupta et al. [6]	MS-COCO dataset	A fine-tuned YOLOv6 model with pruning and transfer learning techniques	The model achieved good results in terms of mAP	The model is used for real-time applications but the model is not able to overcome the challenge of textured background
Kaiyue Liu et al. [23]	URPC, Brackish dataset	YOLOv7 model with Global attention mechanism (GAM) and K-means + + clustering	The model achieved a mAP of 89%	The model achieved good results when compared with other models, but the computing speed of the model is not significant and the availability of high-quality underwater datasets and images remains a major challenge
Haitong Lou et al. [27]	PASCAL VOC2007 dataset	DC-YOLOv8 model	The model achieved a mAP of 83%	The model is not tested on real-time applications and can be further improved with pruning and transfer learning techniques
Yiting Li et al. [25]	VisDrone2019 dataset	YOLOv8s model with BiPAN-FPN	The model achieved a mAP of 91%	The model is not tested on real-time applications and the model does not achieve better results with small objects
Jun-Hwa Kim et al. [26]	Custom dataset	YOLOv8m model with multi-scale image fusion and P2 layer	The model achieved good results in terms of mAP	The model achieved good results when compared with other models but the computing speed of the model is not significant

**Fig. 1** Complete architecture of the proposed model



the data on the overall number of videos in every group and the frames per second of each.

**Algorithm 1** Video Sampling Algorithm

---

```

Input:  $F, V, fps$ 
Output: Selected frames from each video
for  $c \in 1, 2, 3, \dots, F$  do
    1. The sample rate is evaluated for each  $c$ :
    2.  $S_c = \frac{F}{\sum_{i=1}^F F_v * fps^i}$ 
    3. Select frames from each video at the evaluated  $S$ 
    4. End for
    5. Return frames
    
```

---

Where,

$F$  = Number of frames.

$V$  = Number of video categories, in this case, it is 1 as all videos are underwater videos.

$Fps$  = Frames per second.

$C$  = Category of each video.

$S_c$  = Sample rate for each category  $c$ .

**3.3 Underwater image enhancer (UIE)**

Underwater light propagation is hampered by scattering and absorption, much like light traveling through the air. However, there is a tremendous amount of absorption and scattering. When light is severely degraded, it is tough for imaging devices to collect data from a target underwater location. In contrast to air, water is opaque to all other wavelengths and only accessible to the visible portion of the electromagnetic spectrum. It is incredible how quickly light energy degrades in water. By 150 m in depth, barely one percent of light energy remains in the middle oceans' apparent waters. As a result, the object is more challenging to see beyond a 20-m distance, and in muddy coastal waters, the visibility drops below the 5-m threshold. In addition, no sunlight penetrates deeper than 1 km into the ocean. As a result, the quantity of light inside the water's surface is continuously lesser than the intensity of light above it. The UIE module is a combination of three parts as depicted in Eq. (1):

$$A_T = A_d + A_f + A_b, \tag{1}$$

where  $A_T$  = total light that falls on the image sensor and helps in forming the image,  $A_d$  = direct transmission of energy

on the surface of an object  $A_0$ ,  $A_f$  = forward scattered light which is reflected from the object but suffers deflections before entering the sensor,  $A_b$  = backscattered light that falls on the sensor but contains no information about the object.

The reflected component of light can be described mathematically as:

$$A_f = A_0 e^{-c_\lambda r}, \tag{2}$$

where  $r$  = space length between sensor and object,  $c_\lambda$  = attenuation coefficient constant that depends on wavelength  $\lambda$ .

Back-scattering light can be mathematically described as:

$$A_b = B_\infty (1 - e^{-c_\lambda r}), \tag{3}$$

where  $B_\infty$  = back-scattering light signal.

Data augmentation is a technique used in machine learning and deep learning to artificially increase the size of a training dataset by applying various transformations to the existing data. The common transformations applied during data augmentation are rotation, flipping, zooming, translation, shearing, brightness and contrast adjustment, colour jittering, gaussian noise, and several more. In this experiment, we have proposed an 11-step transformation to the underwater image data named Underwater Image Enhancer (UIE) as the method is a combination of different transformations like red channelling, white balancing, Laplacian transformations, sharpening, gamma corrections, saliency image, and normalizing, hence we used the name data augmentation with the proposed UIE technique.

The proposed UIE algorithm is a combination of eleven different transformation steps that are combined to form the desired module. The techniques that are fused are red channel evaluation, Red-channel compensation, white balance evaluation, gamma correction, sharpening, Laplacian contrast weights evaluated, saliency weights evaluated, saturation weights evaluated and normalized weights are evaluated and then finally the enhanced image is produced as output. The original image is passed as an input to the module and all the above-mentioned techniques are evaluated and they help in providing an enhanced image as shown in Fig. 2.

Correcting colours underwater is always a crucial task, hence firstly the white balancing approach is used on the original image. This process removes undesirable colour casts brought on by various illuminates to improve the image's appearance. White balancing suffers noticeably in



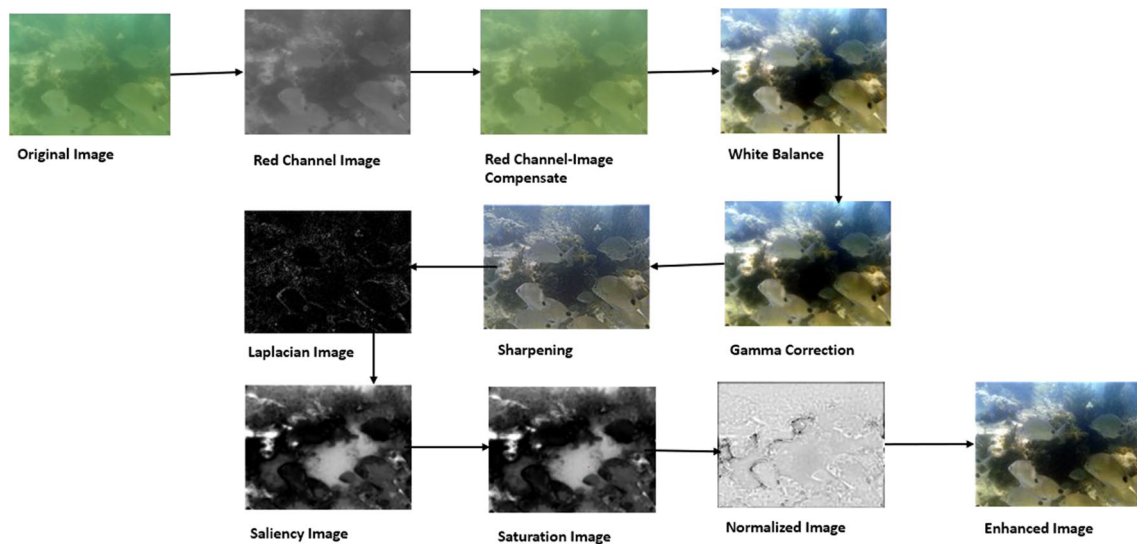


Fig. 2 Complete steps of the UIE model describing the eleven steps of the proposed UIE

water deeper than 30 feet. Gamma correction is applied to the variant of the white-balanced image to obtain the initial input. Gamma correction, which aims to enhance the overall contrast of underwater images. This adjustment increases the contrast between darker and lighter areas at the expense of losing details in the under or overexposed regions. The second input is created that corresponds to a sharpened version of the white-balanced image to make up for this loss. Blending a blurry or unsharp portion of the image with the one that is being sharpened (here, using a Gaussian filter) complies with the unsharp masking concept. Sharpened image  $D$ 's formula is:

$$D = I_g + \alpha(I_g - G * I_g), \tag{4}$$

where  $I_g$  = image to be sharpened (the white balanced image),  $G * I_g$  = Gaussian filter for image  $I_g$ ,  $\alpha$  = parameter.

However, selecting  $\alpha$  is a trivial task. If  $\alpha$  is small then it is not able to sharpen the image  $I$ , and if it is too large then the problem of over-saturation occurs. Hence to overcome this problem, the sharpened image  $D$  is redefined as:

$$D = (I_g + N\{I_g - G * I_g\})/2, \tag{5}$$

where  $N\{I_g - G * I_g\}$  = linear normalization operator or histogram stretching operator.

To ensure that the set of corrected pixel values includes the full range of conceivable dynamic range, this normalization method shifts and scales every single pixel intensity of a picture using a specified shifting and scaling factor. The enhancer also uses some weights like Laplacian, saliency, saturation, and normalized weights. To determine the overall contrast, the Laplacian weights (Lw) determine

the actual magnitude of a Laplacian filter applied to each input luminance channel. However, this weight is insufficient to recover the contrast for the underwater dehazing challenge, mostly because of its limited ability to differentiate between ramp and flat sections. Therefore, an extra and supplemental contrast assessment metric to address this issue has been applied. Saliency weight ( $S_w$ ) highlights the salient items that lose prominence in the underwater image. However, the saliency map often prioritizes the most noticeable locations (those that have elevated brightness values). This restriction is overcome by creating a second weight map based on the observation that saturation reduces in the highlighted regions.

Saturation weight ( $Sat_w$ ), which favours highly saturated regions, helps the merger algorithm to adjust the chromatic information. This weight map is simply calculated as the difference between the luminance  $L_i$  of the  $i$ th input and the  $R_i$ ,  $G_i$ , and  $B_i$  colour channels (for each pixel location in the input  $I_i$ ).

$$\sqrt{\frac{1}{3} [(R_i - L_i)^2 + (G_i - L_i)^2 + (B_i - L_i)^2]}. \tag{6}$$

The above three weight maps are combined into one weight map ( $C_w$ ) as follows for each input. The  $L_w$ ,  $S_w$ , and  $Sat_w$  weight maps are added up to create an aggregated weight map  $C_w$  for each input  $i$ . The weights of each pixel in each map are then divided by the total weights for that same pixel across all maps to normalize the 'i' aggregated maps on a pixel-by-pixel basis. Formally, for each input, the normalized weight maps  $\overline{C_w}$  are computed. As the normalized weights have been calculated, the enhanced image ( $E_x$ ) may be calculated at every pixel  $x$  as shown in Eq. (7):

$$E_x = \sum_{i=1}^i \overline{C_w(x)} I_i(x). \tag{7}$$

While performing this approach, undefined halos occurred in the images, to overcome this situation, multi-scale fusion is used, which is described in Eq. (8) as:

$$E_{a(x)} = \sum_{i=1}^i G_a \{ \overline{C_w(x)} L_a \{ I_i(x) \} \}, \tag{8}$$

where  $a$  = pyramid levels,  $i$  = number of input images,  $G_a$  = Gaussian pyramid at  $a$ th level,  $L_a$  = Laplacian pyramid at  $a$ th level.

The overall process of UIE is shown in Fig. 3 and the outputs at each level of this module are shown in Fig. 4.

The corresponding quantitative evaluation is shown in Table 2 using the two most recent metrics, UCIQE and UIQM. The review of underwater images is the focus of the UCIQE and UIQM metrics. They are used to estimate the blurring impact caused by scattering and the color degradation caused by light absorption in water. Colour and sharpness measurements are combined linearly in UCIQE and UIQM, with

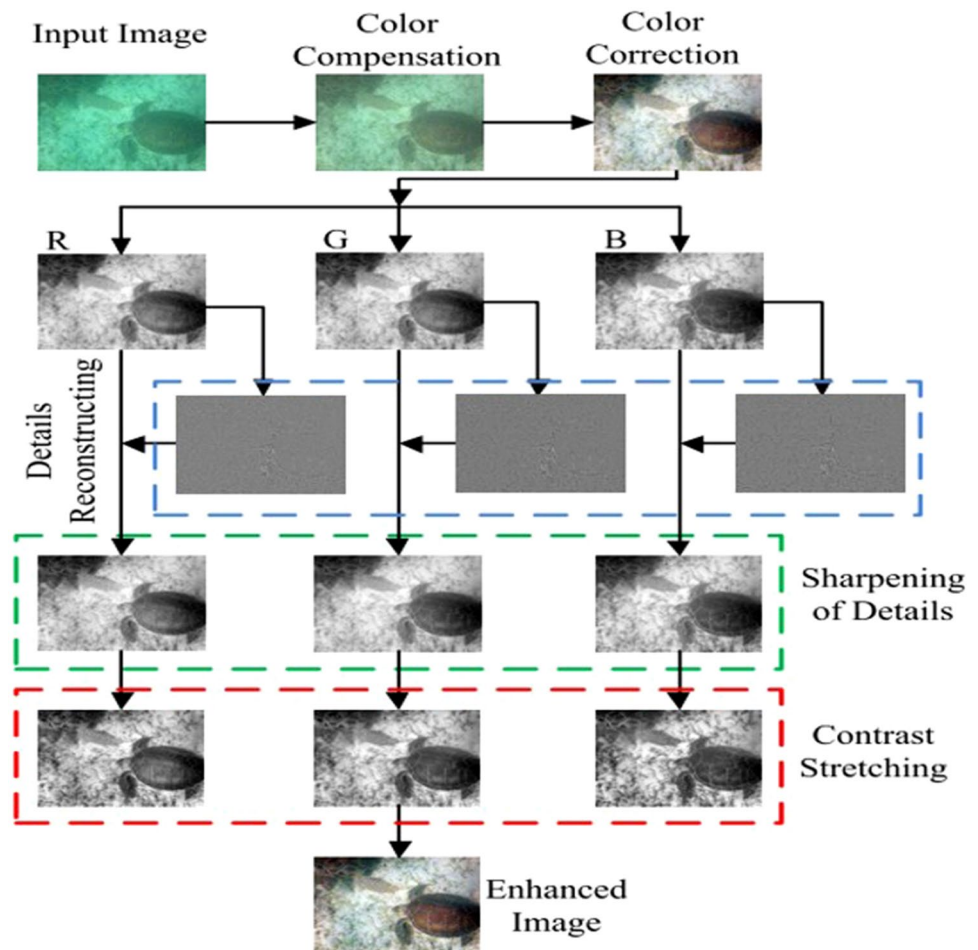
coefficients derived from data on subjective judgment. While the three key underwater picture quality criteria of colourfulness, sharpness, and contrast are addressed by the UIQM metric, the UCIQE measure was created expressly to evaluate the non-uniform colour cast, blurring, and low contrast that characterize underwater photos. While UIQM is based on the human visual system and considers contrast loss, UCIQE solely assesses an image's quality based on the colour distortion brought on by light attenuation.

In Fig. 4, the lighting effects, and scattering are reduced by using different 11 steps of UIE proposed in this study. In Table 2, the UCIQE and UIQM values for some objects present in the images are evaluated. These values are evaluated by the following equations.

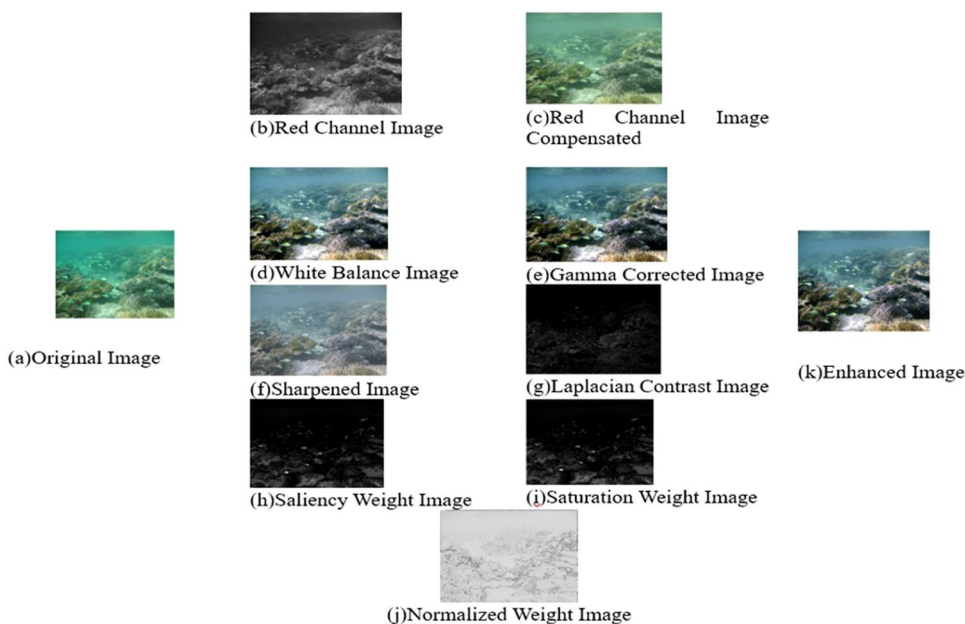
Let  $L_i = [B_i, x_i, y_i]$  value of any pixel in a given space, where,  $B_i, x_i, y_i$  are intensity values in the  $B, x,$  and  $y$  channels. UCIQE is determined as:

$$UCIQE = 0.4680 * \sigma_{ch} + 0.2745 * c_l + 0.2576 * \mu_{sat}, \tag{9}$$

Fig. 3 Overall process of UIE showing all the eleven steps



**Fig. 4** Evaluation results at all levels of UIE when tested on an image from Trash\_ICRA19



**Table 2** Quantitative evaluation of the image used in Fig. 4

Image	UCIQE	UIQM
ShipWreck	0.642	0.678
Fishes	0.677	0.635
CoralReef1	0.678	0.698
CoralReef2	0.741	0.786
CoralReef3	0.699	0.799

where  $\sigma_{ch}$  is the standard deviation of chroma (ch),  $c_l$  is luminance contrast value, and  $\mu_{sat}$  is saturation average. All these values are evaluated separately as follows:

$$\sigma_{ch} = \frac{1}{N} \sum_{i=1}^N (ch_i^2 - \mu_{sat}^2) \tag{10}$$

$$ch_i = \sqrt{x_i^2 + y_i^2} \tag{11}$$

$$\mu_{sat} = \frac{1}{N} \sum_{i=1}^N sat_i \tag{12}$$

$$sat_i = \frac{ch_i}{B_i} \tag{13}$$

UIQM is a combination of UICM, UISM, and UIcon and is evaluated as:

$$UIQM = 0.0282 * UICM + 0.2953 * UISM + 3.5753 * UIcon. \tag{14}$$

UICM is a measurement of light-induced deterioration that is based on the statistical analysis of the variations between the red-green and yellow-blue planes. The strength of Sobel edges, which are independently computed on each colour channel, determines UISM. UIcon is the measurement of the contrast of underwater images.

### 3.4 Feature extractor

The feature extraction process transforms unstructured data into controllable numerical qualities while maintaining the integrity of the original data. Features can be extracted manually or automatically. The traits that are pertinent to a particular situation must be understood and described before manual feature extraction can be performed. Feature extraction for image data is used to represent the important regions of an image as a compact feature vector. An innovative feature extractor method has been presented in this work. The suggested approach is known as the Hybrid Extraction of Features approach (HEFA). The algorithm operates on the first video frame which is provided by the UIE module after enhancement. The algorithm learns about all the values of UCIQE and UIQM for each object present in the scene. The algorithm decomposes on the UCIQE and UIQM complexes. The algorithm analyses on doubled space scales ( $2^1 \dots 2^5$ ) and utilizes  $2^3$  and  $2^5$  scales for extracting the UCIQE and UIQM values respectively. After filtration, down-sampling is performed to remove redundancy from the images, generating a tree up to level  $k$ . These trees are considered feature spaces. The trees are formed for each input video frame, by evaluating squares of transform coefficients. Additionally, a logarithmic operator is applied to normalize the generated



features. Once all the trees are ready, decomposition is done to extract the appropriate features. Finally, each feature is ranked as important and their preceding and succeeding nodes are removed. The complete algorithm is summarized as follows:

**Algorithm 2** HEFA algorithm

---

**Input:**  $UICQE, UIQM, d, n, k$   
**Output:** features

1. For each enhanced image  $E_i$ , in any frame  $n$  perform a decomposition  $d$  to max level  $j$  to construct features as:
2.  $features_n = \log \left( \frac{\sum_n (d_{UIQM, UICQE}^x)^2}{N/2^j} \right)$
3. Construct the trees for features calculated till  $k$  levels
4. Till  $features_n$  determine the optimal decomposition value  $d$
5. Sort all the feature spaces in ascending order
6. Remove the succeeding and preceding features.
7. If feature space = 0 then stop otherwise go to step 3
8. End

---

The backbone of YOLOv8 is used for feature extraction and it uses CSP (Cross Stage Partial) connections. The Cross Stage Partial (CSP) [28] idea is used in the Backbone component of YOLOv8 to divide the feature map into two parts. While YOLO has demonstrated its capacity to detect items in anchor boxes and multiscale settings, it is not very good at accurately determining position when dealing with small objects. It is important to design a deep network to improve the detection accuracy. Therefore, a hybrid network of YOLOv8 and HEFA is proposed to extract features jointly in the backbone of YOLOv8. In HEFA, a node is added that connects the input with the output of the entire model. These inputs and outputs together provide the final output. This combination improves the training speed of the entire model as well.

**3.5 Finetuned YOLOv8**

The architecture of YOLOv8 is similar to that of YOLOv5. It also comprises backbone, head, and neck modules. The Backbone, Neck, Head, and Loss components of the YOLOv8 algorithm architecture are depicted in Fig. 5.

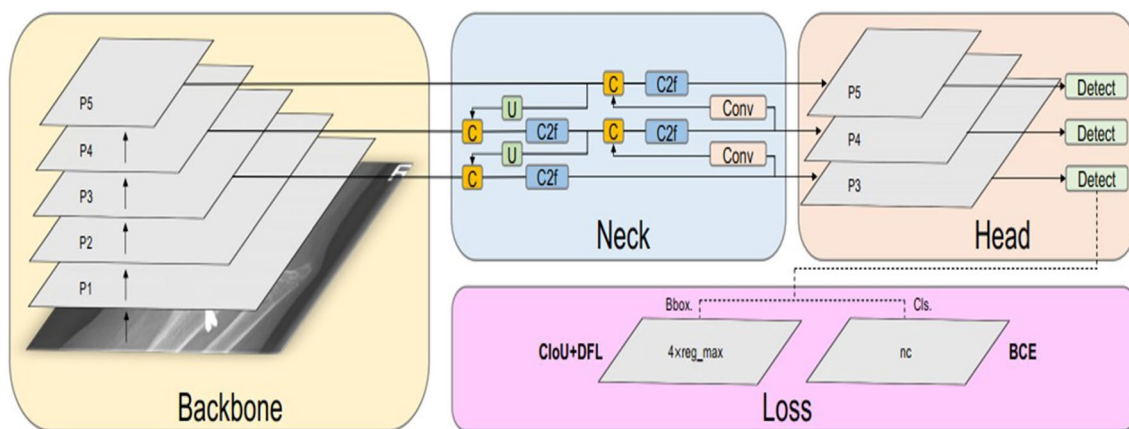
**3.5.1 Backbone module**

Component 1 employs convolution operations, while Part 2 is concatenated with the results of Part 1’s convolution operations. Convolutional neural networks (CNNs) train more effectively thanks to CSP design, which also requires less computational labour to run. YOLOv8 employs a C2f module as opposed to the C3 module used in YOLOv5 [30]. While the C2f module consists of 2 ConvModule and n BottleNeck coupled by Split and Concat methods, the C3 module comprises 3 ConvModule and n BottleNeck. Conv-BN-SiLU makes up ConvModule, as depicted in Fig. 6. By merging the C3 module with YOLOv7 [31], the C2f module is proposed, allowing YOLOv8 to collect deeper gradient flow knowledge while maintaining the compact size of the model.

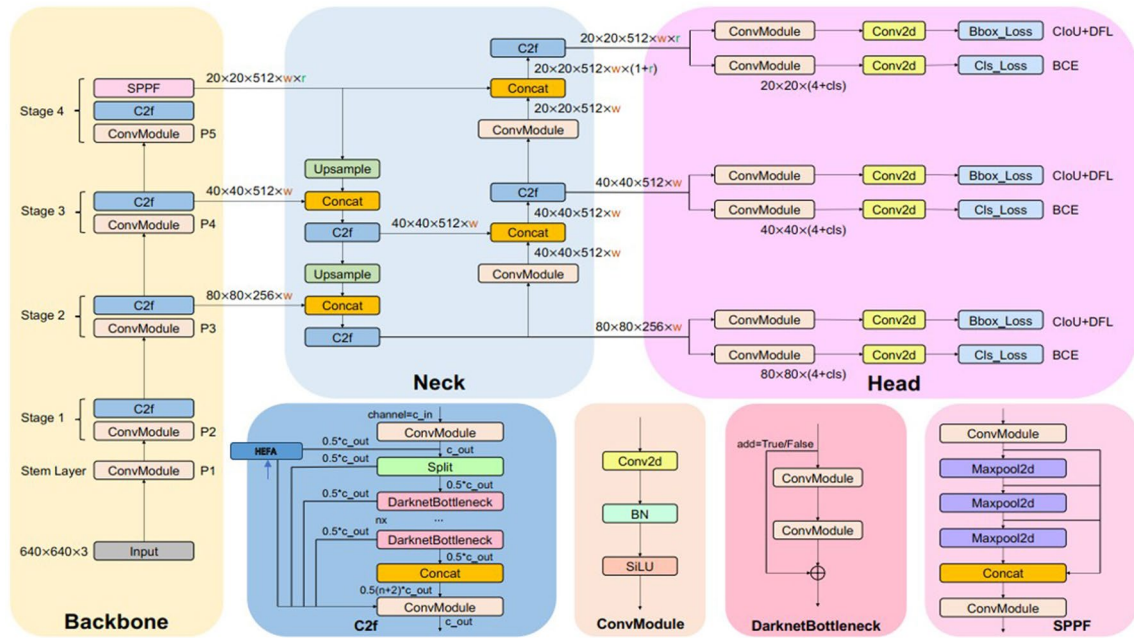
Additionally, the YOLOv8 method decreases the total amount of blocks in each step from 3,6,9,3 utilized in YOLOv5 to 3,6,6,3 to significantly reduce the computational burden of the model. The SPPF module utilized in YOLOv5 is followed in Stage 4 by YOLOv8, too. As seen in Fig. 6, SPPF improves SPP to speed up the model’s inference process.

**3.5.2 Neck module**

Deeper networks yield more detailed feature information and better object prediction outcomes. Deeper networks reduce the information on the location of objects, nevertheless. To



**Fig. 5** Overall architecture of baseline YOLOv8 [29]



**Fig. 6** Details of each module in baseline YOLOv8 integrated with the HEFA algorithm in the backbone module. The input from UIE is given to HEFA directly for feature extraction

avoid information loss for small objects, a multiscale fusion of features employing FPN [32] and PAN [16] frameworks is required. As depicted in Fig. 6, the Neck portion of the design employs multiscale feature fusion of images, where the lower features lose less location information due to fewer convolution layers and the top features gain more information due to more network layers. With YOLOv5, the bottom feature map can be up-sampled to keep more feature information, and the top feature can be down-sampled to preserve more position information. To provide reliable predictions for images of various sizes, these two feature outputs are finally combined. In the up-sampling step, YOLOv8 eliminates the convolution processes while adhering to FPN and PAN structures.

### 3.5.3 Head module

In contrast to YOLOv5, which employs a linked head, YOLOv8 separates the categorization and recognition heads using a decoupled head. As seen in Fig. 6, YOLOv8 preserves only the categorization and regression branches and eliminates the object branch. To correct the exact object position, Anchor-Base prepares a large number of anchors in the image and then calculates the four offsets of the regression item concerning the anchors. Anchor-Free, which locates the item by its centre and then estimates the distance from the centre to the bounding box, takes the role of Anchor-Base in YOLOv8.

### 3.5.4 Loss function

The YOLOv8 algorithm uses the Task Aligned Assigner of TOOD [43] for both negative and positive sample assignment to choose positive samples based on the weighted classification and regression scores, as illustrated in the equation that follows:

$$ta = p^\alpha * \mu^\beta, \tag{15}$$

where ta = task aligned assigner [33],  $p$  = predicted score of labelled class,  $\mu$  = IoU value of predicted bounding box to ground truth bounding box.

YOLOv8 comprises classification and regression branches, out of which the classification branch utilizes the BCE loss function which is depicted in Eq. (16) and the regression branch uses DFL (Distribute Focal Loss) and CIoU loss functions as depicted in Eqs. (17) and (19), respectively:

$$BCE_{loss} = -\frac{1}{N} \sum_{j=1}^N -[l_j \log p_j + (1 - l_j) \log (1 - p_j)], \tag{16}$$

where  $N$  = no. of classes,  $l_j$  = labelled value,  $p_j$  = predicted value

$$DFL(T_n, T_{n+1}) = -((a_{n+1} - a) \log (T_n) + (a - a_n) \log (T_{n+1})), \tag{17}$$

where

$$T_n = \frac{a_{n+1} - a}{a_{n+1} - a_n}, \quad T_{n+1} = \frac{a - a_n}{a_{n+1} - a_n}. \tag{18}$$

DFL focuses on the expansion of probability of values around the object  $a$ . In Eq. (17), " $a$ " is the regression target value which refers to the offset values of four sides of the bounding box,  $T_n, T_{n+1}$  are probability values of  $a_n$  and  $a_{n+1}$ . Equation (18) depicts how these probability values are calculated.

Due to its consideration of the aspect ratio between the anticipated and actual bounding boxes, the CIoU loss function (Complete IoU) aids in the addition of influence factors. It is mostly used for OD.

$$CIoU_{loss} = 1 - IoU + \frac{D^2}{D_C^2} + \alpha \vartheta \tag{19}$$

$$\alpha = \frac{\vartheta^2}{(1 - IoU) + \vartheta}, \tag{20}$$

where  $\vartheta$ =parameter for measuring consistency in aspect ratio,  $\alpha$  is the trade-off parameter defined in Eq. (19),  $D$  is the distance between two central points between two boxes, and  $D_C$  is the diagonal length of the smallest box enclosing two boxes.

### 3.5.5 Finetuning YOLOv8

It is trimmed to further speed up inference and compress the suggested model. Although YOLOv8 showed significant improvements on its predecessors in terms of accuracy and

speed it still has some limitations it does not provide significant accuracy on small objects, does not generalize well to unseen object classes, requires a large dataset to work on, and it is expensive in terms of computational costs. To overcome some of the limitations of baseline YOLOv8, it is pruned using a proposed pruning algorithm by Chhaya Gupta et al. [6]. Once the pruning is done the model's detection accuracy decreases. Hence to improve the same, a transfer learning algorithm is used for improving the detection accuracy of the proposed model [6]. By using the suggested pruning strategy, there is just one hidden  $3 \times 3$  convolutional layer left. Pruning aids in modifying the network's width and depth as well. This study suggests using hidden layer pruning to modify the network depth and regulate the number of residual components in the CSP module. The network depth of the YOLOv8 model after pruning is compared with the baseline YOLOv8 model and results are illustrated in Table 3. The architecture of the pruned YOLOv8 is shown in Fig. 7.

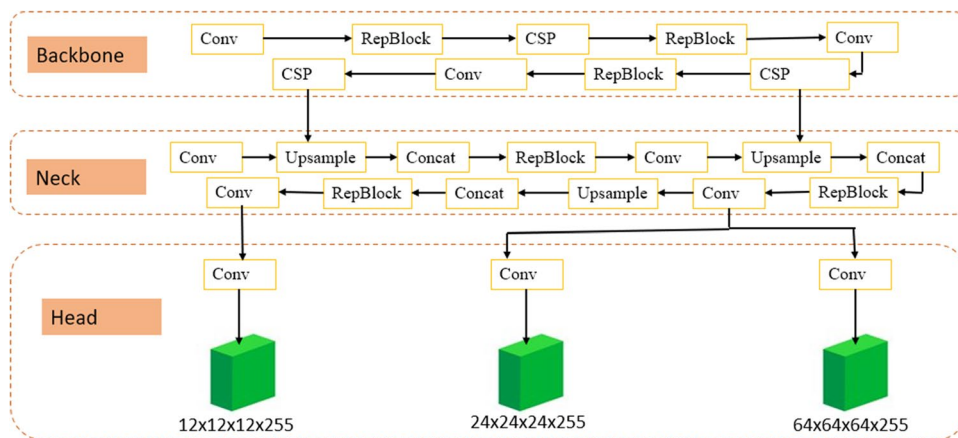
## 4 Experimental results and discussion

The Trash\_ICRA19 dataset was used in this study to train the suggested model. 5720 underwater image training samples make up the datasets. The images are a mix of aquatic animals and the trash present undersea. The dataset also comprises 820 validation images as shown in Fig. 8. The image size has been changed to  $640 \times 640$  pixels. The validation images were used as a test dataset for evaluating the proposed model.

**Table 3** Network depth comparison between baseline YOLOv8 and pruned YOLOv8 after applying pruning

Model	Backbone: CSP1_X			Neck: CSP2_X				
	1st	2nd	3rd	1st	2nd	3rd	4th	5th
Baseline Yolov8	CSP1_1	CSP1_3	CSP1_3	CSP2_2	CSP2_2	CSP2_2	CSP2_2	CSP2_2
Pruned Yolov8	CSP1_2	CSP1_6	CSP1_6	CSP2_1	CSP2_1	CSP2_1	CSP2_1	CSP2_1

**Fig. 7** The architecture of pruned YOLOv8



**Fig. 8** A few images from the Trash\_ICRA19 dataset



The experiment was performed on 2.0.0+cu118 pytorch CUDA 118 NVIDIA version GPU. For the proposed model's training, all of the experiments used 100 epochs. YOLOv8m, with a batch size of 64, is employed as the baseline model. The Adam optimizer was used to train the model with a learning rate of 0.01.

#### 4.1 Evaluation metrics

The proposed model's accuracy, precision, and recall rate are assessed using a mean average accuracy metric to gauge its performance. Precision is defined as the fraction of all boxes correctly predicted to all boxes generated by all networks.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (21)$$

Recall is defined as the ratio of accurately anticipated boxes to actual boxes, and it is as follows:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \quad (22)$$

A category's average precision accuracy, or AP, has an IoU threshold that ranges from 0.5 to 0.95. It refers to the single category indicator known as the area under the PR curve, which is defined as:

$$\text{Avg Precision} = \int_0^1 \text{PR} \, dr, \quad (23)$$

mAP stands for the average AP across all  $d$  categories and is defined as:

$$\text{mAP} = \frac{1}{d} \sum_{i=1}^d \text{AP}_i. \quad (24)$$

#### 4.2 Experimental results

This study presents the enhanced model's training outcomes on the Trash\_ICRA19 dataset after 100 iterations of model training. Figure 9 displays the training and validation sets' performance indicators. The first three columns show the box loss, object loss, and classification loss of the upgraded YOLOv8 model. The three loss curves are displayed in the

first three columns, with the training set's epochs on the  $x$ -axis and the overall loss value on the  $y$ -axis. The curves show that the total loss value decreases with time and stabilizes as training progresses. These results demonstrate that the upgraded YOLOv8 model proposed in this paper has a high level of consistency, accuracy, and a good fitting impact. The notion behind the updated YOLOv8 model was good, as shown in Fig. 9. Three different types of loss are shown in Fig. 9, including categorization loss, objectness loss, and box loss. The box loss quantifies how well the algorithm locates an object's centroid and how completely the object is surrounded by the expected bounding box. Essentially, objectness acts as a gauge for the probability that an object would surface in a suggested area of interest. If objectivity is high, a visible object is likely in the image window. Classification loss provides insight into an algorithm's ability to accurately forecast an object's category. The model's precision, recall, and mAP grew at first, but after roughly 50 epochs, they peaked. In the validation data, the box, objectness, and classification losses also demonstrated a considerable decline up to roughly epoch 50.

The confusion matrix of the recommended approach is shown in Fig. 10, along with the accuracy of prediction of our updated YOLOv8 model on each category of underwater garbage photos in the dataset and the associations between the predictions.

#### 4.3 Ablation study

This section carries out ablation tests to confirm the effectiveness and consistency of the additional strategy in enhancing YOLOv8 and assesses the influence of these improved approaches on the experiment results by removing each one individually. The ablation experiment results on Trash\_ICRA19 are shown in Table 4. "X" denotes the application of the improvement method. Figure 11 displays the curve found in Table 4. The detection findings of the baseline YOLOv8 network, which was utilized to carry out the ablation experiment, are shown in the top row of Table 4. In the following row of the table, the outcomes of using the HEFA algorithm—which enhances the original YOLOv8—are shown. The detection system can collect more detailed semantic data, which raises the mAP@0.5 and mAP@0.5:0.95 values over the initial



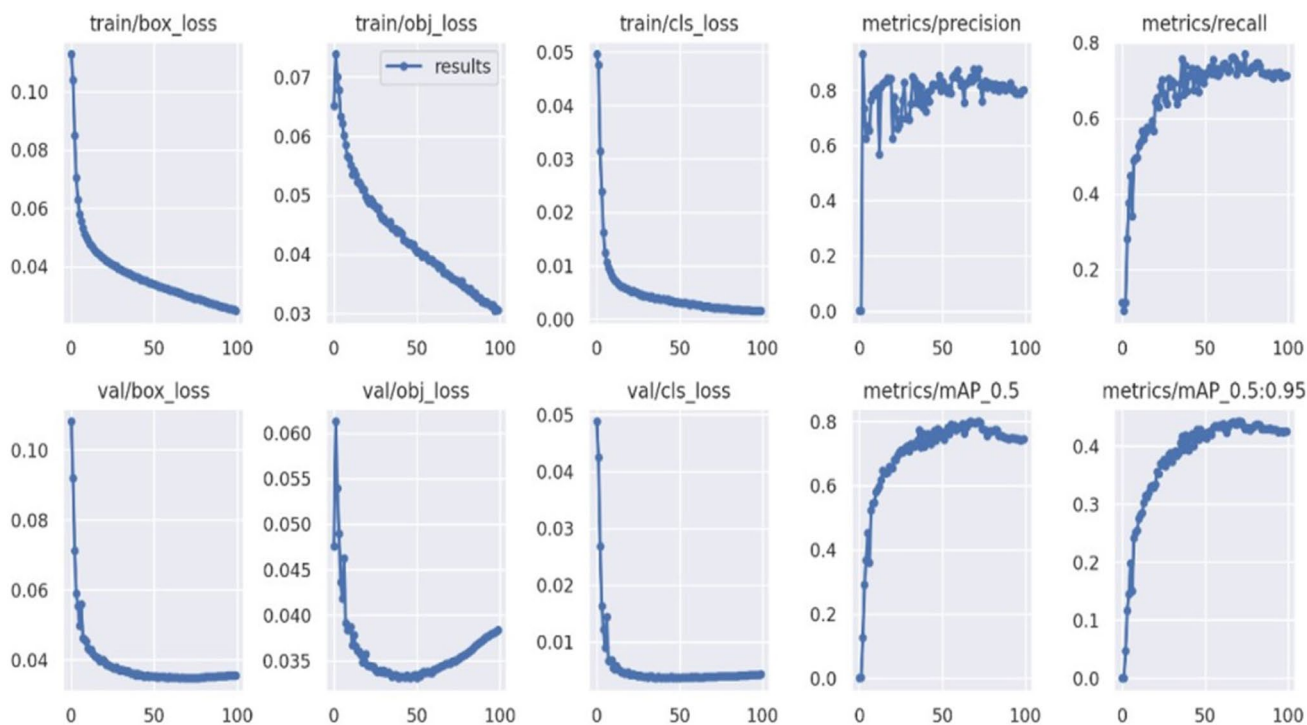


Fig. 9 All the logs depicting the performance evaluation of finetuned YOLOv8

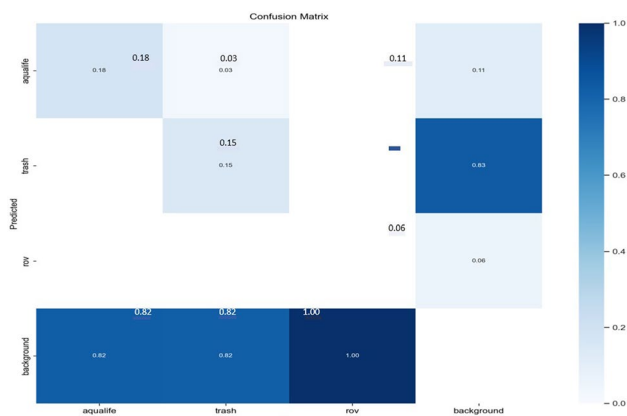


Fig. 10 Confusion matrix for proposed finetuned YOLOv8

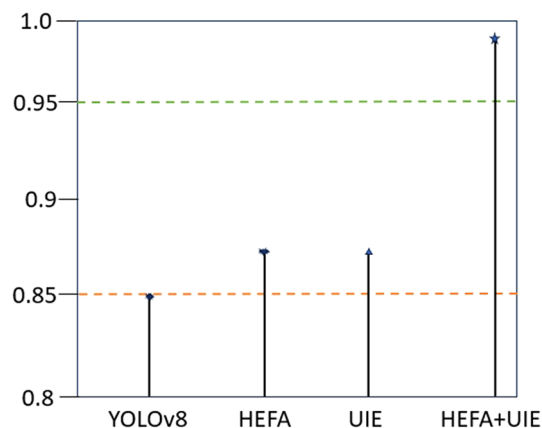


Fig. 11 Ablation study results on Trash\_ICRA19

Table 4 Ablation study on Trash\_ICRA19

HEFA	UIE	mAP@0.5	mAP@0.5:0.95
		0.853	0.824
X		0.875	0.856
	X	0.874	0.855
X	X	0.981	0.945

YOLOv8 values by 0.9% and 1.5%, respectively, in the absence of an enhancement plan. This shows that HEFA has improved the baseline YOLOv8 model by selecting the

relevant features only which helps in enhancing the mAP. The next row of the table then displays the outcomes of employing UIE only for data augmentation in OD. This is because OD in underwater images can be done with greater accuracy thanks to data augmentation techniques employed in image restoration; in comparison to the baseline, the model's mAP@0.5 and mAP@0.5:0.95 increased by 1.5% and 4.4%, respectively. These outcomes show that our modifications to YOLOv8 successfully raised detection accuracy.



### 4.4 Comparative analysis

This study compares state-of-the-art (SOTA) approaches with the suggested methodology. An analysis of the Trash\_ICRA19 dataset uses proposed YOLOv8, YOLOv7, YOLOv5, and Faster R-CNN models. Table 5 and Fig. 12 display the outcomes of the comparative experiment performed on the Trash\_ICRA19 dataset. The mAP@0.5 of 98.1% that the suggested enhanced YOLOv8 detection framework achieved is much higher than that of all other models. Additionally, it outperforms Faster-RCNN, YOLACT [34], YOLOv5 [18], YOLOv7 [35] and DETR [36]. Based on these findings, the performance of our suggested YOLOv8 model design is superior.

Several photos from the test dataset were chosen to confirm the modified model's viability. Figure 12 displays the comparative results of different state-of-the-art methods with the proposed model.

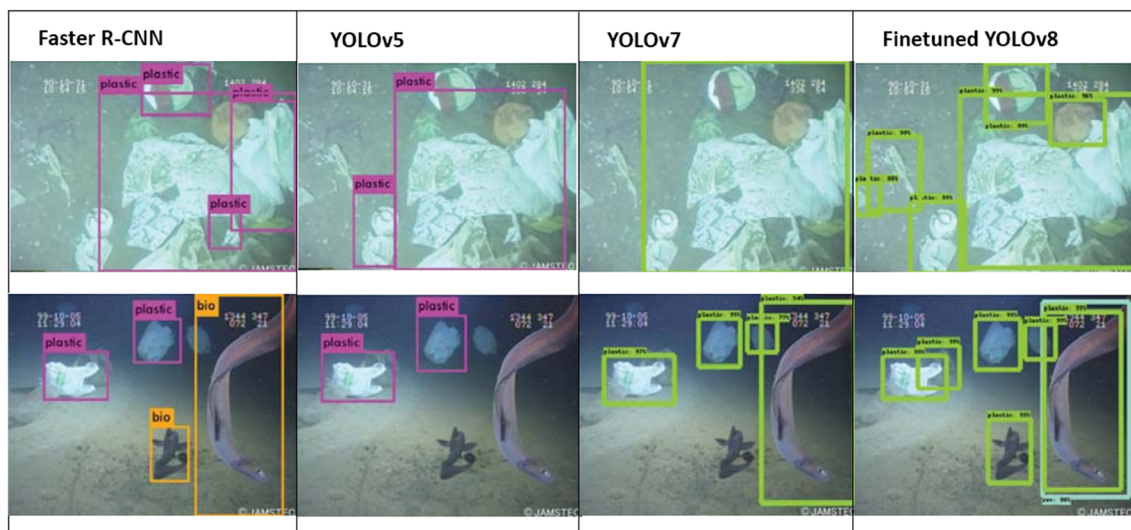
**Table 5** Comparative analysis of different SOTA methods with finetuned YOLOv8

Model	mAP@50 (%)	FPS(GPU)	Inference(ms)
Faster-RCNN	94.5	32	99.3
YOLOv5 [18]	97.5	35	66.4
YOLOv6 [37]	90.0	NA	66.8
YOLOv7 [35]	80.71	32	71.1
YOLACT [34]	36.7	5	99.4
DETR [36]	95.2	NA	NA
Baseline YOLOv8	97.89	34	66.7
Proposed	<b>98.1</b>	<b>35</b>	<b>67.9</b>

In summary, the enhanced YOLOv8 network continued to exhibit strong detection capabilities in settings with intricate backgrounds and low-contrast underwater lighting. The studies' findings demonstrate that the upgraded YOLOv8 object-detection algorithm outperforms current models, increases object recognition precision, and fundamentally satisfies the requirements of underwater target-detection tasks.

### 5 Conclusion and future work

The complicated and ever-changing underwater environment is considered when a novel and improved YOLOv8 underwater rubbish identification model is suggested. The innovative UIE module algorithm, which permits the enhancement of underwater trash photos, was first introduced. Second, a novel feature extractor algorithm was suggested for quickly extracting suitable features from the improved images. The experiment showed that the framework's mAP@0.5 and mAP@0.5:0.95 attained 98.1% and 54.2%, respectively, on the Trash\_ICRA19 dataset. These values were higher than those of the original YOLOv8 and other SOTA models. Because it has a significantly greater object-detection accuracy than YOLOv8, the suggested upgraded technique is a highly beneficial general framework for underwater trash recognition. The improved YOLOv8 model has shown promising results in identifying dense underwater objects and might be widely applied to detecting marine objects in complex underwater environments. Improved transformer blocks and more effective image-enhancement techniques could be the subject of future research to strengthen the network's architecture and boost detection accuracy. A few flaws in the image enhancer suggested in this work will be



**Fig. 12** Comparative results of proposed finetuned YOLOv8 with other SOTA methods

considered in the future. The main drawback of the suggested picture enhancer is that it reduces the size of the output image by degrading some of the original image's pixels. This restriction will be taken into account for future work. In this experiment, the image enhancer used only three weights. The study may be extended further and the exposedness weight map may also be computed, but it has been left as future work.

**Author contributions** All authors have equally contributed for this paper.


**Data availability** The data used in the work is freely accessible via Fulton et al. [38].

## Declarations

**Competing interests** The authors declare no competing interests.

## References

- Namadi, P., Deng, Z.: Deep learning-based ensemble modeling of *Vibrio parahaemolyticus* concentration in marine environment. *Environ. Monit. Assess.* (2023). <https://doi.org/10.1007/s10661-022-10836-9>
- Zhao, W., Han, F., Qiu, X., Peng, X., Zhao, Y., Zhang, J.: Research on the identification and distribution of biofouling using underwater cleaning robot based on deep learning. *Ocean Eng.* **273**, 113909 (2023). <https://doi.org/10.1016/j.oceaneng.2023.113909>
- Xu, S., Zhang, M., Song, W., Mei, H., He, Q., Liotta, A.: A systematic review and analysis of deep learning-based underwater object detection. *Neurocomputing* **527**, 204–232 (2023). <https://doi.org/10.1016/j.neucom.2023.01.056>
- Farid, A., Hussain, F., Khan, K., Shahzad, M., Khan, U., Mahmood, Z.: A fast and accurate real-time vehicle detection method using deep learning for unconstrained environments. *Appl. Sci.* (2023). <https://doi.org/10.3390/app13053059>
- Sangeeta, G.P.: Improved video compression using variable emission step ConvGRU based architecture. *Lect. Notes Data Eng. Commun. Technol.* **61**, 405–415 (2021). [https://doi.org/10.1007/978-981-33-4582-9\\_31/COVER](https://doi.org/10.1007/978-981-33-4582-9_31/COVER)
- Gupta, C., Gill, N.S., Gulia, P., Chatterjee, J.M.: A novel finetuned YOLOv6 transfer learning model for real-time object detection. *J. Real Time Image Process.* (2023). <https://doi.org/10.1007/s11554-023-01299-3>
- Diwan, T., Anirudh, G., Tembhurne, J.V.: Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* (2022). <https://doi.org/10.1007/s11042-022-13644-y>
- Gupta, C., Gill, N.S., Gulia, P.: SSDT: distance tracking model based on deep learning. *Int. J. Electr. Comput. Eng. Syst.* **13**, 339–348 (2022). <https://doi.org/10.32985/ijeces.13.5.2>
- Mittal, U., Chawla, P., Tiwari, R.: EnsembleNet: a hybrid approach for vehicle detection and estimation of traffic density based on faster R-CNN and YOLO models. *Neural Comput. Appl. Comput. Appl.* **35**, 4755–4774 (2023). <https://doi.org/10.1007/s00521-022-07940-9>
- Qiu, Z., Rong, S., Ye, L.: YOLF-ShipNet: improved RetinaNet with pyramid vision transformer. *Int. J. Comput. Intell. Syst.* (2023). <https://doi.org/10.1007/s44196-023-00235-4>
- Peng, W.Y., Peng, Y.T., Lien, W.C., Chen, C.S.: Unveiling of how image restoration contributes to underwater object detection. In: 2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp. 1–2. IEEE (2021). <https://doi.org/10.1109/ICCE-TW52618.2021.9602998>
- Liu, K., Peng, L., Tang, S.: Underwater object detection using TC-YOLO with attention mechanisms. *Sensors* (2023). <https://doi.org/10.3390/s23052567>
- Wang, H., Sun, S., Bai, X., Wang, J., Ren, P.: A reinforcement learning paradigm of configuring visual enhancement for object detection in underwater scenes. *IEEE J. Ocean. Eng.* (2023). <https://doi.org/10.1109/JOE.2022.3226202>
- Song, P., Li, P., Dai, L., Wang, T., Chen, Z.: Boosting R-CNN: reweighting R-CNN samples by RPN's error for underwater object detection. *Neurocomputing* **530**, 150–164 (2023). <https://doi.org/10.1016/j.neucom.2023.01.088>
- Lee, M.F.R., Chen, Y.C.: Artificial intelligence based object detection and tracking for a small underwater robot. *Processes* (2023). <https://doi.org/10.3390/pr11020312>
- Yu, H., Li, X., Feng, Y., Han, S.: Multiple attentional path aggregation network for marine object detection. *Appl. Intell.* **53**, 2434–2451 (2023). <https://doi.org/10.1007/s10489-022-03622-0>
- Son, Y.-T., Jin, S.-Y., Kang, T.-S.: Object detection and classification applying AI (computer vision) to underwater images. *EGU23* (2023). <https://doi.org/10.5194/EGUSPHERE-EGU23-2203>
- Wu, C.M., Sun, Y.Q., Wang, T.J., Liu, Y.L.: Underwater trash detection algorithm based on improved YOLOv5s. *J. Real Time Image Process.* **19**, 911–920 (2022). <https://doi.org/10.1007/s11554-022-01232-0>
- Zhang, X., Fang, X., Pan, M., Yuan, L., Zhang, Y., Yuan, M., Lv, S., Yu, H.: A marine organism detection framework based on the joint optimization of image enhancement and object detection. *Sensors* **21**, 1–17 (2021). <https://doi.org/10.3390/s21217205>
- Wang, C.C., Samani, H., Yang, C.Y.: Object Detection with Deep Learning for Underwater Environment. *Proceedings of 4th International Conference Information Technology Res. Bridg. Digit. Divid. Through Multidiscip. Res.*, pp. 1–6. ICITR (2019). <https://doi.org/10.1109/ICITR49409.2019.9407797>
- Ji, S.J., Ling, Q.H., Han, F.: An improved algorithm for small object detection based on YOLO v4 and multi-scale contextual information. *Comput. Electr. Eng. Electr. Eng.* **105**, 108490 (2023). <https://doi.org/10.1016/j.compeleceng.2022.108490>
- Zhang, J., Zhang, J., Zhou, K., Zhang, Y., Chen, H., Yan, X.: An improved YOLOv5-based underwater object-detection framework. *Sensors* **23**, 1–21 (2023)
- Liu, K., Sun, Q., Sun, D., Peng, L., Yang, M., Wang, N.: Underwater target detection based on improved YOLOv7. *Mar. Sci. Eng.* (2023). <https://doi.org/10.23919/CCC55666.2022.9901920>
- Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., Chen, H.: DC-YOLOv8: small-size object detection algorithm based on camera sensor. *Electronics* **12**(10), 2323 (2023). <https://doi.org/10.20944/preprints202304.0124.v1>
- Li, Y., Fan, Q., Huang, H., Han, Z., Gu, Q.: A modified YOLOv8 detection network for UAV aerial image recognition. *Drones* **7**, 304 (2023)
- Kim, J.H., Kim, N., Won, C.S.: High-speed drone detection based on Yolo-V8. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–2. IEEE (2023)
- Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., Chen, H.: DC-YOLOv8: small-size object detection algorithm based on camera sensor. *Electron* **12**, 1–14 (2023). <https://doi.org/10.3390/electronics12102323>
- Wang, C.Y., Mark Liao, H.Y., Wu, Y.H., Chen, P.Y., Hsieh, J.W., Yeh, I.H.: CSPNet: a new backbone that can enhance learning capability of CNN. *IEEE Comput. Soc. Conf. Comput. Vis.*

- Pattern Recognit. Work. 2020-June, pp. 1571–1580 (2020). <https://doi.org/10.1109/CVPRW50498.2020.00203>
29. Ju, R.-Y., Cai, W.: Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. *Sci. Rep. Rep* **13**, 1–12 (2023)
  30. GitHub - ultralytics/yolov5: YOLOv5  in PyTorch > ONNX > CoreML > TFLite, <https://github.com/ultralytics/yolov5>
  31. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. 1–15 (2022)
  32. Li, X., Yu, H., Chen, H.: Multi-scale aggregation feature pyramid with cornerness for underwater object detection. *Vis. Comput. Comput.* (2023). <https://doi.org/10.1007/s00371-023-02849-3>
  33. Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W.: TOOD: task-aligned one-stage object detection. *Proc. IEEE Int. Conf. Comput. Vis.* (2021). <https://doi.org/10.1109/ICCV48922.2021.00349>
  34. Corrigan, B.C., Tay, Z.Y., Konovessis, D.: Real-time instance segmentation for detection of underwater litter as a plastic source. *J. Mar. Sci. Eng.* (2023). <https://doi.org/10.3390/jmse11081532>
  35. Wang, Z., Zhang, G., Luan, K., Yi, C., Li, M.: Image-fused-guided underwater object detection model based on improved YOLOv7. *Electron* **12**, 1–12 (2023). <https://doi.org/10.3390/electronics12194064>
  36. Yuan, X., Fang, S., Li, N., Ma, Q., Wang, Z., Gao, M., Tang, P., Yu, C., Wang, Y.: Performance comparison of sea cucumber detection by the Yolov5 and DETR approach. (2023)
  37. Walia, J.S., Seemakurthy, K.: Optimized custom dataset for efficient detection of underwater trash, pp. 292–303. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-43360-3\\_24](https://doi.org/10.1007/978-3-031-43360-3_24)
  38. Fulton, M., Hong, J., Islam, M.J., Sattar, J.: Robotic detection of marine litter using deep visual detection models. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, pp. 5752–5758 (2019). <https://doi.org/10.1109/ICRA.2019.8793975>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.