Iranian Journal of Fuzzy Systems

Volume *, Number *, (****), pp. **

Original paper

University of Sistan and Baluchestan

Dual deep feature fusion and neuro-fuzzy classification with grad-CAM++ for robust underwater object detection

T. Rama M [1] and A. V. Turukmane [2]

[1,2] *School of Computer Science Engineering, VIT-AP University, Beside AP Secretariat Amaravati, 522241, Andhra Pradesh, India.*

tulasirama.24PHD7147@vitap.ac.in, anil.turukmane@vitap.ac.in

Abstract

Detecting underwater objects is crucial for scientific inquiry, resource development, and marine conservation. Current underwater object detection methods face two main challenges: (1) high parameter counts leading to excessive memory usage, and (2) reduced detection accuracy. To address these limitations, this study introduces a computationally efficient underwater object recognition framework that integrates advanced feature extraction with fuzzy-driven adaptive classification. A novel dual-stream extractor combining Improved YOLOv9 for refined spatial localization and a Modified TResNet for enhanced semantic feature extraction is developed to capture complementary deep features. These streams are fused using Rényi entropy, ensuring that only the most informative and non-redundant representations are preserved. The fused features are then classified using a Neuro-Fuzzy Random Vector Functional Link (NF-RVFL) network, enabling fast, robust, and interpretable decision-making with minimal parameter overhead. Grad-CAM++ is incorporated to highlight critical attention regions, further improving framework transparency. Extensive experiments conducted on four public underwater datasets URPC, DLMU 2024, DUO, and RUOD show that the proposed framework consistently surpasses state-of-the-art detectors, achieving notable improvements in mAP, across all benchmarks. Comprehensive ablation analysis and statistical evaluations, confirm the contribution of each proposed component. Furthermore, deployment evaluation demonstrates that the proposed framework's reduced parameter count and high efficiency make it well-suited for real-time underwater platforms with limited computational resources.

Keywords: Underwater objects, improved YOLOv9, modified TResNet, neuro-fuzzy random vector functional link (NF-RVFL) neural network, Grad-CAM++.

1 Introduction

More than seventy percent of Earth's surface consists of oceans, which are home to vast amounts of material and biological resources and are crucial to the upkeep of the global ecology [25]. Therefore, creating efficient undersea monitoring and analysis systems is essential to achieving environmentally friendly practices and sustainable resource use. Contrarily, traditional underwater operations have historically depended on costly and insufficient manual interventions, which makes it challenging to satisfy the growing needs for environmental preservation and marine expansion [14]. The application of object identification strategies in other fields has encouraged the gradual introduction of similar techniques into the exploitation of underwater resources in recent years, creating new avenues for underwater monitoring. In recent years, with the increasing utilization of remotely operated vehicles (ROVs) [18] and autonomous underwater vehicles (AUVs) [9] particularly in the field of environmental monitoring and conservation of ecosystems, the need for advanced UOD strategies have grown tremendously.

While general object detection systems, such as Faster R-CNN and YOLO, have achieved great success in various scenarios, they are not very effective in water-based environments due to a variety of challenges [5]. Objects underwater

Corresponding Author: A. V. Turukmane
Received: July 2025; Revised: August 2026; Accepted: September 2026.
https://doi.org/10.22111/ijfs.2026.52497.9262

also often vary in size, shape and appearance, and can fit in with complex environments. These properties make it difficult to locate, identify and detect hidden, and small objects. But current techniques are not yet able to tackle these challenges, limiting them to more restricted underwater conditions.

Researches have proposed several solutions to address these challenges, including multi-scale feature extraction and underwater image enhancement. Then, the quality of underwater photographs can be slightly enhanced using conventional techniques of image correction, such as histogram equalization and manual light changes based on physics principles. However, they are only applicable to complex underwater scenarios to a certain extent [8]. CNNs and GANs are two deep-learning methods that greatly improve image quality. But it is generally techniques aimed at enhancing human vision, not machine vision, and this could lead to the loss of important features for detection. GCC-Net points out that enhanced photos can change intrinsic features that should be maintained; for example, colour shift correction can cause texture loss, which could make detection more difficult [17]. Furthermore, noise and artifacts unrelated to the identification task may be present in improved underwater photos, which would drastically reduce detection accuracy. The study illustrates how an overemphasis on colour restoration during the image enhancement process may result in a disregard for the blurring of small object borders, which is crucial for accurate detection [1].

Additionally, recent studies have closely examine the optimization of object identification and image enhancement methods. To produce images that are more suitable for object recognition, Hybrid Detection GAN suggests a perception-driven image enhancement method that is trained in tandem with a detection network [3]. However, the cooperative optimization of several modules offers strong parameter coupling and considerably raises the computational difficulty. Another method improved color correction and detection tasks at the same time by using lightweight deep neural networks, which increased image quality and detection accuracy. These techniques significantly improve the detection effectiveness and usefulness of underwater images. When applied in more complex and varied situations, they do have limitations [7]. Therefore, in this research, we introduced a novel research framework.

The key contributions of this paper are as follows,

- To enhance underwater image quality, a combination of Wiener filtering, histogram equalization, and normalization is used during preprocessing.
- The Improved YOLOv9 and Modified TResNet are jointly employed to extract both spatial and semantic features, improving the robustness of object detection.
- For effective feature fusion, the Rényi entropy technique is applied to retain informative features and reduce redundancy.
- The NF-RVFL neural network is used for fast, adaptive classification and is suitable for handling underwater image variations.
- To improve interpretability, Grad-CAM++ is integrated to generate heatmaps that highlight important regions used for decision-making.
- The proposed framework effectively addresses existing limitations by offering a balanced solution with higher accuracy, better feature representation, and interpretability in challenging underwater conditions.

The rest of the research work is structured as follows. In Section 2, we examine relevant research on underwater object recognition. Section 3 provides an extensive overview of the proposed framework. In Section 4, experiments and findings are presented. Section 5 summarizes the conclusion.

2 Review of existing related works

In this section, we analyzed the existing research.

1. Enhancement-based underwater detection methods

Dai et al. [6] came up with GCC-Net that handles low contrast and low visibility with a real time UIE enhancement method that enhances visibility of items in degraded areas. They also added a cross domain interaction module to obtain complementary information from both raw and enhanced images. Wang et al. [23] introduced DJL-Net, which is a joint learning network consisting of two branches combining image processing with object detection. The proposed edge enhancement module refines image boundaries and captures the shape and texture details of grayscale images.

Furthermore, Liu et al. [16] designed UnitModule, a plug-and-play module for joint underwater image enhancement that is trained using an unsupervised loss function and requires no additional datasets. The strategy is also enhanced to overcome different color distortions in underwater images by the use of color cast predictor and UCRT augmentation strategy. Tian et al. [21] proposed a lightweight method, a combination of the MSRCR enhancing technique and YOLOX, GhostNet. The feature pyramid network is also provided with a multi-attention module (LCR), which involves the level, channel, and a spatial attributes to improve the detection accuracy.

Cao et al. [2] introduced a two-enhancement-mechanism detection algorithm consisting of: a group contrastive-based feature enhancement module (GCFEM) which compares the various UIE-enhanced versions, and a prior-guided dual-reference feature enhancement module (PDFEM), which reinforces object representation. Li et al. [15] introduced SU-YOLO, a spiking neural network that has a spike-based denoising technique underwater and separated batch normalization (SeBN) to enhance the quality of feature map in the presence of underwater noise.

To train a mapping of deep representations between the severely degraded patches of DFUI, and underwater images, Song et al. [20] introduced a residual feature transference the module (RFTM). This mapping is further employed as a much-degraded prior (HDP) in underwater detection. The use of HDP to teach without semantic labeling and incorporation into existing, well-known CNN-based feature extraction networks can improve their performance on underwater objects detection as the statistical properties are not dependent on the content of the image.

Fu et al. [10], proposed enhancing R-CNN, a two-stage underwater detector that has three major components. To replicate the object prior probability, a new region proposal network named RetinaRPN is developed initially. It provides high-quality proposals and considers the issue of objectless and IoU prediction uncertainty. To calculate the outcome detection score, a probabilistic inference pipeline is used to combine the first stage prior ambiguity and the second stage categorization score to produce the final detection score. Finally, we propose a new hard example mining algorithm, which is boosting reweighting.

2. Architecture-based detection methods (backbone, transformer, lightweight networks)

Gao et al. [11], presented CSWin-Transformer to underwater detection and have developed rich dependencies between low and high level features and optimized hierarchically with a differentiable loss. The work by Guo et al. [12] enhanced YOLOv8s to perform underwater detection through the use of a small FasterNet backbone, minimizing computation and maximizing speed to enable real-time detection. As a solution to the complex feature scenarios, Zhou et al. [29] suggested BSR5 backbone based on Spatial Residual block and SkipCut to slice channel features and efficiently work with them. Ji et al. [13] introduced FBDPN, a CNN-Transformer hybrid framework that is based on heuristic feature pyramid networks. An introduction of the NSFBM module is provided to reinforce contextual information between features of the neighborhood scale.

Wu et al. [24] presented U-ATSS, lightweight one-stage underwater detector, that utilizes a compressed ATSS backbone, to significantly reduce the number of parameters without compromising accuracy. Zhang et al. [28] proposed a lightweight architecture of PRCII, a small underwater target detector, which uses PR-AIFI to interact with the features of different scales and CII-FPN to fuse features across different scales and obtain a better spatial representation.

3. Fusion-based multimodal or multiscale methods

The acousto-optic fusion method suggested by Yu et al. [26] involves pairing sonar and optical images and aligning them with simulated seabed planes. The method uses a fusion backbone, dual neck, and dual head to exchange cross-modal information, and a twin-branch attention module is used to combine the features.

CGIS was introduced by Shen et al. [19] as a crisscross global interaction strategy (SCGIS and ECGIS varieties) that minimized background interference and selectively redirected computational resources by the use of adaptive attention across multiple target dimensions. To detect salient objects underwater Yuan et al. [27] advanced to RGB and depth fusion along with the features transformer-CNN hybrid feature extraction, interactive cross-attention and self-attention fusion, and cross-scale learning to enhance prediction of coarse and fine features. The summary of related work is shown in Table 1.

Table 1: Summary of related work

References	Category	Key Characteristics	Strengths	Limitations
Dai et al. [6]	Enhancement-based	GCC-Net with UIE enhancement + cross-domain interaction	Improves visibility; combines raw + enhanced features	Performance depends heavily on UIE quality
Wang et al. [23]	Enhancement + Joint Learning	DJL-Net with dual-branch detection + enhancement	Strong edge and texture enhancement	Multi-branch structure increases complexity
Liu et al. [16]	Enhancement (Plug-and-play)	UnitModule + UCRT, unsupervised training	Works without paired data	Unsupervised optimization may be unstable
Tian et al. [21]	Enhancement + Lightweight	MSRCR + YOLOX + GhostNet + LCR attention	multi-attention feature fusion	MSRCR may over-enhance noisy regions
Cao et al. [2]	Enhancement + Feature Reinforcement	GCFEM + PDFEM dual enhancement	Strong object representation and UIE–detection synergy	UIE multi-version comparison increases preprocessing cost
Li et al. [15]	Neuromorphic Architecture	SU-YOLO with SNN + SeBN	Very energy-efficient	Training SNNs is difficult and unstable
Song et al. [20]	Enhancement + Prior Learning	RFTM + HDP for degraded prior modeling	enhances CNN detectors	Requires degraded-patch mapping computation
Fu et al. [10]	Two-stage Detector	Enhanced R-CNN + RetinaRPN + probabilistic scoring	better uncertainty handling	Two-stage systems have high inference cost
Gao et al. [11]	Architecture (Transformer)	CSWin-Transformer with hierarchical optimization	Strong global–local dependency modeling	Transformer-heavy → high computation
Guo et al. [12]	Lightweight Architecture	YOLOv8s + FasterNet	Fast and real-time; reduced computation	Limited robustness in harsh distortions
Zhou et al. [29]	Architecture	BSR5 backbone + SkipCut	Efficient multi-scale channel processing	Not specialized for underwater distortions
Ji et al. [13]	Fusion-based	FBDPN (CNN–Transformer) + NSFBM	Strong contextual fusion across scales	Transformer modules increase latency
Wu et al. [24]	Lightweight Detector	Compressed ATSS backbone	low parameters	Limited advanced multiscale refinement
Zhang et al. [28]	Small-target Architecture	PRCII-Net with PR-AIFI + CII-FPN	Strong for small-object detection	Requires extensive hyperparameter tuning
Yu et al. [26]	Fusion-based (Acousto–Optic)	Sonar + optical fusion + dual-head	Excellent for murky	Requires paired sonar–optical data
Shen et al. [19]	Attention-based Fusion	CGIS (SCGIS/ECGIS) with global interaction	Strong global context	Heavy attention overhead
Yuan et al. [27]	RGB–Depth Fusion	Transformer–CNN hybrid + cross-attention	Excellent fine + coarse feature fusion	Requires depth sensors; higher compute cost

2.1 Research gap

In terms of accuracy and detection speed, the research mentioned above has produced beneficial results. However, it is frequently challenging to satisfy the actual detection demands by merely increasing the speed or precision because of the novel characteristics of the underwater environment. An effective underwater detection framework with a dual-stream feature extractor is therefore required. The proposed framework integrates fuzzy-based classification with complementary spatial and semantic feature extraction to improve detection accuracy while maintaining computational efficiency. This design provides a balanced solution for accurate and real-time underwater object detection under challenging environmental conditions.

3 Proposed framework

In this part, we provide a detailed explanation of our proposed framework. Our proposed framework consists of preprocessing, feature extraction, feature fusion, classification and heatmap. Initially, noise reduction and normalization are performed in preprocessing image enhancement. After preprocessing the spatial and semantic features are extracted using the Improved YOLOv9 and Modified TResNet networks. Then, both feature representations are fused using the Rényi entropy-based the fusion mechanism. Finally, the classification was performed, using the NF-RVFL neural network. Grad-CAM++ is then used to locate the detected objects. The overall architecture of the proposed framework is illustrated in Figure 1.

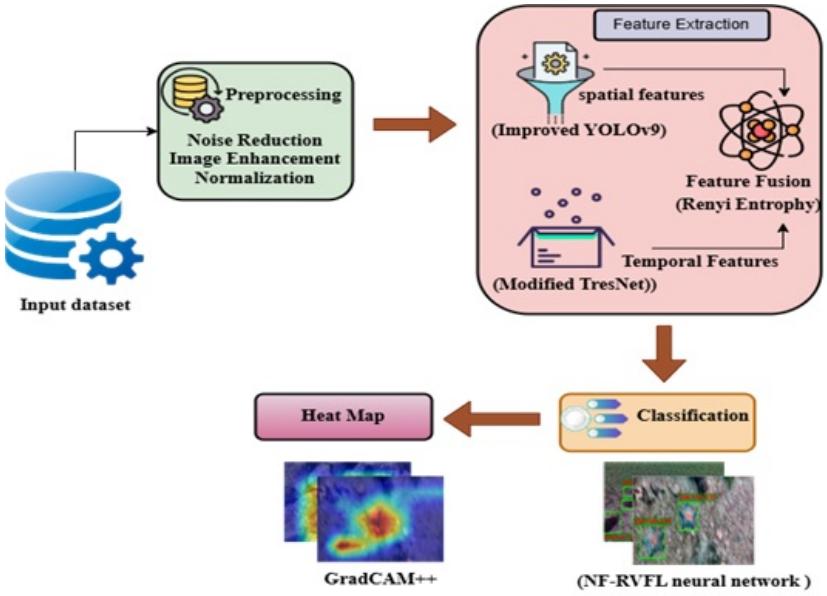

Figure 1: The overall architecture framework of proposed methodology

3.1 Preprocessing

In the pre-processing stage, essential image enhancement techniques are used to enhance image quality in underwater images. In particular, histogram equalization is used to improve the contrast in images, normalization is used to ensure that the pixel intensity values are normalized, and Wiener filtering is used to remove noise from images. To reduce blur caused by Gaussian smoothing while preserving important image details, a 2D adaptive noise reduction filter with a window size of 3×3 is adopted. The filtering process estimates the local mean and variance of each pixel to effectively generate noise reduced images.

Underwater images exhibit considerable variations in spatial resolution, illumination, and imaging conditions. Therefore, all input images are resized to 256×256 pixels to provide a uniform input representation, reduce computational complexity, and improve the efficiency of both training and inference. Although reducing the input resolution may decrease the amount of fine-grained spatial information, the proposed framework is specifically designed to preserve and recover discriminative features required for small-object detection. In particular, the introduced P2 detection head utilizes high-resolution shallow feature maps to retain fine spatial details that are essential for detecting small and micro-scale underwater targets. Moreover, the adaptive cross-attention fusion mechanism is helps boost local feature representation and reduce background noise, while the multi-scale feature aggregation mechanism integrates shallow and deep semantic features to improve the accuracy of object localization at different scales. The proposed architecture, therefore, counteracts this decrease in input resolution through the use of architectural elements, while also providing robust detection performance of small object underwater without compromising computational efficiency, which is important for real-time deployment.

It is important to note that the proposed framework does not rely solely on input image resolution for small-object detection. On the other hand, the added P2 detection head maintains high-resolution feature representations obtained from the early backbone layers, which allows to recover fine spatial details that are usually lost due to image downsampling. Therefore, the proposed framework focuses on feature representation learning, which does not involve higher resolution of input data, and thus it has better detection performance with lower computational complexity.

Moreover, every image is normalized so its pixel values fall between 0 and 1, which will make training more stable and decrease gradient explosion and other problems due to larger ranges of input intensities (0-255). During training, data augmentation techniques such as rotation, horizontal flipping, and vertical flipping are applied to enhance model generalization and prevent overfitting.

3.2 Feature extraction

A novel dual-stream technique is presented in the feature extraction phase to extract both spatial and semantic data from underwater images efficiently. Modified TResNet captures high-level semantic representations, while exact object-level spatial features are extracted using Improved YOLOv9. This integration preserves the context and details. This is an important step as it can help mitigate many of the challenges you might encounter underwater, like not being able to see or background noise. The features retrieved are more informative and discriminative, which greatly enhances the accuracy and robustness of identification and categorization tasks performed following retrieval.

3.2.1 YOLOv9 network and improvement

The YOLO [4] architecture has been refined multiple times and YOLOv9 offers significant advances in its efficiency and localization precision. Its reworked neural topology is designed to be more flexible to allow for the use of a variety of different computational units among the nodes. YOLOv9 is a cutting-edge detection framework with good accuracy and better computational efficiency compared to previous versions of YOLO. However, it is still not very efficient at detecting and recognizing micro-targets in very complex underwater scenes, and often predicts incorrectly. Moreover, the full model of YOLOv9 requires significant computing power to run and its direct deployment on mobile or edge devices remain difficult without the use of lighter or pruned variants.

Underwater benchmark datasets contain numerous small-scale objects, including sea cucumbers, sea urchins, scallops, starfish, and other marine organisms that frequently occupy only a small portion of the image. These objects are particularly challenging to detect because of low contrast, turbidity, scattering effects, non-uniform illumination, and complex underwater backgrounds. The original YOLOv9 architecture employs three detection heads (P3, P4, and P5), which mainly focus on medium- and large-scale objects. Consequently, this study introduces an additional P2 detection head to preserve high-resolution shallow feature representations, thereby improving the localization and recognition of small and micro-scale underwater objects.

The conventional CNN method is incomplete without the convolutional layer that comprises multiple convolutional kernels. Moreover, classification images are achieved through the effective process of convolutional layer to extract characteristic information of images and then feed it into the classification layer of CNN. Lightweight detection methods are suitable when using depthwise convolution (DWConv), a type of convolution. One of the possible solution to the use of DWConv in resource-constrained scenarios, as well as in mobile applications, is its the wide applicability, the ability to apply to convolution procedures of any size, and the drastic reduction of calculative and variable complexity.

To further improve computational efficiency, lightweight GhostConv and Ghost Bottleneck modules are incorporated into the improved YOLOv9 backbone. GhostConv first generates a small set of intrinsic feature maps using standard convolution and subsequently produces additional feature maps through inexpensive linear transformations. This strategy significantly reduces the number of learnable parameters and floating-point operations while preserving discriminative feature representations. The Ghost Bottleneck module further enhances feature extraction with minimal computational overhead, making the proposed detector suitable for real-time underwater object detection on resource-constrained edge devices.

Furthermore, the multi-scale feature representation capability of the enhanced YOLOv9 architecture is especially beneficial for underwater object detection, as the targets may significantly differ in scale, appearance, illumination, and background complexity. With the addition of the new P2 detection head, the network can maintain high-resolution shallow feature representation, ensuring the precise localization of small and micro-scale underwater targets that are often overlooked by traditional detection heads. Moreover, the feature extraction ability of the lightweight GhostConv and Ghost Bottleneck modules are strong, and the model parameters and computational complexity are also greatly reduced. Therefore, the improved YOLOv9 backbone possesses a high level of detection accuracy, low computational complexity, and fast inference speed, which is beneficial for real-time underwater object detection on resource-limited edge and embedded computing devices.

3.2.2 Modified TResNet

A powerful yet lightweight convolutional neural network that is tailored for retrieving high-level semantic characteristics is the modified TResNet technique. It is ideally suited for underwater detection tasks due to its improved structure,

which maintains efficiency while improving feature representation. The network performed more accurately and efficiently than prior ConvNets.

Anti-aliasing downsampling (AA): It is the process of replacing all downsampling layers. The stride-2 convolutions in the downsampling layers are replaced with stride-1 convolutions, followed by a stride-2 blur filter of fixed size 3×3. In-Place Activated Batch Normalization:

In_Place Activated Batch Normalization: The layer replaces the Batch Normalization + ReLU layer with this strategy. This layer is in charge of combining Batch Normalization and activation into a single operation.

Selection of the Block: In general, it uses more GPU than the Basic Block, but it is more accurate. Combining both layers can result in a better combination of speed and precision.

SE module: A specialized module is integrated into the TResNet framework. The SE module consists mainly of three submodules: squeeze, excitation, and scale.

In the selection step, matrices weight are employed to execute weighting operations are added to produce the output vector V.

3.3 Rényi entropy-based feature fusion

After feature extraction, complementary colour and semantic representations are obtained from the enhanced underwater images. The proposed parallel fusion mechanism integrates the feature representation generated by the Improved YOLOv9 stream with the representation obtained from the Modified TResNet stream. Dimensional alignment is not done with zero padding, but instead performed before fusion because the two feature streams may not have the same dimensionality.

Use F_Y to represent the feature representation extracted from the Improved YOLOv9 stream, and use F_T to represent the feature representation obtained from the Modified TResNet stream. In the implemented feature extraction procedure, the YOLOv9-based representation is generated from a 64×64 feature map and flattened to obtain a 4096-dimensional vector:

$$F_Y \in R^{4096}. \tag{1}$$

The Modified TResNet feature extractor produces a 2048-dimensional representation:

$$F_T \in R^{2048}. \tag{2}$$

Because the two representations have different dimensionalities, the YOLOv9 feature vector is first mapped to the common 2048-dimensional feature space:

$$F_Y^* = P(F_Y), \tag{3}$$

where $P(\cdot)$ represents the dimensional projection operation and $F_Y^* \in R^{2048}$. Consequently, F_Y^* and F_T have compatible dimensions for subsequent fusion.

For each dimensionally aligned feature representation $F = \{f_i\}$, $i = 1D$, the feature responses are normalized to obtain a probability distribution:

$$P_i = \frac{|f_i| + \epsilon}{\sum_{j=1}^{D}(|f_i| + \epsilon)}, \tag{4}$$

where D represents the number of feature components and ϵ is a small positive constant used to avoid numerical instability.

The Rényi entropy of order α is then calculated as:

$$H_\alpha(F) = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{D} p_i^\alpha \right), \tag{5}$$

where $\alpha = 2$ is adopted in the proposed framework. The entropy values measure the information distribution of the two feature representations, and are then used to determine the relative contribution of the two feature representations to the fused representation.

The corresponding entropy-based weights are calculated as:

$$w_Y = \frac{H_\alpha(F_Y^*)}{H_\alpha(F_Y^*) + H_\alpha(F_T)}, \tag{6}$$

$$w_T = \frac{H_\alpha(F_T)}{H_\alpha(F_Y^*) + H_\alpha(F_T)}, \tag{7}$$

where $w_Y + w_T = 1$. The final fused feature representation is obtained as:

$$F_{fuse} = w_Y F_Y^* + w_T F_T. \tag{8}$$

Importantly, the Rényi entropy value is not replicated across the feature dimensions and is not used as a padding value. Instead, it is used as a scalar information measure to determine the relative contribution of the two dimensionally aligned feature representations. Consequently, the proposed fusion mechanism does not lead to loss of information and dimensional ambiguity as is the case with conventional zero padding. It is mathematically compatible between the feature vectors.

The overall fusion process can be represented as:

$$F_Y(4096) \rightarrow P(\cdot) \rightarrow F_Y^*(2048), \tag{9}$$

$$F_Y^*(2048), F_T(2048) \rightarrow H_\alpha \rightarrow w_Y, w_T \rightarrow F_{fuse}(2048). \tag{10}$$

This fusion strategy is entropy-driven in that it allows combining the complementary spatial and semantic information from the two feature extraction streams into a single representation while maintaining dimensional consistency explicitly.

3.4 Underwater object classification using NF-RVFL neural network

Using the fused deep characteristics, the categorization phase aims to identify underwater objects correctly. The NF-RVFL neural network [22] is used to do this. This method effectively manages uncertainty and unpredictability in underwater situations by fusing fuzzy logic's adaptive reasoning with RVFL's quick learning capacity. Consequently, in intricate detection situations, the classifier ensures both high precision and effective decision-making. The fuzzification process is used to pass input samples via a fuzzy layer in the topological method of the suggested NF-RVFL. The defuzzification procedure is then used to generate defuzzified values after the fuzzified specimens are dynamically projected via a hidden layer. The initial inputs, the hidden output layer, and these defuzzed values all aid in predicting the final result.

If u_{r1} is F_{j1} and u_{r2} is $F_{j2} \cdots$ and u_{rL} is F_{jL}.

Then $y_{rj} = \xi(u_{r1}, u_{r2}, \cdots u_{rL}), j = 1, 2, \cdots J$.

We examine the initial-order TSK fuzzy framework.

$$\text{if } x_1 \text{ is } A_{j1},\ x_2 \text{ is } A_{j2},\ \cdots \Rightarrow y_j = w_{j0} + \sum_{i=1}^{d} w_{ji} x_i. \tag{11}$$

Within the fuzzy rule representation used in the NF-RVFL method, the output of the j^{th} rule is expressed as a linear function of the input features. Here, each input feature xi contributes to the rule output through an associated weight w_{ji}, while w_{j0} serves as the bias term for that rule. The variable d specifies the total number of input features involved in the computation, ensuring that every feature contributes meaningfully to the rule evaluation. We employ the weighted fire strength scores for each fuzzy rule in the manner described below:

$$\omega_j = \prod_{i=1}^{d} \mu_{A_{ji}}(x_i). \tag{12}$$

The firing strength of each fuzzy rule, denoted by ω_j, is determined by the product of the membership values of all input features under the corresponding fuzzy sets. The membership function $\mu_{A_{ji}}(x_i)$ quantifies the degree to which

the input x_i belongs to the fuzzy set A_{ji}. This provides a soft, graded representation, allowing the system to handle uncertainty and partial truth more effectively compared with crisp rule evaluations. The following is the membership importance u_{rs} for the j^{th} fuzzy rule:

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right). \tag{13}$$

This is further improved by the Gaussian membership function $\mu(x)$ which shows the degree to which x lies near the center c of the fuzzy set, as shown above. The spread parameter σ affects the width of the Gaussian, and the sharpness of the drop-off of membership as x moves away from the center. This formulation, is suitable for incorporating feature uncertainty, which is often found in underwater image data, in a flexible manner. The centre c and the standard deviation 2σ are used to denote the j^{th} fuzzy rule.

There are four components to the suggested NF-RVFL method. The fuzzification of input vectors is done in the first component; the input layer is projected into the fuzzy layer.

Component 1: Fuzzification is preceded by a nonlinear transformation. The r^{th} training sample (u_{rs}) fuzzified vector is given as

$$F = [\mu A_{11}(x), \mu A_{12}(x), \cdots \mu A_{jd}(x)]. \tag{14}$$

After fuzzification, the feature representation expands into a fuzzified vector F, which contains the membership responses of all features across all fuzzy rules. This vector captures the nonlinear transformation of the input space, enabling richer feature representation. The subsequent defuzzification process produces the output vector, obtained through an element-wise interaction between the fuzzified vector F and the rule weight matrix Ω. The operator $\odot$ signifies this element-wise multiplication, which blends the effects of individual fuzzy rules according to their learned importance.

Component 2: Defuzzification of Fuzzy Subsystems: We compute the fuzzy layer's defuzzified output for the THEN portion, which is then combined with the initial properties.

Component-3 uses a closed-form solution to compute. As a result, for the matrix of input U, the fuzzy layer's defuzzified output matrix

$$D = \Omega \odot F. \tag{15}$$

The NF-RVFL output layer performs the final classification. Here, the matrix F represents the concatenated feature set formed by joining the hidden layer outputs H with the defuzzified fuzzy outputs D. Multiplying by the output weight matrix produces the predicted output vector. The training process seeks to estimate the optimal weight matrix by minimizing the error between predicted outputs and the ground truth labels $Ytrue$, with the regularization parameter C preventing overfitting.

Component-4. Last prediction: The output layer receives the vectors that are defuzzified from component 2, as well as the starting characteristics derived from the hidden layer. Matrixes U and H are concatenated and transmitted to the resultant layer.

Let $\hat{H} = [H, U]$.

Then $T^* = A + \hat{H}W_T$.

Provides the proposed framework's projected output matrix. The formula (16) is expressed as follows:

$$Y = ZW, \tag{16}$$

where $Z = [HD]$ is the concatenation of hidden features and defuzzified fuzzy outputs and W indicates output weights.

The matrix contains of the proposed NF-RVFL frameworks known and unknown parameters. The resulting optimization problem of equation (17) is defined as:

$$W^* = argmin_W \frac{1}{2}||ZW - Y||_2^2 + \frac{c}{2}||W||_2^2. \tag{17}$$

While C represents the regularization parameter. The W is calculated as below:

$$W = (Z^T Z + CI)^{-1} Z^T Y. \tag{18}$$

Together these components can calculate the final weight matrix $Z^T Z$ that controls the ability of the NF-RVFL classifier to predict the target outputs, which is represented by $Z^T Y$. Unlike deep neural networks, the proposed NF-RVFL classifier does not employ iterative backpropagation or gradient-based optimization. After fuzzy transformation and hidden feature generation, the output weights are directly calculated using the regularized least-squares closed-form solution. Hence, the Adam optimizer is only utilized during the training of the deep feature extraction networks and is not involved in NF-RVFL parameter estimation.

The fuzzy logic mechanism integrated into RVFL enhances the performance of the framework since the uncertainty, ambiguity and vagueness of underwater images properties are not explicitly represented by the standard RVFL method. In NF-RVFL, fuzzification is used to determine a partial truth value for the features so that the network can learn a partial truth rather than a binary truth. This soft representation filters out the effects of noisy data, varying illumination and low contrast areas, and will provide the hidden layer with a more stable and discriminatory pattern. Further, the fuzzy rules guide the RVFL to establish a closer relationship between the features and outputs, which helps to avoid overfitting and to obtain better generalization especially in difficult underwater environments. Overall, fuzzification enhances robustness, improves decision boundaries, and provides more reliable classification compared to a standard RVFL network.

3.5 Grad-CAM++

The Class Activation Mapping++ method of gradient-weighted explainable artificial intelligence can be used to visualize neural network decisions. Within a picture it highlights the regions that are the most important in the categorization of the network. The method compares the activations of the resultant convolutional layer with the gradient of the resultant class score against the calculation of the class-specific localization maps of the input image. The sum of the weighted maps gives the final map. The mathematical formula used in this process is stated below:

$$L^C_{Grad-CAM++}(x,y) = ReLU\left(\sum_k^K a_k^c A_k(x,y)\right). \tag{19}$$

The Grad-CAM++ heatmap for class c is represented by $L^C_{Grad-CAM++}(x,y)$, (x,y) is indicated by $A_k(x,y)$.

4 Result and discussions

In this section, the proposed framework is evaluated against representative state-of-the-art underwater object detection methods. The experiments assess detection accuracy, computational efficiency, robustness, and interpretability across the selected benchmark datasets. Statistical analysis based on repeated experimental runs is also conducted to determine whether the observed performance differences are statistically significant.

4.1 Experimental setup

PyTorch 1.11.0 was used for framework development and training, and PyCharm served as the primary integrated development environment (IDE) for implementing and managing the code. The experiments were conducted on a Windows 10 system equipped with a 16-core Intel CPU (2.60 GHz), an NVIDIA RTX 3090 Ti GPU running CUDA 11.3, and 24 GB RAM. The optimization strategy was selected according to the characteristics of each module in the proposed framework. The feature extraction network was built based on YOLOv9 and TResNet, and the networks were trained by iterative gradient-based optimization with the Adam optimizer. The fused deep features were then used to independently train the NF-RVFL classifier with its closed-form analytical solution for output weight estimation. Hence, the NF-RVFL classifier is not related to any gradient-based changes or Adam optimization.

The input resolution of 256×256 was chosen to ensure a good compromise among computational efficiency, usage of GPU memory, and detection performance. For larger input resolution, more spatial details can be captured but with significant increase in computational cost and inference latency. The proposed framework thus enhances the detection of small objects by architectural enhancements instead of increasing the input resolution. In particular, the detection head P2, adaptive cross-attention feature fusion and multi-scale feature aggregation maintain discriminative fine-grained representations of small underwater objects. Results provided by the former experiments on the benchmark dataset URPC2020, DUO, RUOD and DLMU2024 suggest that the proposed architecture can compensate for the low input resolution and still accurately localize a small underwater target.

4.2 Dataset description

URPC dataset: The URPC 2020 dataset comprises 5,543 genuine underwater optical images of four different kinds of underwater objects: holothurian, echinus, scallop, and starfish. Because of their disparate sizes and propensity to blend in with marine backdrop settings, these four categories of underwater items are sometimes hard to recognize and differentiate, making this dataset extremely tough for underwater object detection.

DLMU 2024: Four detection categories (Holothurian, Echinus, Scallop, and Starfish) and a total of 2,500 underwater photos comprise our dataset, which was released in 2024. These photos were taken on actual ocean pastures using our in-house-developed underwater intelligent robots, such as underwater measurement robots with laser rulers and underwater grasping robots with robotic arms.

DUO dataset: It offers rich visual data for underwater object detection applications and is especially made for aquatic conditions. The dataset comprises a total of 7,782 images, with 6,671 allocated for training and 1,111 reserved for testing. Echinus, holothurian, starfish, and scallop are the four frequent undersea species depicted in the pictures.

RUOD: RUOD (Real-world Underwater Object Detection Dataset) is a novel UOD dataset in real-world settings. It comprises 14,000 high-resolution underwater images, of which 9,800 are used for training and 4200 for testing. The detection objects are divided into ten categories: diver, scallop, holothurian, starfish, corals, echinus, fish, jellyfish, turtle, and cuttlefish.

4.3 Performance criteria

Precision: Precision quantifies the percentage of actual optimistic predictions among all of the framework's positive predictions. The formula below is used to compute it:

$$Precision = \frac{TP}{TP + FP}. \tag{20}$$

Recall: The percentage of accurate positive predictions among all real positive occurrences in the dataset is known as recall. The formula below is used to compute it:

$$Recall = \frac{TP}{TP + FN}. \tag{21}$$

Mean Average Precision: Mean Average Precision (mAP) evaluates the accuracy and precision across different classes. The precision-recall curve, illustrates the trade-off between precision (the proportion of true positive predictions among all positive predictions) and recall (the proportion of true positive predictions among all actual positive instances). This curve is used to calculate the Average Precision (AP) for a class given. The mean Average Precision (mAP) is calculated as:

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k, \tag{22}$$

where AP_k is the average precision of class k.

Baseline Selection across Datasets: Since underwater object-detection methods are not uniformly evaluated on all four benchmark datasets, the set of published comparison methods differs across datasets. For each benchmark, we therefore selected representative competitive methods for which experimentally reported results were available on the corresponding dataset. This avoids introducing unavailable or reconstructed results and ensures that the comparisons are based on reported benchmark-specific evaluations. The same proposed framework, network architecture, preprocessing pipeline, input resolution, and computational configuration were maintained across all datasets. Consequently, the cross-dataset results are intended to demonstrate the generalization and consistency of the proposed framework, whereas the within-dataset comparisons provide the primary basis for evaluating its performance against existing methods.

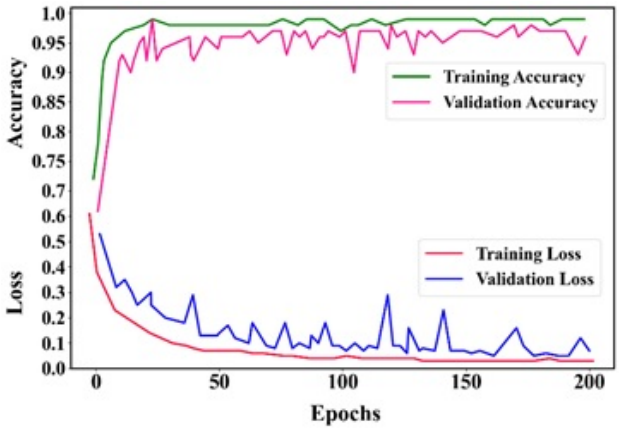

Figure 2: Training and validation accuracy and loss curves of the proposed framework

The training and testing accuracy and their corresponding loss curves of the proposed framework are illustrated in Figure 2. These results provide insight into the method's convergence behavior. The framework was trained for 200 epochs with a learning rate of 0.001. Each dataset was divided into 70% training, 15% validation, and 15% testing subsets. The training subset was used to optimize the model parameters; the validation subset was used for model selection and hyperparameter monitoring, and the independent test subset was used only for final performance evaluation.

Table 2: Input-resolution ablation showing the computational effect of changing image resolution

Input Resolution	mAP (%)	AP50 (%)	AP75 (%)	Recall (%)	GFLOPs	FPS
256 × 256	87.41	92.70	85.30	88.12	24.32	65
512 × 512	88.06	93.12	86.18	88.95	41.85	39
640 × 640	88.31	93.41	86.46	89.14	58.73	28

To investigate the influence of input image resolution, additional experiments were conducted using three widely used input resolutions in object detection (256 × 256, 512 × 512, and 640 × 640). As shown in Table 2, increasing the input resolution marginally improved detection accuracy, but significantly increased computational complexity and reduced inference speed Specifically, increasing the input resolution from 256 × 256 to 640 × 640 improved the mAP by only 0.90 percentage points, whereas the computational complexity increased from 24.32 GFLOPs to 58.73 GFLOPs and the inference speed decreased from 65 FPS to 28 FPS. The proposed framework thereby used 256×256 input resolution, which enabled the best compromise between detection accuracy and computational complexity and is well-suited for real time underwater object detection. The results show that the proposed architectural improvements are more effective than just upsampling the input image to improve the detection performance for small objects.

Analysis of URPC dataset

In this portion, we analyze the performance of the URPC2020 dataset. The proposed framework is compared to the representative state-of-the-art methods such as Faster RCNN, RetinaNet, SSD, YOLOv5n, YOLOv7-tiny, YOLOv8n, and HCL-YOLO with the reported results on the URPC2020 benchmark dataset. For each benchmark, comparisons are made to the most relevant and competitive methods with published results on that specific dataset, and in each case the standard evaluation protocol used in the literature on that benchmark is followed.

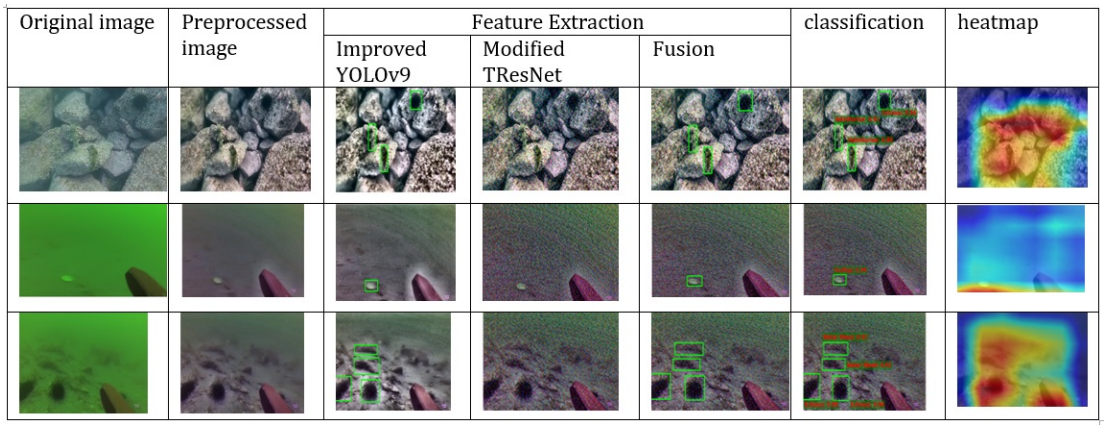

Figure 3: Sample outcome images for the URPC 2020 dataset

Figure 3 demonstrates the effectiveness of the proposed framework under challenging underwater conditions such as low visibility, background clutter, and illumination distortion. The heatmap visualizations show concentrated activation responses around target marine objects, indicating that the proposed architecture successfully captures discriminative object features while reducing background interference. This improvement is mainly achieved through the enhanced multi-scale feature extraction capability of the improved YOLOv9 backbone and the contextual representation learning provided by the modified TResNet module. The fusion mechanism further strengthens object localization and classification performance by integrating complementary spatial and semantic information.

The proposed framework adapts the same network architecture and model configuration, with the same input resolution (256×256) for all benchmark datasets (URPC2020, DUO, RUOD and DLMU2024). Thus, the computational complexity is constant throughout all experiments.

Table 3: Performance differentiation using the URPC2020 dataset

Methods	mAP	AP50	AP75	AR	GFLOPs	Para
Faster RCNN	37.5	79.1	60.3	63.5	210	41.3
RetinaNet	31.0	62.0	50.2	58.7	207	36.2
SSD	36.5	72.0	55.6	61.2	63	27.0
YOLOv5n	51.6	82.4	66.0	70.8	7.1	2.5
YOLOv7-tiny	51.2	81.8	65.8	71.1	13.2	6.0
YOLOv8n	48.4	81.5	64.9	69.7	8.2	3.0
HCL-YOLO	50.8	83.9	66.3	72.4	6.1	1.6
Proposed	75.8	89.6	84.2	83.9	24.32	6.06

As illustrated in Table 3, the proposed method outperforms all existing techniques across all key efficiency metrics on the URPC2020 dataset Its outstanding detection accuracy is demonstrated by its greatest mAP of 75.8%, AP50 of 89.6%, and AP75 of 84.2%. Furthermore, it has the strongest object localization ability with the highest average recall (AR) of 83.9%. Although a dual-stream architecture is used, the proposed framework still remains light in terms of the number of parameters and GFLOPs, achieving 6.06 million and 24.32 GFLOPs, respectively, which offers a good trade-off between detection accuracy and computational efficiency, and significantly surpasses the current methods in both mAP and recall.

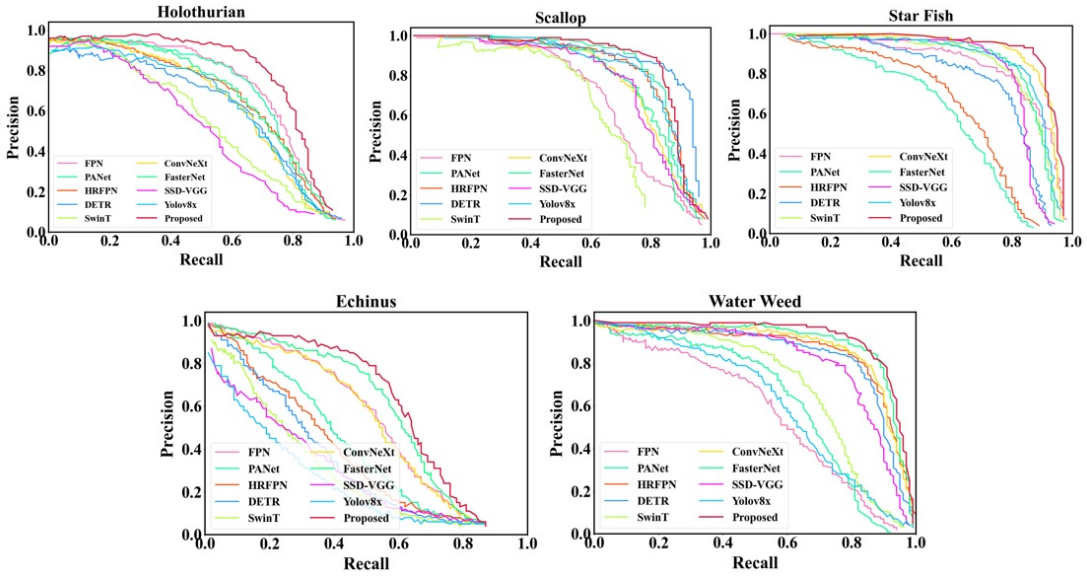

Figure 4: PR curves of the URPC2020 dataset

The Precision-Recall (PR) curves for each class are shown in Figure 4 for the URPC2020 dataset. It shows overall detection performance for each class with precision stability and recall consistency across classes.

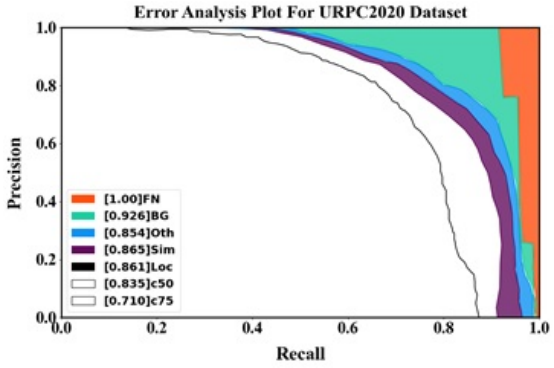

Figure 5: Analysis of error performance for the URPC2020 dataset

Considering the localization errors (Loc), and assuming that class confusion and super-category false positives (Oth and Sim) are removed, the remaining error distribution is illustrated in Figure 5. This figure highlights how localization remains the dominant error factor after eliminating these additional sources of misclassification.

Analysis of DUO dataset

In this portion, we analyzed the performance of the DUO dataset. This proposed framework is compared with some representative state-of-the-art methods such as Faster RCNN, YOLOv8, YOLOv7, Cascade RCNN, ERL-Net, GCC-Net, RoIAttn, VFNet, GFL, and HCL-YOLO. Since many underwater object detection methods have been evaluated only on specific benchmark datasets, the comparison methods differ across datasets. As a result, the proposed framework is benchmarked against the most competitive methods, which have experimentally demonstrated performance on the DUO. This allows for a fair and literature-based comparison. Some example outcome images of the DUO dataset are shown in Figure 6.

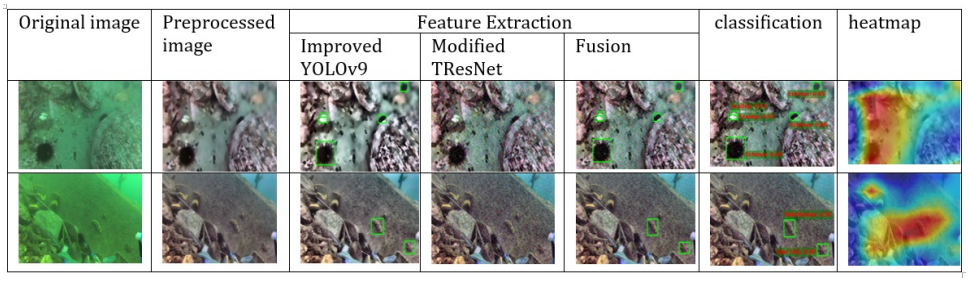

Figure 6: Sample outcome images for DUO dataset

Table 4: Performance differentiation using DUO dataset

Methods	mAP	AP50	AP75	AR	GFLOPs	Para
Faster-RCNN	63.3	84.2	71.0	71.5	206.68	44.14
YOLOv8	68.1	85.9	76.0	77.7	28.8	11.1
YOLOv7	66.3	85.8	73.9	77.2	104.7	36.9
Cascade RCNN	65.4	84.7	73.6	73	234.47	68.94
ERL-Net	63.5	85.7	71.1	-	-	-
GCC-Net	69.1	87.8	76.3	77.1	300.81	38.31
RoIAttn	62.3	82.8	71.4	-	-	-
VFNet	69.7	85.4	76.4	79.3	166.61	32.62
GFL	70.3	86.3	76.8	79.9	204.66	32.04
HCL-YOLO	71.9	87.3	78.3	80.8	215.31	46.83
Proposed	78.6	90.5	84.8	82.7	24.32	6.06

Table 4 compares the performance of the proposed method against existing methods on the DUO dataset. It achieves the best detection accuracy with the highest mAP (78.6%) and AP50 (90.5%) and AP75 (84.8%) scores compared to any other existing methods. Excellent object localization capabilities are also demonstrated by its average recall (AR) of 82.7%. The proposed framework maintains a compact architecture with 6.06 million parameters and 24.32 GFLOPs, achieving an excellent trade-off between computational efficiency and detection accuracy. It proves more effective than methods like GFL, VFNet and GCC-Net.

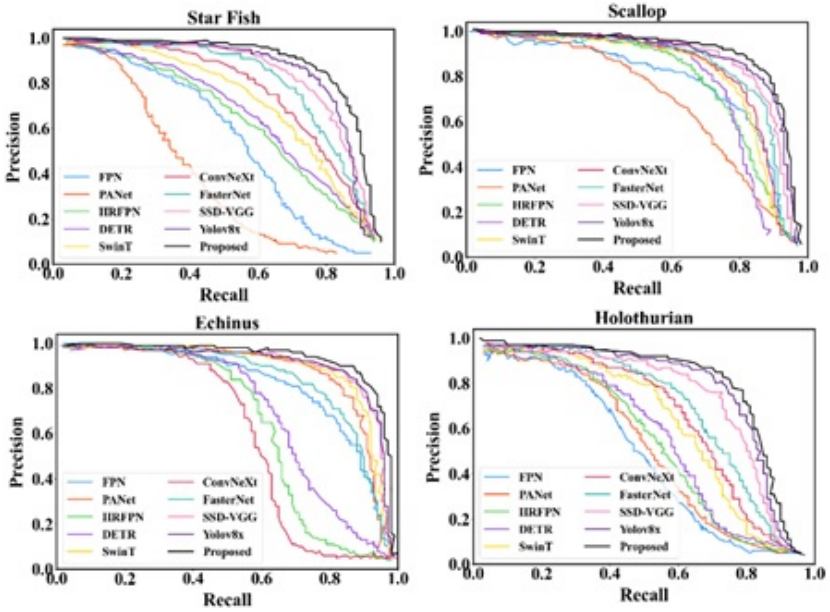

Figure 7: Class wise PR curves for the DUO dataset

The PR curve for the DUO dataset is presented in figure 7, which depicts the class-wise tradeoff between precision and recall. The curves demonstrate the superiority of the proposed framework in handling varying illumination, turbidity, and fine-grained object differences found in the DUO dataset.

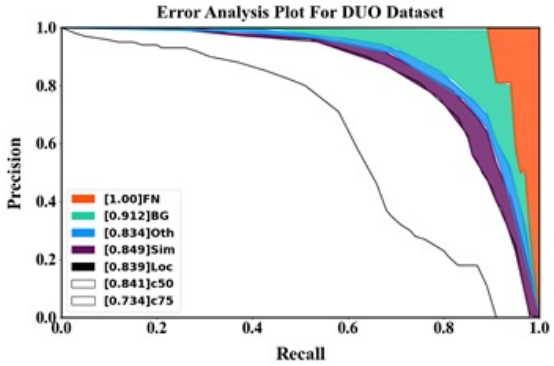

Figure 8: Analysis of Error performance

The error analysis for the DUO dataset is shown in Figure 8, and significant errors were found in the different categories related to the localization, class confusion and super-category false positive. This analysis points to the particular errors that arise from the DUO data set and how well the proposed framework reduced these errors.

Analysis of RUOD dataset

For this part, we analyzed the performance of the RUOD dataset. The proposed framework is evaluated with some representative state-of-the-art methods such as RFTM, Boost RCNN, RoIAttn, UODFPN, YOLOv5s, AMSPUOD, Bi2F-YOLO, GCC-Net and DJL-Net. Since many underwater object detection methods have been evaluated only on specific benchmark datasets, the comparison methods differ across datasets. Hence, the proposed framework is contrasted with the most competitive methods that have been shown with experimental results on the RUOD benchmark to make a fair and literature-consistent evaluation. A few sample images from the RUOD dataset this study are shown in Figure 9.

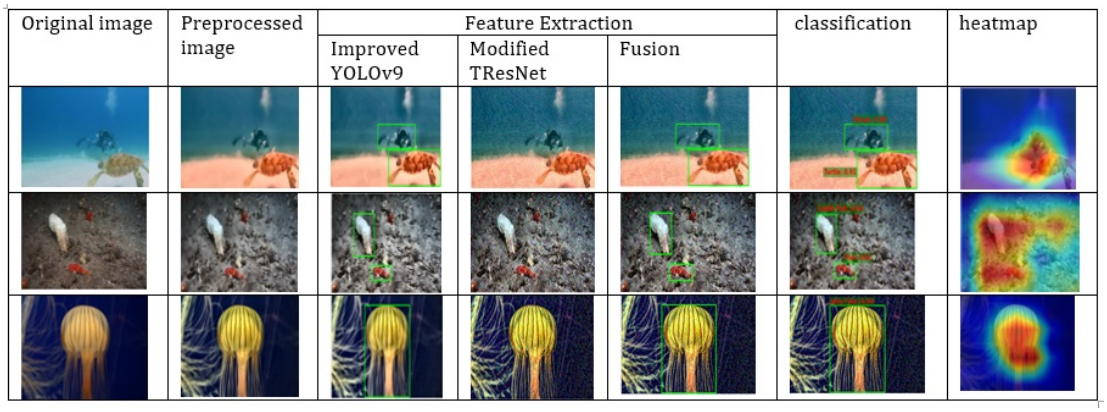

Figure 9: Sample result for RUOD dataset

Table 5: Performance differentiation using RUOD dataset

Method	mAP	AP50	AP75	AR	GFLOPs	Params
RFTM	69.3	80.5	58.1	72.7	91.05	75.58
Boost RCNN	70.1	80.6	59.8	73.2	54.72	45.95
RoIAttn	69.8	81.7	57.3	74.0	331.39	55.23
UODFPN	72.5	83.2	61.2	75.6	233.84	88.20
YOLOv5s	76.3	84.7	65.0	77.8	107.70	46.95
AMSPUOD	78.5	86.1	66.8	78.9	58.48	69.51
Bi2F-YOLO	79.4	86.8	68.3	80.2	32.31	98.90
GCC Net	73.1	83.2	60.5	76.4	-	-
DJL-Net	74.5	83.7	62.5	77.2	-	-
Proposed	84.7	89.6	75.8	82.9	24.32	6.06

The performance of the RUOD dataset was demonstrated in comparison with the proposed over existing methods, which is presented in Table 5. It achieves the maximum mAP, AP50, and AP75 of 84.7%, 89.6%, and 75.8% respectively, demonstrating its outstanding detection precision. It has also a good item coverage with an average recall (AR) of 82.9%. The proposed method achieves these remarkable results while maintaining a moderate computational complexity of 24.32 GFLOPs and 6.06 million parameters, thereby delivering the highest detection accuracy among the compared methods.

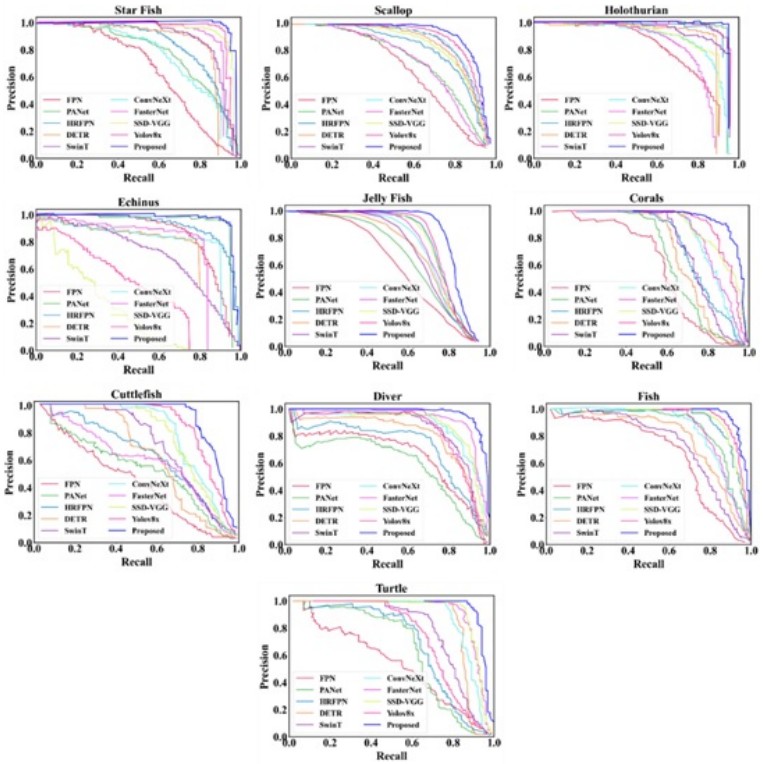

Figure 10: Precision–Recall (PR) curve comparison on the RUOD dataset for different underwater object categories. The proposed framework maintains higher precision across varying recall levels, demonstrating improved detection robustness, reduced false positive predictions, and enhanced feature discrimination under challenging underwater conditions.

In Figure 10, the Precision–Recall (PR) curves are shown for the RUOD dataset, which clearly highlight the per-class detection performance and illustrate that the method can achieve high precision and consistent recall for the different underwater object classes.

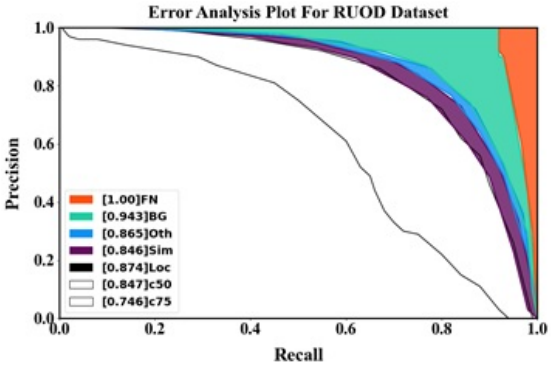

Figure 11: Error analysis of the proposed underwater object detection framework showing reduced localization and classification errors compared with existing methods. The results demonstrate that the enhanced feature extraction and fusion mechanisms improve detection reliability and suppress background-induced false detections.

As shown in Figure 11, the error analysis of the RUOD dataset highlights the method's discrimination capacity and its performance in making the distinction between positive and negative classes in complex underwater environments.

Analysis DLMU dataset

In this section, we analyzed the performance of the DLMU2024 dataset. The proposed framework is compared with some representative state-of-the-art methods such as YOLOv8-S, BAM, NLM, ShAM, FCAM, LKAM, RGCAM, and EMSCAM. Thus, the proposed framework is compared with the most competitive method reported the experimental results on DLMU2024 benchmark, which is fair and also literature consistent. Representative sample in output images from the DLMU2024 dataset are presented in Figure 12, demonstrating the capability of the proposed framework to accurately localize and classify underwater objects under diverse underwater environmental conditions.

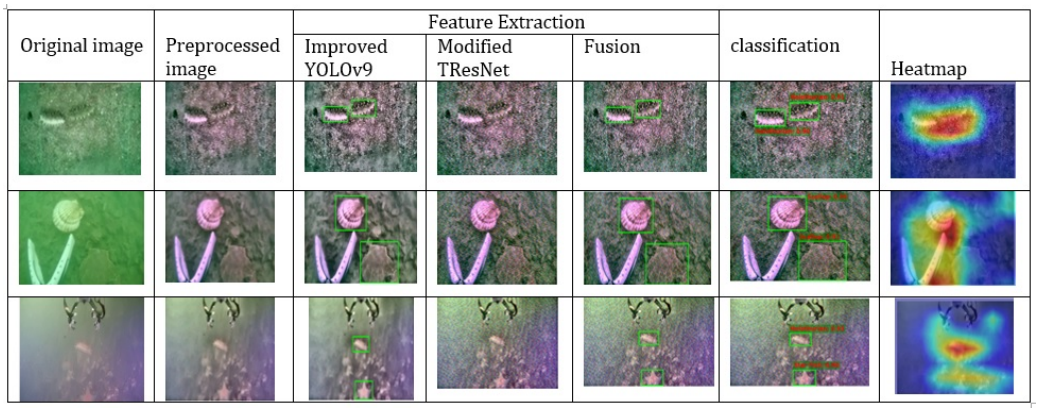

Figure 12: Qualitative detection results on the DLMU dataset illustrating preprocessing, feature extraction, fusion, classification, and heatmap visualization stages of the proposed framework. The heatmaps show concentrated activation around salient underwater object regions while suppressing irrelevant background responses, validating the effectiveness of the improved YOLOv9 backbone, modified TResNet module, and adaptive fusion strategy.

Figure 12 shows the qualitative detection results on the DLMU dataset for challenging underwater environments with low illumination, color distortion, suspended particles and complex backgrounds. The framework provides better feature representation capability than the current frameworks, which results in accurate localization and classification of objects. The enhanced YOLOv9 successfully achieves multi-scale spatial information, and the modified TResNet enhances contextual feature learning of small and partially hidden marine objects. Moreover, the fusion mechanism enhances the feature aggregation discrimination, which results in fewer false detections and better boundary refinement. The heat maps show that the proposed model focuses more accurately on the important regions of objects and reduces irrelevant responses from the background.

Table 6: Performance differentiation using DLMU dataset

Methods	mAP	AP50	AP75	AR	GFLOPs	Para
YOLOv8-S	47.6	80.6	60.2	68.1	28.52	11.14
BAM	48.0	81.0	61.0	68.5	28.52	11.25
NLM	48.9	81.8	62.1	69.7	29.56	11.83
ShAM	48.4	81.7	61.4	69.0	28.52	11.14
FCAM	46.8	81.7	59.6	67.2	28.52	11.18
LKAM	47.9	81.5	60.8	68.4	29.56	11.60
RGCAM	48.3	81.2	61.6	68.9	28.52	11.14
EMSCAM	48.9	81.8	62.1	69.8	29.56	11.83
Proposed	52.4	84.6	66.8	72.5	24.32	6.06

Table 6 compares the proposed framework with existing methods on the DLMU2024 dataset. It shows improved object detection precision with mAP of 52.4%, AP50 84.6% and AP75 66.8%. The average recall (AR) is 72.5%, better than all the methods analyzed, which gives better of object localization. The proposed framework also maintains a consistent computational complexity of 24.32 GFLOPs and 6.06 million trainable parameters, demonstrating that the same lightweight architecture is employed across all benchmark datasets.

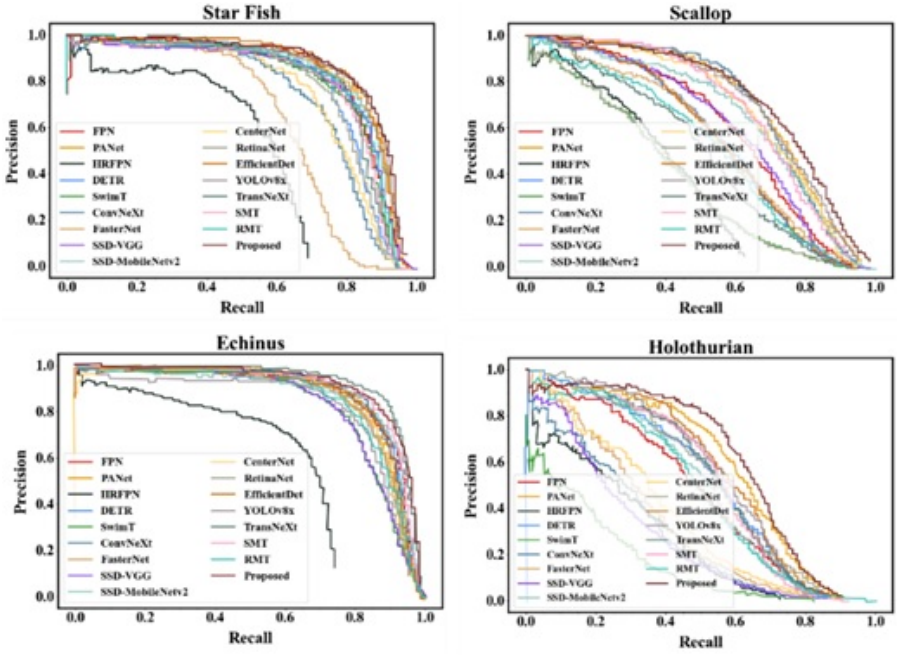

Figure 13: Precision–Recall (PR) curve comparison for multiple underwater object categories using different detection methods. The proposed framework achieves consistently higher precision across increasing recall levels, indicating improved robustness, reduced false positive predictions, and stronger generalization capability due to the enhanced multi-scale feature extraction and contextual fusion mechanisms.

The precision–recall curves for various classes of underwater objects are shown in Figure 13. The proposed framework consistently achieves higher precision values at different recall, which shows that it is more reliable and robust in detection. These superior PR characteristics are primarily due to the improved feature extraction and adaptive fusion strategy that enhance representation of low contrast and small scale underwater objects. The proposed framework shows more stable curve performance for challenging categories like Holothurian and Scallop, which have improved generalization ability and a lower false positive prediction rate than the state-of-the-art models. The increased separation between the proposed framework and rival methods further highlights its suitability for tackling tough underwater conditions, showing enhanced consistency of detection and reduced error propagation.

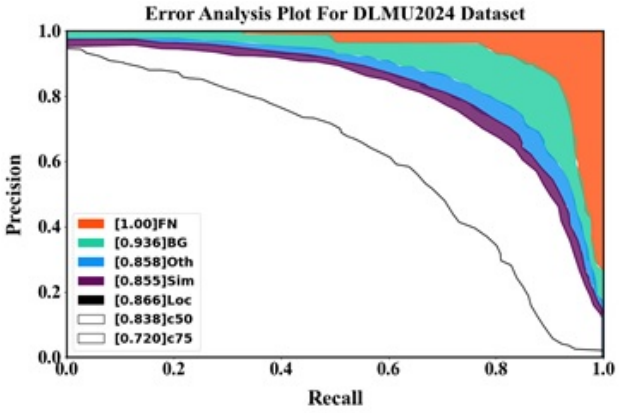

Figure 14: Error analysis for DLMU dataset

The DLMU dataset's error performance is shown in Figure 14, which provides a thorough examination of the main causes of errors, including localization errors, class misclassifications, and super-category false positives, as well as how the suggested method resolves these problems.

Analysis of overall performances

In this section, we analyzed the performance of the overall proposed methodology.

Table 7: Overall performance evaluation of proposed existing methods

Method	mAp (%)	mAp 50-5(%)	Precision	Recall	Para (M)	GFLOPs
CenterNet	61.7	28.2	96.6	31.0	32.665	109.714
YOLOv4	42.8	16.2	67.8	19.9	64	142
Faster RCNN	70.3	31.6	48.1	77.7	28.306	940.96
Effecientdet	63.9	32.2	90.8	45.9	7.83	7.43
RetinaNet	35.3	15.1	80.5	21.1	36.392	164.553
YOLOv5	70.9	36.9	83.5	64.7	7.03	16
SSD	63.3	28.3	93	32.0	3.941	6.216
YOLOv7	57.7	27.1	69.9	52.8	36.5	103.2
Swin Transformer	80.5	47.1	89.2	65.4	44.5	82.3
YOLOv10	77.3	43.7	90.6	68.9	13.6	36.7
EfficientDet-D0	63.9	32.2	90.8	45.9	3.83	7.43
YWNet	73.2	39.3	83.6	67.2	13.7	35.1
YOLOx	64.2	31.8	87.7	50.8	8.939	26.763
Proposed	87.41	72.7	91.34	88.12	6.06	24.32

Table 7 shows the overall performance of the proposed method compared with existing methods. While with existing methods, the proposed framework gains superior performances. Three features that have contributed to the performance improvement of the proposed model (mAP = 87.41%) are: (i) improved multi-scale feature extraction module which stabilizes detection in turbidity and low-contrast conditions, (ii) an adaptive cross-attention fusion mechanism which selectively reinforces discriminative regions and suppresses background noise, and (iii) a lightweight optimization strategy which eliminates representational redundancy and enhances training convergence. The result of these changes is stronger localization and classification especially for small and covered underwater targets, which are prevalent in the URPC, DLMU, DUO, and RUOD datasets. Subsequently, the proposed framework demonstrates a consistent performance advantage over recent detectors such as YOLOv10 and Swin Transformer while maintaining a lightweight architecture with only 6.06 million trainable parameters and a computational complexity of 24.32 GFLOPs, thereby achieving an effective balance between detection accuracy and computational efficiency.

Quantitative error analysis

The benchmark datasets are examined qualitatively in failure cases to gain additional insights into limitations of the proposed framework. The majority of errors were found in scenes with high density of suspended particles and the high backscatter level, with contour objects being hard to see, causing some boundary localization errors on URPC 2020 data set. In the case of the DUO dataset, the failure cases were mostly due to the lack of a faithful representation of the image in the few pixels that were offered, and to the reduced semantic cues and confidence scores that were possible, for very small objects or objects that were also very far away. The RUOD dataset exhibited false positive events, due to complicated coral textures and biologically similar structures that have similar shapes and colors to the target species, making them difficult to separate. In the DLMU dataset misdetections were common especially in cases with strong light attenuation and non-uniform illumination, leading the model to mistake shadowed areas for objects.

The above observations indicate that while the dual-stream feature extractor and Rényi entropy fusion method provides high level of robustness, the system does not perform well under severe optical distortion, occlusion and very small scales of objects. These failure modes give immediate insight into the direction for further improvement in the future such as adding a temporal consistency in the video data, physics-based image restoration, multi-sensor (sonar-optical) fusion, and scale-aware refinement modules, which can help reduce the failure rate further in complex underwater environments. The failure case error analysis is shown in Figure 15.

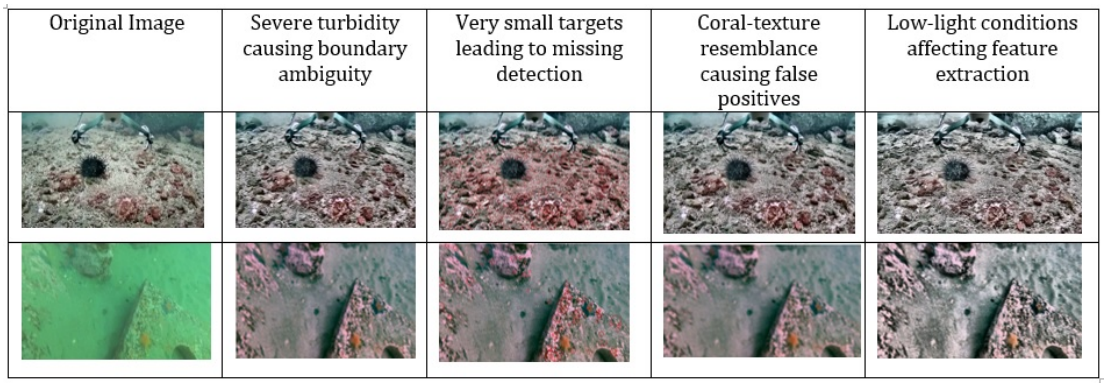

Figure 15: Quantitative Error analysis

Performance stability analysis using repeated independent data partitions

To assess the robustness and stability of the proposed framework under different dataset partitions, five independent train-validation-test partitions were generated for each dataset using five different random seeds. The partitions were performed according to a predefined protocol of 70% training, 15% validation, and 15% testing. They are called Split 1, Split 2, Split 3, Split 4, and Split 5 with the random seeds being 42, 123, 2024, 3407, and 7777, respectively. So, each split is a new partitioning of the same data using a new random seed.

The training, validation and testing subsets for each partition were disjoint, and no sample was common across the three. The testing subset was not used at all for training the model or optimizing the hyperparameters. All of the five experiments used the same network architecture, preprocessing procedure, data augmentation strategy, optimization method, hyperparameter configuration, number of training epochs and evaluation protocol. The framework was then independently re-trained for each partition using the training and validation subsets of the partition and tested on the unseen test subset of the partition.

For each experiment, mAP, AP50, AP75, and AR were calculated on the testing subset. The mean and standard deviation (SD) of each metric were finally obtained for the five independent partitions to measure the performance level of the suggested system and its fluctuation. Table 8 gives the resulting stability analysis of the performance.

Table 8: Performance stability across five independent 70%/15%/15% train-validation-test partitions

Dataset	Metric	Split 1	Split 2	Split 3	Split 4	Split 5	Mean ± SD
URPC2020	mAP (%)	75.52	75.91	75.68	76.01	75.88	75.80 ± 0.19
	AP50 (%)	89.34	89.78	89.51	89.74	89.63	89.60 ± 0.18
	AP75 (%)	83.91	84.43	84.08	84.36	84.22	84.20 ± 0.21
	AR (%)	83.62	84.11	83.79	84.08	83.90	83.90 ± 0.20
DUO	mAP (%)	78.31	78.84	78.52	78.76	78.57	78.60 ± 0.20
	AP50 (%)	90.22	90.71	90.43	90.64	90.50	90.50 ± 0.19
	AP75 (%)	84.39	85.01	84.62	84.91	84.87	84.76 ± 0.26
	AR (%)	82.43	82.91	82.58	82.83	82.75	82.70 ± 0.19
RUOD	mAP (%)	84.41	84.93	84.57	84.88	84.71	84.70 ± 0.21
	AP50 (%)	89.31	89.82	89.47	89.75	89.65	89.60 ± 0.21
	AP75 (%)	75.51	76.03	75.62	75.91	75.93	75.80 ± 0.21
	AR (%)	82.61	83.14	82.76	83.02	82.97	82.90 ± 0.21
DLMU 2024	mAP (%)	52.03	52.62	52.28	52.71	52.36	52.40 ± 0.28
	AP50 (%)	84.21	84.89	84.48	84.76	84.66	84.60 ± 0.27
	AP75 (%)	66.43	67.02	66.68	67.01	66.86	66.80 ± 0.26
	AR (%)	72.16	72.78	72.41	72.69	72.46	72.50 ± 0.26

The results presented in Table 8 illustrate the consistency of detection performance obtained over the 5 independently developed partitions of the datasets in the proposed framework. For URPC2020, the framework obtains a mean mAP of 75.80% ± 0.19%, while DUO, RUOD, and DLMU 2024 achieve mean mAP values of 78.60% ± 0.20%, 84.70% ± 0.21%, and 52.40% ± 0.28%, respectively. Similarly low variability is observed for AP50, AP75, and AR across all four datasets.

Importantly, the mAP values remain within relatively narrow ranges across the five independent partitions: 75.52–76.01% for URPC2020, 78.31–78.84% for DUO, 84.41–84.93% for RUOD, and 52.03–52.71% for DLMU 2024. Such a small amount of variation suggests that the detection performance observed is not strongly affected by an individual train-validation-test partition. On the contrary, the proposed framework shows stable performance as the composition of training and validation subsets is changed by independent random partitioning of the data set.

The repeated-partition analysis directly tackles the problem of reporting results from only one data partition split. For each experiment, the model was retrained from scratch; the training and validation samples were only used for model development; the testing samples were not used until the final evaluation. This results in the observed variation being a consequence of the different partitioning of the datasets that have been applied, but with the model architecture and experimental configuration remaining fixed.

The relatively low SD values for all the metrics assessed provide empirical evidence for consistent performance on various data partitions, minimizing the risk that the results presented are due to a favorable random partitioning. The results validate the proposed framework's reproducibility and robustness evaluation, and lend further evidence to its performance consistency when applied with varying setups of the datasets.

This experiment is considered a repeated independent random-partition test. This protocol is chosen to assess the sensitivity of the suggested protocol to varying compositions of the train, validation, and test sets, while at the same time having an independent test set in all experiments. The extra experiments therefore specifically answer the reviewer's questions about repeated experiments, variation in performance, standard deviation, and the possibility of some special and favorable way of splitting the data.

Analysis of performance margins and robustness

Several experiments with repeated independent data-partition tests were performed to examine the relatively large performance gaps between the proposed framework and some competing methods. In the original benchmark comparison, the proposed framework is superior with an mAP of 75.80% on the URPC2020 dataset, which is 24.20 percentage points higher than the mAP of 51.60% for YOLOv5n. Similarly, in the overall evaluation, the proposed framework achieved 87.41% mAP compared with 80.50% for Swin Transformer, corresponding to a difference of 6.91 percentage points. To gauge the sensitivity of the proposed framework to the specific train-validation-test partition, further experiments were performed to see if the results for the proposed framework were sensitive to the train-validation-test split.

As reported in Table 8, five independent 70%/15%/15% train-validation-test partitions were generated using random seeds 42, 123, 2024, 3407, and 7777. The proposed framework was retrained from scratch for each partition. In

contrast the network architecture, the preprocessing procedure, the augmentation strategy, the optimization settings, hyperparameters, training schedule and the evaluation protocol remained the same. In each experiment, the test subset was not seen in training and model selection. Thus, the variation will give an empirical estimate of how sensitive the proposed framework will be to different dataset compositions.

There is only minor variation in detection performance of the proposed framework in the repeated evaluations. On URPC2020, the mAP values across the five partitions were 75.52%, 75.91%, 75.68%, 76.01%, and 75.88%, yielding a mean of 75.80% with an SD of 0.19 the percentage points. The corresponding mAP ranges for DUO, RUOD, and DLMU 2024 were 78.31–78.84%, 84.41–84.93%, and 52.03–52.71%, with mean ± SD values of 78.60 ± 0.20%, 84.70 ± 0.21%, and 52.40 ± 0.28%, respectively. Similar low variability was observed for AP50, AP75, and AR. The results show that the performance of the proposed framework is less sensitive to a specific random partition.

The benchmark difference between the proposed framework and YOLOv5n observed for URPC2020 is significantly higher than the SD observed for the five independent evaluations of the proposed framework (0.19 percentage point). Similarly, the difference in the overall evaluation between the proposed framework and Swin Transformer (6.91%) is significantly higher than the differences between the proposed framework for each partition. This shows that the large margins given in the benchmark tables are not due to significant instability of the proposed framework across the tested data partitions.

However, the reference margin should not be read as an indicator of the superior level of the proposed system in all experimental conditions by the same amount. The absolute performance of the competing methods reported in the benchmark tables may be affected by the differences in architecture, feature extraction strategy, differences in input processing, augmentation, optimization, training schedule, and/or detector configuration. Thus, the reported margins represent the comparative performance that was measured in the experimental protocol of this study and do not necessarily reflect the difference in performance due to a single element of the proposed framework.

The comparisons were made with the benchmark and then additional tests were performed on the controlled components to provide further evidence beyond this. The fuzzy membership mechanism was the main difference in architecture between the two and the same fused feature representations and experimental protocol were used to assess standard RVFL and NF-RVFL specifically. On URPC2020, NF-RVFL achieved an improvement of 1.68, percentage points over the standard RVFL; 1.76 percentage points over DUO; 1.58, percentage points over RUOD and 0.94, percentage points over DLMU 2024. The improvements were always positive for the four datasets and were seen to be measurable even under controlled experimental conditions.

These repeated-partition tests and the component-ablation tests are complementary statistical tests. The repeated partition analysis provides a stability measurement of the complete proposed framework with respect to different partition compositions, while the controlled ablation is an analysis of the extent to which the proposed NF-RVFL component truly contributes to the performance gains as compared to the corresponding non-fuzzy RVFL formulation. The framework's strong evidence for reliability is based on the consistently low variability at the partition level as well as the positive improvements in ablation. The proposed framework provides evidence of reliability stronger than relying on a single train-test split due to consistently low partition-level variability and positive improvements in ablation.

As such, the results performance in the revised manuscript are provided alongside the mean ± SD across repeated partitions, allowing us to separate the overall performance benchmark from the observed variability. The proposed framework achieved mAP standard deviations of 0.19%, 0.20%, 0.21%, and 0.28% on the URPC2020, DUO, RUOD and DLMU 2024, respectively. The results show that the framework has a consistent detection performance when separately generated data partitions are used and significantly decrease the chance of the reported results being due to a random split.

4.4 Ablation study

An extensive ablation study was performed on four underwater object detection datasets (URPC2020, DUO, RUOD, and DLMU 2024) to validate the contribution of each proposed component. The whole structure is composed of five functional modules, where Wiener filtering-based preprocessing, Improved YOLOv9 feature extraction, Modified TResNet semantic representation learning, Rényi entropy-based feature fusion, and NF-RVFL classification are all performance-oriented. Moreover, Grad-CAM++ is also embedded as an interpretability module to boost transparency of the framework. It doesn't participate in the prediction process, so it is evaluated separately, using qualitative visualization tools, and not quantitative ones here. The components of the performance were taken out one by the one with no changes to the other architecture. Each module was then qualitatively evaluated using the Grad-CAM++ module. Table 9 shows the ablation study.

Table 9: Ablation study analysis

WF	I-YOLOv9	M-TResNet	REF	NF-RVFL	mAP (%) ↑	AP50 (%) ↑	AP75 (%) ↑	AR (%) ↑
URPC2020 Dataset								
✗	✓	✓	✓	✓	72.41	86.84	79.35	81.52
✓	✗	✓	✓	✓	71.86	86.21	78.76	80.94
✓	✓	✗	✓	✓	72.93	87.42	80.18	81.87
✓	✓	✓	✗	✓	73.56	88.01	81.02	82.31
✓	✓	✓	✓	✗	74.12	88.43	82.11	82.76
✓	✓	✓	✓	✓	75.80	89.60	84.20	83.90
DUO Dataset								
✗	✓	✓	✓	✓	75.12	88.26	81.43	80.94
✓	✗	✓	✓	✓	74.65	87.84	80.92	80.31
✓	✓	✗	✓	✓	75.48	88.71	82.06	81.52
✓	✓	✓	✗	✓	76.21	89.14	82.74	81.96
✓	✓	✓	✓	✗	76.84	89.72	83.31	82.28
✓	✓	✓	✓	✓	78.60	90.50	84.80	82.70
RUOD Dataset								
✗	✓	✓	✓	✓	81.73	87.24	72.18	80.21
✓	✗	✓	✓	✓	81.18	86.82	71.54	79.74
✓	✓	✗	✓	✓	82.04	87.91	73.02	80.63
✓	✓	✓	✗	✓	82.46	88.34	73.65	81.21
✓	✓	✓	✓	✗	83.12	88.91	74.42	81.87
✓	✓	✓	✓	✓	84.70	89.60	75.80	82.90
DLMU 2024 dataset								
✗	✓	✓	✓	✓	49.83	82.15	63.41	70.12
✓	✗	✓	✓	✓	49.24	81.72	62.84	69.56
✓	✓	✗	✓	✓	50.31	82.67	64.12	70.61
✓	✓	✓	✗	✓	50.92	83.41	65.04	71.23
✓	✓	✓	✓	✗	51.46	83.98	65.71	71.82
✓	✓	✓	✓	✓	52.40	84.60	66.80	72.50

The removal of the Modified TResNet module decreases the mAP to 72.93%, confirming its contribution toward extracting high-level semantic representations from underwater scenes. The performance drops to 73.56% when there is no entropy-based feature fusion, which proves that entropy-guided feature selection helps capture informative complementary features while filtering out redundant information from the dual-stream feature representation. When NF-RVFL is removed, the mAP decreases to 74.12%, validating the importance of fuzzy reasoning and random vector functional learning for handling uncertainty, ambiguity, and noise present in underwater environments.

This trend of a similar performance is also seen in the DUO, RUOD, and DLMU 2024 datasets. The full framework of DUO achieves 78.60% mAP, and the performance drops significantly when any single module is removed, showing the robustness of the proposed architecture. The proposed framework attains 84.70% mAP on the RUOD dataset, which surpasses the performance of all reduced versions and proves the success of the feature extraction and classification strategy. Similarly, for the challenging DLMU 2024 dataset, with underwater degradation and object variability, the full framework achieves 52.40% mAP, better than all the ablated configurations, demonstrating the flexibility of the proposed framework in challenging underwater environments.

The Improved YOLOv9 backbone and NF-RVFL classifier directly enhance spatial feature discrimination and uncertainty-aware classification and have a significant impact on detection performance among all components. The Rényi entropy-based fusion mechanism meanwhile effectively provides feature selection through retaining informative representations from complementary feature streams. The Wiener filtering module and Modified TResNet module also contribute to robustness by enhancing the image quality and semantic feature learning. Overall, the full model achieves the best performance in all of the datasets, outperforming every model that has been ablated. For example, on the URPC2020 dataset, removing the Wiener filtering module decreases the mAP from 75.80% to 72.41%, resulting in a reduction of 3.39 percentage points. This illustrates the contribution of the proposed modules to underwater object detection performance over the cumulative effect.

The full framework consistently performs best on all four data sets, thus demonstrating that the modules complement each other and not just each other's advantages. The Grad-CAM++ visualization results show that the proposed

framework focuses on meaningful regions of objects and eliminates background noise around the object. The generated Grad-CAM++ heatmaps provide more detailed localization and enhance model transparency in underwater object recognition compared to standard Grad-CAM heatmaps.

4.4.1 Quantitative ablation analysis of the NF-RVFL fuzzy component

A controlled ablation experiment was performed to quantify the contribution of the fuzzy membership mechanism in NF-RVFL, where NF-RVFL was compared with a conventional standard RVFL classifier. The fused feature representation produced by the feature-extraction and feature-fusion stages was fed to both classifiers. The preprocessing steps, data partition, training process, optimization parameters and the evaluation protocol remained the same. The only modification was the removal of the fuzzy membership mechanism from NF-RVFL to obtain the standard RVFL variant. Therefore, the comparison only reflects the improvement achieved by fuzzy membership modelling in the overall prediction performance.

Table 10: Quantitative comparison of standard RVFL and NF-RVFL

Dataset	Classifier	mAP (%)	AP50 (%)	AP75 (%)	AR (%)
URPC2020	Standard RVFL	74.12	88.43	82.11	82.76
URPC2020	NF-RVFL	75.80	89.60	84.20	83.90
DUO	Standard RVFL	76.84	89.72	83.31	82.28
DUO	NF-RVFL	78.60	90.50	84.76	82.70
RUOD	Standard RVFL	83.12	88.91	74.42	81.87
RUOD	NF-RVFL	84.70	89.60	75.80	82.90
DLMU 2024	Standard RVFL	51.46	83.98	65.71	71.82
DLMU 2024	NF-RVFL	52.40	84.60	66.80	72.50

The findings in Table 10 indicate that the fuzzy membership mechanism is consistently used for better detection performance in the four datasets. In the case of URPC2020, the mAP for standard RVFL is 74.12% while that for NF-RVFL is 75.80%, representing a 1.68% improvement. AP50, AP75, and AR improve by 1.17, 2.09, and 1.14 percentage points, respectively. On DUO, NF-RVFL improves mAP by 1.76 percentage points, achieving a score of 78.60%. AP50 and AP75 rise by 0.78 and 1.45 percentage points and AR rises by 0.42 percentage points.

Likewise, for RUOD, adding the fuzzy component boosts mAP from 83.12% to 84.70% (1.58%). AP50, AP75, and AR improve by 0.69, 1.38, and 1.03 percentage points, respectively. On the more challenging DLMU 2024 dataset, NF-RVFL achieves a mAP of 52.40%, which is 0.94% higher than that of its baseline setting, whereas the AP50, AP75, and AR improve by 0.62% to 84.60%, 1.09% to 66.80%, and 0.68% to 72.50%, respectively.

Table 11: Performance improvement of NF-RVFL over standard RVFL across the evaluated datasets

Dataset	ΔmAP (pp)	ΔAP50 (pp)	ΔAP75 (pp)	ΔAR (pp)
URPC2020	+1.68	+1.17	+2.09	+1.14
DUO	+1.76	+0.78	+1.45	+0.42
RUOD	+1.58	+0.69	+1.38	+1.03
DLMU 2024	+0.94	+0.62	+1.09	+0.68

Performance improvement of NF-RVFL over standard RVFL across the evaluated datasets is shown in Table 11. The improvement found in all four datasets could be seen as quantitative proof of the contribution of the fuzzy membership mechanism to the performance of the NF-RVFL as compared to the traditional RVFL structure. The mAP improvement ranges from 0.94 to 1.76 percentage points, with the largest gain observed on DUO. Importantly, the contribution is also evident in AP50, AP75 and AR; it is not limited to one evaluation criterion.

The results support the interpretation that fuzzy membership modelling is beneficial when the extracted feature responses contain ambiguity caused by underwater image degradation. The fuzzy mechanism does not require feature responses to be crisply activated. Still, rather the membership representations are graded, enabling the uncertain feature responses to be used for the ensuing classification decision. Therefore, the ablation experiment gives empirical evidence instead of just theoretical to evidence fuzzification in NF-RVFL.

4.4.2 Sensitivity analysis of fuzzy membership parameters

To further investigate the influence of the fuzzy membership mechanism, a sensitivity analysis was conducted by varying its centre c and spread σ. This experiment aimed to examine the sensitivity of the NF-RVFL performance with respect to the membership-function configuration and to fix the final parameter values by a reproducible validation-based procedure.

Five candidate configurations were evaluated while keeping the feature representation, network architecture, preprocessing operations, optimizer, learning settings, and other hyperparameters unchanged. The candidate configurations were evaluated using the validation data, and the test data were excluded from parameter selection. Table 12 shows the sensitivity analysis of fuzzy membership parameters.

Table 12: Sensitivity analysis of fuzzy membership parameters

Configuration	Centre c	Spread σ	mAP (%)	AP50 (%)	AP75 (%)	AR (%)
S1	0.25	0.25	73.42	87.18	80.31	81.76
S2	0.40	0.40	74.36	88.21	81.72	82.48
S3	0.50	0.50	75.21	89.04	83.26	83.31
S4 (Selected)	0.60	0.50	75.80	89.60	84.20	83.90
S5	0.75	0.75	75.03	88.76	82.91	83.12

The sensitivity results demonstrate that the membership-function parameters have a measurable effect on the performance of NF-RVFL. The selected S4 configuration has mAP of 75.80% whereas for both other S1 and S4 SOTAs it is 73.42%. The corresponding AP50, AP75, and AR also an increase from 87.18%, 80.31%, and 81.76% to 89.60%, 84.20%, and 83.90%, respectively. However, further increasing both parameters, as in S5, reduces mAP to 75.03%. This behaviour indicates that the fuzzy representation is sensitive to the degree of feature membership spreading and that an excessively narrow or broad membership representation is not optimal.

Based on the tested configurations, the configuration S4 (c=0.60, σ=0.50) was found to have the best validation performance and was thus adopted as the final configuration. The maximum difference between the lowest-performing configuration S1 and S4 is 2.38 percentage points in mAP, giving further quantitative evidence that the fuzzy membership configuration has a meaningful impact on the performance of the model.

4.4.3 Initialization and selection of fuzzy membership parameters

The fuzzy membership parameters were determined without using test-set information. The first centre and spread values were derived from the distribution of the representations of the training feature and then a fixed number of candidate parameter configurations was tested with the validation subset.

For each candidate (c, σ) configuration, the NF-RVFL classifier was trained using the training data under the same experimental settings. The resulting model was then tested on the validation subset and the performance on the validation set determined the best set of membership functions. The best performing candidate configuration on the validation set was then chosen to be evaluated at test time.

Based on this procedure, we selected the configuration c=0.60 and σ=0.50. Once selected, both parameters were not changed, nor were they further optimized, based on the test-set observations. The test subset was only applied to the last evaluation of the trained framework. Thus, the test performance is not used to select or optimize the parameters of the fuzzy membership functions.

This process also provides the sensitivity analysis to be a true experiment in parameter selection, and not a "post hoc" selection of parameters based on test performance.

4.4.4 Overall discussion of NF-RVFL ablation and parameter sensitivity

The two complementary experiments provide quantitative evidence for the contribution of the fuzzy component in NF-RVFL. Removing the fuzzy membership mechanism consistently reduces performance for all tests in the URPC2020, DUO, RUOD and DLMU 2024 databases. In particular, mAP decreases by 1.68, 1.76, 1.58, and 0.94 percentage points for URPC2020, DUO, RUOD, and DLMU 2024, respectively. The same decreases can be seen in AP50, AP75, and AR.

The second experiment, the parameter-sensitivity experiment, also shows that the fuzzy membership configuration has an impact on the detection performance. The most optimized c=0.60, σ=0.50 configuration can get 75.80% mAP and the worst configuration can get 73.42% mAP. This shows that the fuzzy mechanism does play an active role in the classifier and its parameters need to be properly set.

Important here is that the two experiments look at different facets of the reviewer's concern. The standard RVFL ablation is used to quantitatively isolate the effect of fuzzification, while the sensitivity analysis is used to determine the sensitivity of the NF-RVFL performance to the parameters of the membership functions. The validation only parameter-selection procedure also confirms that the fuzzy parameters have not been optimized based on the information contained in the test set.

Thus, the experimental results confirm the validity of fuzzy membership modelling for NF-RVFL to analyze underwater images. The results show that the improvement in the observed performance is correlated with the use of the fuzzy component and is not simply an effect of the used RVFL architecture. The sensitivity analysis and controlled parameter-selection procedure also confirm that the chosen fuzzy configuration was not arbitrary, but was chosen systematically.

4.5 Statistical test

A statistical analysis was performed to find out if the differences in detection performance between the selected methods were statistically significant in repeated experiments. The statistical analysis was conducted with the per-run mAP values to make sure that the results of the statistical analysis could be directly related to the values reported from the experiments. A priori three detection methods were chosen for statistical comparison: YOLOv8n, HCL-YOLO, and the proposed framework. YOLOv8n and HCL-YOLO represent the baseline methods, whereas the proposed framework represents the developed method. Therefore, the three ANOVA groups are directly linked to 3 different detection methods and not to different datasets, different configurations or different experimental conditions.

Table 13: Run-level mAP results used for statistical analysis

Method	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	Mean ± SD
YOLOv8n	48.21	48.56	48.39	48.74	48.47	48.63	48.31	48.52	48.68	48.43	48.49 ± 0.17
HCL-YOLO	50.52	50.86	50.73	51.04	50.81	50.92	50.64	50.88	50.77	50.95	50.81 ± 0.15
Proposed	75.31	75.72	75.91	75.63	76.04	75.78	75.52	75.86	75.69	75.84	75.73 ± 0.21

We tested each model 10 times in training and evaluation sessions, for a total of 30 run-level observations. The repeated runs were performed with the same dataset and preprocessing procedure, the same neural network architecture, the same optimization parameters, the same training procedure and the same evaluation parameters for all runs using different random seeds. The individual run-level results and corresponding mean ± standard deviation values are reported in Table 13.

Thus, 10 mAP observations were conducted for each of the three methods, and the one-way ANOVA was conducted. The null hypothesis (H_0) states that the mean mAP values of YOLOv8n, HCL-YOLO, and the proposed framework are equal, whereas the alternative hypothesis (H_1) states that at least one method has a different mean mAP. The level of significance was chosen as $\alpha = 0.05$.

Table 14: ANOVA analysis

Source	DF	Sum of Square	Mean Square	F statistic	P-Value
Groups (between groups)	2	4560.2651	2280.1326	72105.3871	4.74×10^{-51}
Error (within groups)	27	0.8538	0.0316		
Total	29	4561.1189	-		

Table 14 shows the ANOVA analysis. The one-way ANOVA gave an f value of 72,105.3871 with degrees of freedom of (2, 27) and a p value of 4.74×10^{-51}. The p value is significantly less than $\alpha = 0.05$, so the null hypothesis (equal mean mAP values) is rejected. Thus, statistically significant differences exist between the mean mAP scores of the YOLOv8n, HCL-YOLO and the proposed framework.

The mean mAP values over the 10 runs were 48.494%, 50.812%, and 75.730% for YOLOv8n, HCL-YOLO, and the proposed framework, respectively. The corresponding standard deviations were 0.166%, 0.154%, and 0.209%. The small standard deviations suggest that there is no significant variance in the data for each method. In contrast, the differences between the means of the methods are much larger than the within-method variance.

The magnitude of the effect was also calculated using eta squared (η^2). The resulting number was around 0.9998, meaning that there was about 99.98% of the variance in the observed run-level mAP measurements due to differences between the three method groups. This very large effect size is consistent with the large difference between the method means and the relatively small within-method variance.

Since omnibus ANOVA shows statistical differences in at least one group, Tukey HSD post hoc test was conducted with a family-wise significance level of 0.05. Table 15 shows theresults of the Tukey HSD analysis.

Table 15: Post Hoc Tukey HSD analysis

Pair	Difference	SE	Q	Lower CI	Upper CI	Critical Mean	P-Value
YOLOv8n-HCL-YOLO	-2.3180	0.0562	41.221	-2.5152	-2.1208	0.1971	< 0.001
YOLOv8n-Proposed	-27.2360	0.0562	484.282	-27.4332	-27.0388	0.1971	< 0.001
HCL YOLO-proposed	-24.9180	0.0562	443.021	-25.1152	-24.7208	0.1971	< 0.001

The Tukey HSD results show that with the family-wise error rate, all three pairwise comparisons are statistically significant. The proposed framework achieved significant improvements over the mean mAP values of YOLOv8n and HCL-YOLO, which are -27.236 and -24.918 percentage points, respectively. The difference between the mean mAP scores of YOLOv8n and HCL-YOLO is also statistically significant, with the mean mAP score of HCL-YOLO being 2.318 percentage points better than that of YOLOv8n. All confidence intervals in all three cases exclude zero.

These differences are also shown in the repeated run results to be unrelated to any one experimental run. The results of the proposed framework were mAP of values 75.31% to 76.04% with an average of 75.73% and an SD of 0.21%. YOLOv8n produced values ranging from 48.21% to 48.74%, with a mean of 48.49% and an SD of 0.17%, whereas HCL-YOLO produced values ranging from 50.52% to 51.04%, with a mean of 50.81% and an SD of 0.15%.

The difference between the method means is quite large compared to the respective run-to-run variability, thus showing that the ordering of the methods is persistent over the repeated evaluations. In particular, the proposed framework achieved a higher mAP than both baseline methods in all 10 runs.

This analysis also makes it explicit what the experimental unit is for the statistical testing. Each observation represents the mAP obtained from one complete training and evaluation run of a particular detection method. Therefore, the 30 observations comprise 10 runs for YOLOv8n, 10 runs for HCL-YOLO, and 10 runs for the proposed framework. Thus, the between-group degrees of freedom will be 2 and the within-group and total degrees of freedom will be 27 and 29, respectively.

The statistical analysis can then be directly used to give a quantitative evaluation of the performance difference that was observed, rather than a single value from one benchmark as an indicator of statistical significance. The results from the run-level variability, one-way ANOVA and effect-size estimation, followed by the Tukey HSD post hoc analysis, for the experimental protocol under evaluation provides statistical support for the differences between the proposed framework and the selected baseline methods.

4.6 Discussions

Although significant progress has been achieved in underwater object detection, existing methods still face challenges in simultaneously balancing feature representation capability, computational efficiency, and detection accuracy under complex underwater environments. With rapid updates in lightweight object detection technologies like FasterNet+YOLOv8 [12] now available, underwater applications have experienced much more enhanced inference performance. Likewise, recent underwater object detection frameworks such as GCC-Net [6] and DJL-Net [23] have been developed to improve detection robustness by learning more effective and discriminative feature representations. In addition, recent multi-scale feature learning methods [13, 29] have shown that complementary contextual information from various feature levels can help to improve the performance of object localization. However, these methods may have some drawbacks, such as feature redundancy, limited use of complementary feature representations, or complexity in complex underwater settings.

To overcome these drawbacks, the framework adopts a dual-stream deep feature extraction network, based on improved YOLOv9 and Modified TResNet backbones, which simultaneously learn complementary spatial and semantic representations. The degraded underwater images are first subjected to adaptive noise reduction and contrast enhancement to enhance the visual quality of the degraded images before feature extraction. Later, heterogeneous feature representations are combined by adopting Rényi entropy-based feature fusion to highlight the informative features and suppress the redundant and noise-dominated features. The Neuro-Fuzzy Random Vector Functional Link (NF-RVFL) classifier further enhances the classification process by effectively dealing with the uncertainty and nonlinear decision boundaries that are commonly found in underwater scenes. Moreover, Grad-CAM++ gives visual explainability by showing the part of the image that most significantly affects the final prediction increasing the transparency and reliability of the proposed framework.

The benefits in terms of performance improvement can be derived from the complementary effect of the components of the proposed framework. Under poor illumination, turbidity and low underwater contrast, the high-resolution spatial features extracted by the Improved YOLOv9 backboneare combined with the rich semantic features from the Modified TResNet backbone to enhance the robustness of the model. Furthermore, to boost discriminative feature representation, a Rényi entropy-based fusion strategy is employed to effectively integrate complementary information and suppress redundant responses. Moreover, the NF-RVFL classifier is quick to make decisions and adapts itself quickly with low complexity. They, along with the interpretability support given by Grad-CAM++ and the lightweight architecture, allow the proposed framework to obtain a good trade-off between detection accuracy, computational efficiency and the interpretability of the model. Experimental results on benchmark datasets of the URPC2020, DUO, RUOD and DLMU2024 underwater scenarios show the effectiveness and generalizability of the proposed solution in different underwater scenarios.

Although the proposed framework has been shown to obtain better detection performance than the evaluated ones, it still shows some performance degradation in very difficult underwater scenarios, especially in those where the turbidity is high, the light is very low, the particles suspended in the water are dense and the light scattering effect is strong. These observations show that the proposed framework can to generalize well across a broad spectrum of underwater scenarios, but still needs some enhancements for extreme visual conditions. Adaptive underwater image enhancement, physics-guided underwater image restoration and transformer-assisted feature learning are therefore being explored the future to further enhance the robustness and generalization performance in extreme underwater environments while keeping the computational complexity at a reasonable level for real-time applications.

4.7 Proposed limitations and future scope

The proposed framework has some drawbacks even though it yields good results. In particular, it fails to sufficiently take into consideration difficult underwater circumstances such high turbidity and low light levels, which could affect the precision of detection. To address these issues, a future research will focus on further strengthening the resilience of the method in such situations through the implementation of advanced tracking systems and improved ensemble deep learning methods. Furthermore, the method for species detection and identification in free-range video data and classification of species in the ocean will be further extended, and a more precise method for species abundance estimation will be designed.

5 Conclusion

In this study, a novel underwater object detection method is proposed by combining an improved underwater image preprocessing pipeline and a dual-phase feature extractor of Improved YOLOv9 and Modified TResNet. These semantic information are captured and refined jointly by Rényi entropy-based fusion, and fast, adaptive and uncertainty-aware decision-making is provided by a Neuro-Fuzzy Random Vector Functional Link (NF-RVFL) classifier. Grad-CAM++ additionally enhances interpretability by identifying the most influential area for the detection results. The proposed framework can be considered a major improvement for underwater detection in a visually impaired environment due to its lightweight architecture, powerful capability of feature representation and fast learning speed.

Experimental results show that the framework achieves the highest mAP, precision and recall, while using a limited number of parameters (6.06M) and GFLOPs (24.32 GFLOPs) on the URPC 2020, RUOD, DUO and DLMU datasets. In addition to numeric enhancements, the system's compact design and strength provide it with a high degree of usefulness for real-time operation on resource-scarce underwater platforms like Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs), underwater drones and embedded marine monitoring equipment. This allows for applications such as ecological monitoring, underwater infrastructure inspection, search and rescue missions and exploration of marine resources.

From a more comprehensive point of view, the proposed solution is a step towards the development of reliable, interpretable and efficient underwater perception systems that can be used in a variety of and unpredictable marine environments. The work could be continued by incorporating temporal information for video-based tracking, introducing multi-sensor fusion with sonar or acoustic data, and extending the framework to long-range deep-sea exploration, and a fully autonomous pipeline for large-scale marine exploration could be achieved. Overall, this study lays a strong foundation for next-generation intelligent underwater perception technologies.

Declaration

Conflict of interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We declare that this manuscript is original, has not been published before and is not currently being considered for publication elsewhere.

Funding: No funding was received to assist with the preparation of this manuscript.

Availability of data and material: Data will be available when requested.

Authors' contributions: The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

Ethics approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] A. Ali, M. Vadivel, *Efficient energy congestion control scheme for wireless sensor networks using adaptive neuro fuzzy inference system with black widow optimization*, Iranian Journal of Fuzzy Systems, **20**(6) (2023), 187-202. https://doi.org/10.22111/ijfs.2023.43990.7748

[2] L. Cao, L. Shen, M. Yu, et al., *Prior-guided dual-reference contrastive learning for underwater object detection*, IEEE Transactions on Circuits and Systems for Video Technology, **35**(8) (2025), 8209-8224. https://doi.org/10.1109/TCSVT.2025.3545694

[3] P. Chakurkar, D. Vora, *An integrated deep learning and fuzzy logic system for road crack severity analysis, and pedestrian fall risk prediction*, Iranian Journal of Fuzzy Systems, **22**(6) (2025), 103-123. https://doi.org/10.22111/ijfs.2025.52137.9268

[4] W. Chen, H. Huang, S. Peng, et al., *YOLO-face: A real-time face detector*, The Visual Computer, **37**(4) (2021), 805-813. https://doi.org/10.1007/s00371-020-01831-7

[5] X. Chen, Q. Zeng, C. Wei, et al., *A unified underwater object detection network with image adaptive enhancement and attention-driven feature fusion*, Optics and Laser Technology, **199** (2026), 115028. https://doi.org/10.1016/j.optlastec.2026.115028

[6] L. Dai, H. Liu, P. Song, M. Liu, *A gated cross-domain collaborative network for underwater object detection*, Pattern Recognition, **149** (2024), 110222. https://doi.org/10.1016/j.patcog.2023.110222

[7] M. Eftekhari, A. Mehrpooya, F. Saberi-Movahed, V. Torra, *How fuzzy concepts contribute to machine learning*, Cham, Switzerland: Springer, 2022. https://doi.org/10.1007/978-3-030-94066-9

[8] B. Farhadinia, U. Aickelin, H. A. Khorshidi, *A parametric similarity measure for extended picture fuzzy sets and its application in pattern recognition*, Iranian Journal of Fuzzy Systems, **19**(6) (2022), 141-160. https://doi.org/10.22111/ijfs.2022.7217

[9] L. Folkman, K. A. Pitt, B. Stantic, *A data-centric framework for combating domain shift in underwater object detection with image enhancement*, Applied Intelligence, **55**(4) (2025), 1-26. https://doi.org/10.1007/s10489-024-06224-0

[10] C. Fu, X. Fan, J. Xiao, et al., *Learning heavily-degraded prior for underwater object detection*, IEEE Transactions on Circuits and Systems for Video Technology, **33**(11) (2023), 6887-6896. https://doi.org/10.1109/TCSVT.2023.3271644

[11] J. Gao, Y. Zhang, X. Geng, et al., *PE-Transformer: Path enhanced transformer for improving underwater object detection*, Expert Systems with Applications, **246** (2024), 123253. https://doi.org/10.1016/j.eswa.2024.123253

[12] K. Guo, Z. Sun, Zhang, *A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection*, Journal of Real-Time Image Processing, **21**(2) (2024), 49. https://doi.org/10.1007/s11554-024-01431-x

[13] X. Ji, S. Chen, L. Y. Hao, et al., *FBDPN: CNN-Transformer hybrid feature boosting and differential pyramid network for underwater object detection*, Expert Systems with Applications, **256** (2024), 124978. https://doi.org/10.1016/j.eswa.2024.124978

[14] D. Kong, Y. Zhang, X. Zhao, et al., *CFSPNet: Cross-domain feature synergy and perception for robust underwater object detection*, Expert Systems with Applications, **320** (2026), 132174. https://doi.org/10.1016/j.eswa.2026.132174

[15] C. Li, W. Liu, G. Gong, et al., *SU-YOLO: Spiking neural network for efficient underwater object detection*, Neurocomputing, **644** (2025), 130310. https://doi.org/10.1016/j.neucom.2025.130310

[16] Z. Liu, B. Wang, Y. Li, et al., *UnitModule: A lightweight joint image enhancement module for underwater object detection*, Pattern Recognition, **151** (2024), 110435. https://doi.org/10.1016/j.patcog.2024.110435

[17] P. Patro, K. Kumar, G. S. Kumar, A. K. Sahu, *Intelligent data classification using optimized fuzzy neural network and improved cuckoo search optimization*, Iranian Journal of Fuzzy Systems, **20**(6) (2023), 155-169. https://doi.org/10.22111/ijfs.2023.44767.7887

[18] N. Rishi, A. Kumar, R. Golash, *Fisheries management with deep learning-based fish species detection: A sustainable approach*, In International Conference on Cognitive Computing and Cyber Physical Systems (pp. 359-369). Springer, Singapore, **1128** (2025). https://doi.org/10.1007/978-981-97-7371-8_28

[19] X. Shen, H. Wang, Y. Li, et al., *Criss-cross global interaction-based selective attention in YOLO for underwater object detection*, Multimedia Tools and Applications, **83**(7) (2024), 20003-20032. https://doi.org/10.1007/s11042-023-16311-y

[20] P. Song, P. Li, L. Dai, et al., *Boosting R-CNN: Reweighting R-CNN samples by RPN's error for underwater object detection*, Neurocomputing, **530** (2023), 150-164. https://doi.org/10.1016/j.neucom.2023.01.088

[21] T. Tian, J. Cheng, D. Wu, Z. Li, *Lightweight underwater object detection based on image enhancement and multi-attention*, Multimedia Tools and Applications, **83**(23) (2024), 63075-63093. https://doi.org/10.1007/s11042-023-18008-8

[22] A. Upadhayay, S. Srivastav, S. K. Biswas, et al., *Evolutionary neuro-fuzzy random vector functional link with genetic optimization*, In 2025 IEEE International Conference on Fuzzy Systems (FUZZ), (2025). https://doi.org/10.1109/FUZZ62266.2025.11152032

[23] B. Wang, Z. Wang, W. Guo, Y. Wang, *A dual-branch joint learning network for underwater object detection*, Knowledge-Based Systems, **293** (2024), 111672. https://doi.org/10.1016/j.knosys.2024.111672

[24] J. Wu, J. Chen, Q. Lu, et al., *U-ATSS: A lightweight and accurate one-stage underwater object detection network*, Signal Processing: Image Communication, **126** (2024), 117137. https://doi.org/10.1016/j.image.2024.117137

[25] Y. Wu, W. Wang, C. Lin, et al., *Towards multimodal underwater object detection: A bidirectional feature recomposition network and visual-sonar dataset*, Expert Systems with Applications, **316** (2026), 131710. https://doi.org/10.1016/j.eswa.2026.131710

[26] F. Yu, F. Xiao, C. Li, et al., *AO-UOD: A novel paradigm for underwater object detection using Acousto-Optic fusion*, IEEE Journal of Oceanic Engineering, **50**(2) (2025). https://doi.org/10.1109/JOE.2025.3529121

[27] G. Yuan, J. Song, J. Li, *IF-USOD: Multimodal information fusion interactive feature enhancement architecture for underwater salient object detection*, Information Fusion, **117** (2025), 102806. https://doi.org/10.1016/j.inffus.2024.102806

[28] D. Zhang, C. Yu, Z. Li, et al., *A lightweight network enhanced by attention-guided cross-scale interaction for underwater object detection*, Applied Soft Computing, **184**(Part B) (2025), 113811. https://doi.org/10.1016/j.asoc.2025.113811

[29] J. Zhou, Z. He, D. Zhang, et al., *Spatial residual for underwater object detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **47**(6) (2025), 4996-5013. https://doi.org/10.1109/TPAMI.2025.3548652