Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080975 Published Online: 27 May 2026

ARTICLE

Underwater Objects Detection Based on a Multi-Stage Deep Learning Framework

Rana Lateef[1] **and Asmaa Abdul Jabbar**[2,*]

[1]Department of Cybersecurity Science, College of Science, Al-Iraqia Science University, Baghdad, Iraq
[2]Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, Iraq

*Corresponding Author: Asmaa Abdul Jabbar. Email: asmaasadiq@uomustansiriyah.edu.iq

Received: 20 February 2026; Accepted: 29 April 2026

ABSTRACT: The challenges of underwater object detection are derived from complex environmental conditions, including light scattering, absorption, and turbidity. The deep learning approaches have enhanced the detection of objects in these low-visual conditions. This work presents a multi-stage object-detection framework for the underwater environment that performs well on the Semantic Segmentation of Underwater Imagery (SUIM) benchmark. To begin with, there is the adaptive Multi-Scale Retinex with Color Restoration (MSRCR) algorithm, which improves image quality by correcting color distortions and increasing contrast. Second, an augmented YOLOv8 model (with a ResNet-50 backbone and the Convolutional Block Attention Module (CBAM)) is used to extract powerful features for object detection in low-light conditions. Lastly, a LightGBM classifier selects initial detections using contextual information to reduce false positives. The proposed model is evaluated on the SUIM dataset, with ground-truth segmentation masks converted to bounding boxes according to standard COCO protocols for detection-based training and evaluation. Comparative experiments against reimplemented YOLO-based underwater detectors demonstrate that the proposed model achieves a macro-average mAP@0.5 of 86.33%, outperforming YOLOv8-nano (80.13%), YOLOv7+enhancement (84.23%), and AIT-YOLOv7 (83.4%) on the SUIM benchmark under similar conditions.

KEYWORDS: YOLOv8; ResNet-50; multi-scale retinex with color restoration (MSRCR); LightGBM; underwater images

1 Introduction

Water covers over 70% of Earth's surface. As such, aquatic environments are an important aspect in sustaining mankind, ensuring the stability of the world's ecological system, and serving as the engine of economic growth through fisheries, shipping, and resource harvesting. The ocean environment is inherently intricate and contains organisms of diverse sizes and morphologies, which are often widely dispersed in space. This complexity, in itself, creates major problems for an automated underwater imaging system, which is essential in marine research, ecosystem management, biodiversity conservation, and the protection of endangered species. Besides, numerous interesting objects, including rare marine species and archaeological artifacts, are small and often well-disguised against a sophisticated background, which severely challenges conventional strategies [1,2].

The physical characteristics of the water medium greatly complicate underwater conception. Phenomena such as non-uniform color distortion, light absorption, and scattering are considered to degrade underwater imaging [3]. The light is attenuated as it travels through water, and various wavelengths are

absorbed at varying rates, giving a characteristic bluish-green color and loss of important color details. Meanwhile, the suspended element, plankton and sediment in the sea, causes the scattering, which degrades the contrast, gives the image a hazy effect, and adds more noise to it. Degradation variables, such as lighting, turbidity, and low contrast, significantly obscure object features and complicate identification and classification tasks [4].

Even though the traditional image processing and computer vision techniques have been used to address these concerns, the complex and dynamic nature of underwater degradation often limits their effectiveness [1]. Over the past few years, artificial intelligence, specifically deep learning and convolutional neural networks (CNNs), has advanced at an impressive pace, offering a new set of powerful object detection tools. These new methods have shown great detection and processing speeds in many areas, slowly replacing older technologies [4].

Nevertheless, the straightforward implementation of the single, standard deep learning models to underwater images may not produce optimal results. The degradation that is unique and multi-faceted of the underwater environment may mask the characteristics on which these networks are based, resulting in false-positives and high rates of false-positives, particularly with small or hidden objects. This has led to the development of multi-stage deep learning frameworks that decouple the complex underwater detection problem into manageable sub-tasks [2,3,5].

Therefore, multi-stage deep learning frameworks have been developed that decouple the detection task in a complex underwater environment into manageable subtasks. At the same time, despite the rapid development in this field, problems remain in the accurate detection of small, obscured, and camouflaged objects [3,5].

The main contributions of this work are (1) a new multi-stage framework of underwater object detection that synergistically integrates an adaptive MSRCR image enhancement, an augmented YOLOv8 network with parallel ResNet-50 and CBAM attention to robustly extract features with a contextual refinement at a stage of LightGBM that minimizes false positives; (2) a systematic adaptation of the SUIM segmentation dataset using standard COCO for bounding-box-based detection; and (3) dual-branch architecture with attention-guided feature fusion is effective in terms of the poor visibility, color distortion, and complex backgrounds in underwater imagery.

The individual components of the proposed framework—MSRCR enhancement, ResNet-50 feature extraction, CBAM attention, and LightGBM classification—are established techniques in computer vision. However, existing works typically apply these methods in isolation or in simple two-stage combinations. Our contribution lies in their integration into a three-stage pipeline designed specifically for underwater detection, where: (i) the MSRCR enhancement is adaptively tuned for underwater color correction, (ii) the dual-branch feature extraction fuses multi-scale representations from both YOLOv8 and ResNet-50 with attention-guided fusion, and (iii) the LightGBM refiner uses a 51-dimensional feature vector that combines detection, geometric, statistical, attention, and contextual cues. This specific composition, along with the systematic adaptation of the SUIM dataset for detection evaluation, has been reported in this work.

The other parts of this work are arranged in the following way. Section 2 explores related literature in underwater image enhancement and object detection. Section 3 expounds on the proposed multi-stage architecture, the MSRCR enhancement block, the augmented YOLOv8 detector with ResNet-50 and CBAM, and the refinement stage, which is implemented with LightGBM. Section 4 demonstrates the SUIM data and the measurement indicators. Section 5 shows and reports the experimental findings, ablation experiments, and comparisons with the state-of-the-art methods. Section 6 wraps up the paper and gives the directions that are to be taken.

2 Related Works

Recent progress in underwater object detection and segmentation has explored different deep learning architectures and mechanisms to overcome limitations in low image quality, cross-domain and occlusion, and open-set conditions. This section provides a review of current studies on underwater detection and segmentation across different standard datasets.

A two-stage system for underwater image classification in open-set circumstances has been suggested by Akhtarshenas and Toosi [6]. In this work, the reconstruction error is implemented to identify an unknown species with an autoencoder, followed by EfficientNet-B0 for known species. The suggested work was applied to the WildFish dataset. While it is effective in minimizing false alarms, the technique is sensitive to threshold change and may not work well in states where reconstruction error is ambiguous, which limits its practicality in various underwater conditions.

A Fuzzy Chaos Multi-population Flow Direction Algorithm (FCMFDA) has been used to combine a fully connected layer of the DenseNet201 backbone with an optimized Extreme Learning Machine (ELM) classifier proposed by Yang et al. [7]. This method improves the classification accuracy and convergence rate of traditional CNNs and other metaheuristic-optimized ELMs on both the Fish4Knowledge and URPC 2018 datasets. Nevertheless, the approach is specifically designed for classification and is not naturally extended to object detection.

Wang et al. [8] introduced an enhancement to YOLOv7, with a parallel image enhancement branch and Contextual Transformer (CoT) blocks, to minimize feature degradation in underwater images. The use of focal Efficient Intersection over Union (EIoU) loss improves bounding box regression under occlusion. Nevertheless, the model gains robust results on the URPC2020 and UTDAC2020 datasets; the fixed enhancement module lacks domain adaptability, potentially limiting performance across different underwater conditions.

Dakhil and Khayeat [9] employed an encoder-decoder architecture with VGG16 and Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN)-based preprocessing for semantic segmentation on the SUIM dataset. Although the model's performance is balanced and efficient, the enhancement adopted on the Generative Adversarial Network (GAN) network increases preprocessing overhead, which may introduce artifacts that degrade segmentation accuracy.

Zhou et al. [2] proposed an underwater optical detection network (UODN) model that employs the YOLOv8 framework and incorporates Cross Stage Multi-Branch (CSMB) and Large Kernel Spatial Pyramid (LKSP) modules to enhance the low quality of the input feature extraction. The model has suitable real-time performance, but its architecture makes deployment on constrained platforms difficult.

Pachaiyappan et al. [10] introduced the AIT-YOLOv7, which combines CBAM, Modified Swin Transformer Blocks, and a diffusion-based U-Net for image enhancement and detection. This diffusion model causes remarkable computational overhead, making it less suitable for real-time applications, even though it performs better on the TrashCan dataset compared to the previous YOLO model.

UWSegFormer is a segmentation framework that involves a transformer-based segmentation and quality attention, multi-scale aggregation, and an edge learning module, has been introduced by Zuo et al. [11]. While the proposed model achieves advanced results at low computational cost on both the SUIM and DUT datasets, its hierarchical transformer architecture limits its flexibility, which creates a trade-off between precision and parameter efficiency.

A DeepLabv3+ semantic segmentation model with dynamic multi-scale fusion, CBAM, and deep supervision model has been suggested by Wang et al. [12]. The model excels at handling scale variation and complex backgrounds in the SUIM dataset, but at the same time suffers from higher complexity and has not been evaluated on domain-shifted underwater data.

Chen et al. [13] introduced an enhanced SegFormer model by exchanging the Mix Transformer backbone with a Swin Transformer, then integrating the Efficient Multi-scale Attention (EMA) mechanism into the downsampling stages and the decoder, and finally adding a Feature Pyramid Network (FPN) to the decoder for better multi-scale fusion. The model balances accuracy and efficiency on the SUIM dataset, but it struggles to identify small, distant objects under poor lighting conditions. Also, the complexity of the proposed model may limit its deployment on low-resource systems.

Despite the reviewed approaches demonstrating remarkable progress, few approaches simultaneously address the challenges of image enhancement, feature refinement, and detection under adverse underwater conditions within a unified framework. This will be discussed in the following sections of this manuscript.

3 The Proposed Multi-Stage Model

Fig. 1 shows a three-stage pipeline proposed for detecting the underwater objects. The framework starts with an image-quality improvement stage using Multi-Scale Retinex with Color Restoration (MSRCR) to address underwater color distortion and low contrast. Then, in Stage 2, an augmented detection network is applied to the improved image, using a YOLOv8-nano backbone as a feature extractor and a pre-trained ResNet-50, which is further enhanced by a Convolutional Block Attention Module (CBAM). In Stage 3, the extracted features are combined, condensed into a 51-dimensional vector, and trained with a LightGBM binary classifier. Post-processing filtering and Non-Maximum Suppression are then applied to introduce the final detection result.

Stage 1: Image Enhancement via Adaptive MSRCR: The Retinex theory, proposed by Land and McCann, describes the human visual system's ability to perceive color consistency across different lighting conditions. The visual system does not perceive absolute light intensity, but rather relative lightness within local areas of a scene. Image enhancement algorithms based on Retinex are built on this concept [14]. First, the input image $I(x, y)$ is decomposed by a Retinex algorithm into two components in the logarithmic domain:

$$Log\ R\,(x, y) = Log\ I\,(x, y) - Log\ L\,(x, y) \tag{1}$$

Reflectance $R(x, y)$: This component is the fundamental information about the colors of the objects in the scene, the intrinsic qualities of the objects.

Illumination $L(x, y)$: This component captures the non-uniform state of light, such as ambient light in the water. The illumination component $L(x, y)$ is typically estimated by convolving the input image with a Gaussian filter:

$$L\,(x, y) = I\,(x, y) * G\,(x, y) \tag{2}$$

where $I(x, y)$ denotes the underwater image, $G(x, y)$ is the Gaussian function, and * denotes the convolution operation [15]. The Gaussian kernel is defined as:

$$G\,(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2-y^2}{2\sigma^2}} \tag{3}$$

With the scale parameter σ controlling the kernel width. The Single-Scale Retinex (SSR) uses a single scale (standard deviation) for the Gaussian function. However, a single scale is often insufficient, where a small scale maintains sharp contours but may introduce halo effects, whereas a large scale eliminates halo effects at the cost of local contrast.

The Multi-Scale Retinex (MSR) algorithm overcomes this limitation by combining the results of multiple SSR operations that have been applied at different scales with weighting averaging:

$$Log\ R\,(x,y) = \sum_{n=1}^{N} W_n[\log\ I\,(x,y) - Log\ L_n(x,y)] \tag{4}$$

Now, N indicates the number of scales, L_n is the illumination component estimated using scale sigma, and Wn denotes the weighting factor for the n-th scale. The combination of information from multiple scales in MSR offers stronger enhancement, compressing the dynamic range and preserving better color saturation and edges, which is key to restoring poor color in underwater imagery [15].

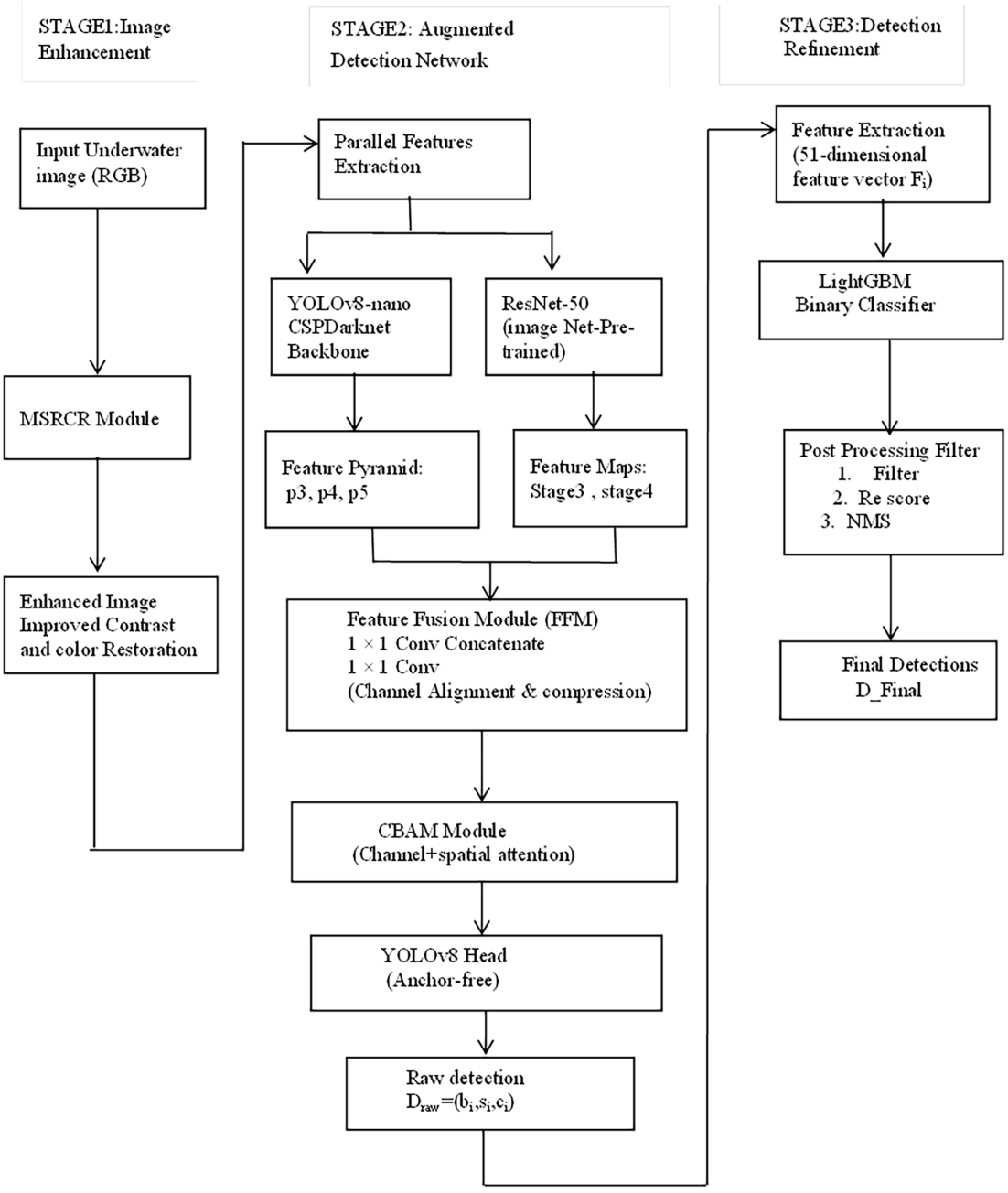

Figure 1: The architecture of the proposed multi-stage underwater object detection framework. Stage 1: An adaptive MSRCR image enhancement. Stage 2: Parallel feature extraction using YOLOv8-nano and ResNet-50 with CBAM attention, followed by feature fusion. Stage 3: LightGBM-based detection refinement.

The Retinex algorithm is founded on the grey world hypothesis, which states that the average reflectance across color channels is equal. When this assumption is violated, Retinex processing may create a "grey-out" effect then resulting in color desaturation. Therefore, a color restoration process is required to confirm accurate color representation of images with grey-world violations. Multi-Scale Retinex with Color Restoration (MSRCR) addresses this limitation by incorporating a color-restoration function that enhances images through dynamic-range compression while preserving color accuracy. The color restoration component corrects color distortion and at the same time maintains natural colors in the enhanced images [16]. The MSRCR output for the i-th color channel is stated as:

$$R_{MSRCRi}(x, y) = G.\left[C_i(x, y) * R_{MSRi}(x, y) - b\right] \tag{5}$$

where i indexes the three color channels, $RMSRi(x, y)$ denotes the output of the Multi-Scale Retinex for the i-th channel, $Ci(x, y)$ is the color restoration function for the i-th channel, the final gain factor is represented by G, and b represents the offset constant.

To maximize dynamic range utilization, an adaptive mechanism that modifies the gain parameter G based on global image statistics is employed by the following equation:

$$G = \frac{255}{max_{i,x,y}(R_{MSRCRi}(x, y))} \tag{6}$$

where the numerator value (255) represents the maximum pixel value across all channels and spatial locations after color restoration. Using this channel-wise adaptation prevents the "grey-out" effect that is widely observed in standard Retinex implementations when applied to underwater images with non-uniform color casts. The final enhanced image is attained by performing Eq. (5) on each color channel separately and then combining the results.

Stage 2: Augmented Object Detection Network. To address the challenges of small, blurred underwater objects and gradient vanishing in deep networks, the basic detector adopts a modified architecture of the YOLOv8, which is enhanced with a ResNet-50 branch and Convolutional Block Attention Modules (CBAM) to address the challenges of small, blurred underwater objects and gradient vanishing in deep networks [17]. This stage performs the following steps:

1. Input and Preprocessing: The images enhanced by MSRCR (Stage 1) are resized to 640 × 640 pixels and fed into two feature extractors simultaneously. They are a YOLOv8-nano backbone and a pre-trained ResNet-50.

YOLOv8-nano Backbone (CSPDarknet): the YOLOv8-nano variant has been employed as a base detector for its favorable speed-accuracy trade-off. The primary pyramid features extracted from CSPDarknet backbone at three scales (P3, P4, P5), which are essential for multi-scale object detection [18].

P3: stride 8, spatial size 80 × 80, channels = 64

P4: stride 16, spatial size 40 × 40, channels = 128

P5: stride 32, spatial size 20 × 20, channels = 256

The feature maps capture multi-scale information that is essential for detecting objects of varying sizes.

The Auxiliary ResNet-50 Feature Extraction: The ResNet-50 network (initialized with ImageNet weights), processes the same 640 × 640 pixel input in parallel and as a complementary feature extractor. The feature maps are extracted from the final convolutional layers of the two-stage blocks (stage 3 and stage 4). These maps are adopted because they contain high-level semantic information and also retain sufficient spatial resolution to assist in localization.

Stage 3 (after conv3_x): stride 8, spatial size 80 × 80, channels = 512.

Stage 4 (after conv4_x): stride 16, spatial size 40 × 40, channels = 10,243.

2. Feature Fusion Modules: The fusion of YOLOv8 pyramid and ResNet-50 features is accomplished at two levels (P4 and P5) through a lightweight Feature Fusion Module (FFM). (The original YOLOv8 P3 features are kept unchanged, as low-level details are already well captured by the backbone. Each FFM consists of the following steps (all convolutional layers are followed by BatchNorm and SiLU activation unless noted otherwise):

– Fusion at P4 level (stride 16):

ResNet stage 3 features (80 × 80 × 512) are first downsampled to match the spatial size of P4 (40 × 40). This is done by a 1 × 1 convolution reducing channels to 128, followed by a 3 × 3 convolution with stride 2 (padding 1) that outputs 40 × 40 × 128. The resulting feature map is concatenated channel-wise with the YOLOv8 P4 map (40 × 40 × 128), producing a 40 × 40 × 256 tensor.

A 1 × 1 convolution compresses the channels back to 128, yielding the fused P4 representation.

Finally, a CBAM (Convolutional Block Attention Module) is applied to this fused map to emphasize informative channels and spatial regions sequentially. The output is an enhanced P4 feature map (40 × 40 × 128) that now incorporates both YOLOv8 details and ResNet-50 semantics.

– Fusion at P5 level (stride 32):

ResNet stage 4 features (40 × 40 × 1024) are processed to match the P5 resolution (20 × 20). First, a 1 × 1 convolution reduces the number of channels to 256. Then, a 3 × 3 convolution with a stride of 2 (padding 1) downsamples the spatial size to 20 × 20 while preserving 256 channels. This map is concatenated with the YOLOv8 P5 map (20 × 20 × 256) to form a 20 × 20 × 512 tensor.

A 1 × 1 convolution reduces the channel count to 256, producing the fused P5 representation.

A CBAM is again applied, yielding an enhanced P5 feature map (20 × 20 × 256) that fuses high-level semantic cues from ResNet with the original YOLOv8 features.

3. The Neck and Head integration with YOLOv8. The improved feature pyramid has changed to:

P3 (80 × 80 × 64)—based on the YOLOv8 backbone (no fusion).

P4_fused (40 × 40 × 128)—after FFM and CBAM

P5_fused (20 × 20 × 256)—after FFM and CBAM

These three feature maps are fed into the regular YOLOv8 neck (PANet) for multi-scale feature aggregation, and then into the YOLOv8 detection head. The head generates raw detections: $D_{raw} = (b_i, s_i, c_i)$, where b_i is the bounding-box coordinates, s_i is the objectness score, and c_i is the initial class probability. Table 1 summarizes the Tensor Dimensions and the operations.

Stage 3: LightGBM-Based Detection Refinement. Because of the background's complexity and the low contrast of the objects, the raw detections (Draw) produced by stage 2 often contain false positives. To improve these results, a LightGBM classifier has been employed to analyze a wide set of contextual and geometric features acquired from each network.

Table 1: Tensor dimensions and fusion operations in Stage 2. For each feature level (P3, P4, P5), the table shows the source network, input tensor size, fusion operation, and output tensor size after feature fusion and CBAM attention.

Level	Source	Input Size	Operation for Fusion	Output Size
P3	YOLOv8	80 × 80 × 64		80 × 80 × 64
P4	YOLOv8(P4) + ResNet Stage3	40 × 40 × 128 & 80 × 80 × 512	– 1 × 1 conv. (512→128), – 3 × 3 conv. sride-2 (40 × 40 × 128), – concat. (40 × 40 × 256), – 1 × 1 conv (256→128), – CBAM	40 × 40 × 128
P5	YOLOv8(P5) + ResNet Stage4	20 × 20 × 256 & 40 × 40 × 1024	– 1 × 1 conv. (1024→256), – 3 × 3 conv. sride-2 (20 × 20 × 256), – concat. (20 × 20 × 512), – 1 × 1 conv (512→256), – CBAM	20 × 20 × 256

Feature Engineering for LightGBM: For each raw detection proposal i, a 51-dimensional feature vector F_i is constructed to represent the candidate detection from Stage 2 comprehensively. The 51 features were selected based on three principles: (1) complementary information from different stages of the pipeline, (2) established effectiveness in prior object detection refinement literature, and (3) computational efficiency during inference. It encompasses:

- Three (3) detection-based features: These features reflect the detector's uncertainty and raw confidence. Also, Class entropy helps identify ambiguous predictions where the detector is unsure between multiple classes.
- Four (4) geometric features: Bounding box aspect ratio and relative area encode scale priors; for example, for 'diver', the false positives often have unrealistic aspect ratios. So, the center coordinates help suppress edge artifacts.
- Six (6) statistical features from the MSRCR-enhanced image: mean and standard deviation for each RGB channel within the bounding box region. These capture local image quality; low-contrast regions or color anomalies correlate with false positives.
- Twelve (12) statistical features from YOLOv8 feature maps: mean, std, max, min extracted from the three YOLOv8 scales (P3, P4, P5) at the bounding box locations. These multi-scale features (P3, P4, P5) provide objectness information at different resolutions. Low activation at P5 (high-level semantics) with high activation at P3 (low-level details) often shows texture-based false positives.
- Eight (8) statistical features from ResNet-50: Also, mean, std, max, min from stage3 and stage4 provide semantic context that YOLOv8 may miss, especially for small or occluded objects.
- Fifteen (15) attention-based features from CBAM: channel attention mean and spatial attention mean, std, max, min for each of the three YOLOv8 scales (P3_fused, P4_fused, P5_fused) and the two ResNet stages (stage 3 and stage 4)—total 5 sources and 3 statistics = 15 features. These attention weights indicate which spatial regions and channels the network focuses on. Unusual attention patterns (e.g., uniformly distributed attention) often correlate with false positives in low-visibility conditions.
- and Three (3) contextual features: mean neighbor objectness, neighbor count, and distance to nearest detection are relationships between neighboring detections. False positives rarely appear in isolation; they often cluster or appear far from other objects.

Although some features are correlated (e.g., mean and standard deviation from the same source), LightGBM's tree-based architecture is inherently robust to correlated features and automatically selects the most informative splits. The full feature set was retained to avoid premature information loss, and reliance on LightGBM's built-in feature importance and regularization (feature_fraction, lambda_l1, lambda_l2) was used to mitigate overfitting.

These rich features enable LightGBM to effectively distinguish true detections from false positives by using geometry, appearance, context, and attention-derived cues, meeting the requirements specified for underwater images, such as poor visibility, color distortion, and distracting backgrounds.

Training the Refiner: The LightGBM model is trained as a binary classifier (True Detection vs. False Positive). Training data is generated by applying Stage 2 to the training set of the SUIM dataset, and to avoid data leakage and ensure unbiased evaluation, the original training set was divided into two disjoint groups:

Detector training set (80%) used exclusively to train the YOLOv8-ResNet-CBAM detector.

LightGBM training set (20%): This set was held out from detector training and used to generate proposals for training the LightGBM classifier. This guarantees that the refiner never sees images used to train the base detector, thereby eliminating leakage.

The validation set (also disjoint) is used for early stopping and hyperparameter tuning. All evaluation metrics reported in the paper are computed on a separate test set.

Positive and Negative Sample Definition

For each image in the LightGBM training set, we run the frozen detector and collect all raw proposals. Each proposal is matched to ground-truth boxes using bounding box:

Positive samples (label = 1) with overlap ≥ 0.5, and

Negative samples (label = 0) with overlap < 0.3.

Class Imbalance and Loss Function:

Because negative proposals widely outnumber positive ones, the Custom focal loss objective, which replaces LightGBM's default binary cross-entropy loss, has been employed. The LightGBM does not natively support focal loss, so a custom objective function using LightGBM's fobj API was used. The focal loss for binary classification is defined as:

$$FL(p, y) = -\alpha (1 - p_t)^{\gamma} \log(p_t), p_t = \begin{cases} p & if\ y = 1 \\ 1 - p & if\ y = 0 \end{cases} \tag{7}$$

where p is the predicted probability (after sigmoid transformation),

$y \in \{0, 1\}$ is the ground-truth label,

$\alpha = 0.25$ balances positive/negative importance, and

$\gamma = 2.0$ focuses training on hard misclassified examples.

The Hyperparameters Selection for LightGBM

After grid search on the validation set, the following LightGBM hyperparameters are selected:

num_leaves: 63

learning_rate: 0.05

feature_fraction: 0.8

bagging_fraction: 0.8

bagging_freq: 5

min_child_samples: 20

lambda_l1: 0.1

lambda_l2: 0.2

num_boost_round: 500 (with early stopping if validation loss does not improve for 50 rounds)

early_stop_rounds: 50

metric: binary_logloss

Custom objective: $\alpha = 0.25$, $\gamma = 2.0$

is_unbalance: not used.

All LightGBM hyperparameters (num_leaves, learning_rate, feature_fraction, etc.) were determined by grid search in the validation set, which was different from the detector training set and LightGBM training set. The search ranges were: num_leaves [31, 63, 127], learning_rate [0.01, 0.05, 0.1], feature_fraction [0.7, 0.8, 0.9], and lambda_l1/l2 [0.0, 0.1, 0.2]. The best parameters (num_leaves = 63, learning_rate = 0.05, . . .) were those that achieved the lowest binary log-loss on the validation set after 500 rounds of boosting with early stopping (50 rounds). And the focal loss parameters were fixed at $\alpha = 0.25$ and $\gamma = 2.0$ following common practice.

Inference and Post-Processing: During inference, each raw detection from Stage 2 is converted into its feature vector F_i and passed to the trained LightGBM classifier. The produce a refinement score $r_i \in [0, 1]$ and the final set of detections D_{final} is achieved by the following:

Filtering: Removing all proposals where $r_i < 0.5$.

Re-Scoring: For retained proposals, the final confidence score is computed by $S_i^{final} = \sqrt{S_i . r_i}$ harmonizing the detector's confidence and the refiner's assessment.

Non-Maximum Suppression (NMS): Standard NMS with an overlap threshold of 0.45 is applied on D_{final} to remove duplicates.

This stage successfully reduces the false alarms caused by background textures, such as rocks mistaken for reefs.

4 Dataset and Evaluation Metrics

The Semantic Segmentation of Underwater Imagery (SUIM) dataset is used in this work, which was proposed by the Stanford University team. The dataset comprise several categories, which are considered key elements of organisms and underwater objects. These are the water body background (BW), human divers (HD), aquatic plants/flora (PF), wrecks/ruins (WR), robots and instruments (RO), reefs and other invertebrates (RI), fish and other vertebrates (FV), and sea-floor and rocks (SR) [9]. The SUIM dataset incorporates 1525 RGB images for training and validation, and 110 images for testing, to make it easier to evaluate benchmarks. These images were selected from a large collection of images collected during oceanic journeys and human-robot synergies conducted at varying water depths. These are C-spatial-resolution images of different dimensions of 1906 × 1080, 1280 × 720, 640 × 480, 256 × 256 [19]. Fig. 2, exposes the number of objects in each category, the correlation between them, and the distribution of RGB channel values in the SUIM dataset.

The SUIM dataset is essentially imbalanced, reflecting the natural frequency of objects in underwater environments. Table 2 reveals the image count for each category, the percentage of each one, and the average area that each category occupies within those images. Also, Table 2 shows that BW, Fv and RI are the majority

classes, while RO, WR, and PF are minority classes and both SR and HD fall into moderate and frequent categories, respectively.

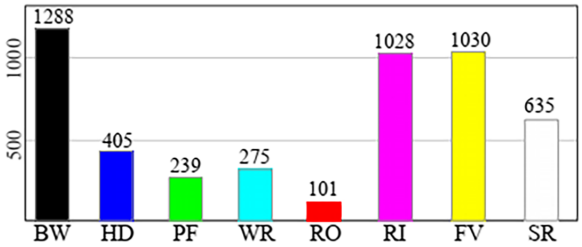

Figure 2: The population of SUIM object categories [9].

Table 2: The distribution of SUIM dataset classes.

Class	Count of Images	Images (% of Total)	Average Area per Image
Fish_vertebrates (FV)	1030	66.5%	11.8%
Wreck_ruins (WR)	275	17.7%	41.9%
Human_diver (HD)	405	26.1%	7.5%
Aquatic plants/flora (PF)	239	15.4%	15.1%
Robots and instruments (RO)	100	6.5%	4.7%
Reefs and other invertebrates (RI)	1028	66.3%	52.3%
Sea-floor and rocks (SR)	635	41.0%	34.8%
Water body background (BW)	1288	83.1%	37.5%

The standard COCO detection protocol was adopted to assess the proposed framework. The primary metric used is the mean Average Precision at threshold 0.5 (mAP@0.5). Secondary and auxiliary metrics, class precision and recall provide insight into false-positive and false-negative rates and enable comparison with detection-based methods. All metrics are based on False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN). Where TP represents the overlapping area, FP represents the predicted area outside the ground truth, and FN represents the ground truth area missed by the prediction. The precision: this parameter is applied to indicate the capacity of the proposed model to make correct prediction by the next equation [20]:

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall: this measure determines the ratio of the correctly recognized objects amongst all the real objects existing in the image through the application of the equation [21]:

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

F1 Score: The harmonic mean of Precision and Recall is the F1 Score; it gauges how accurate optimistic predictions are. It can be mathematically represented and simplified as [22,23]:

$$F1\,Score = \frac{2TP}{2TP + Fp + FN} \tag{10}$$

In this work, the SUIM dataset is used to assess the multi-stage underwater object detection framework. Because the SUIM is a semantic segmentation benchmark that provides pixel-level annotations, we adjust it for object detection by converting the ground-truth segmentation masks into bounding boxes, where, for each connected component of pixels belonging to an object instance, an axis-aligned minimum bounding rectangle is fitted based on standard practice for detection tasks [24,25].

The standard COCO evaluation protocol is used to calculate all detection metrics at the bounding-box level, where a detection is considered a true positive if the predicted bounding box has sufficient overlap with the ground truth box (threshold 0.5). All reported metrics follow this protocol. While the SUIM dataset contains eight semantic categories, not all are appropriate for object detection evaluation. In standard object detection benchmarks (e.g., COCO, Pascal VOC), detection targets are discrete, localizable entities with spatial extent. Classes such as "Water body background" (BW) and "Sea-floor and rocks" (SR) represent amorphous background regions or substrate types rather than coherent objects. Therefore, including these as detection targets is conceptually problematic, where a "background" class cannot be meaningfully localized with a bounding box. Therefore, all reported results in this work are based exclusively on the six semantically meaningful object categories: Fish Vertebrates (FV), Wreck Ruins (WR), Human Diver (HD), Aquatic Plants/Flora (PF), Robots and Instruments (RO), and Reefs and Invertebrates (RI). Classes BW (Water body background) and SR (Sea-floor and rocks) were removed from the training and evaluation sets. Also, all experiments, including detector training, LightGBM refinement, and final evaluation, were conducted on the remaining six classes only.

5 Experimental Results

Here, the findings from training the system with the SUIM Dataset will be reported, along with the test findings from the testing phase of the Dataset. Fig. 3, presents underwater images of various objects from the Dataset's categories, each identified by a frame. While Fig. 3, demonstrates successful detection examples, the box-based visualizations alone are insufficient to validate the robustness of a three-stage system fully. To provide further diagnostic evidence, the per-class precision, recall, F1 Score and standard deviations across multiple independent runs are calculated and presented in Tables 3 and 4, showing stable performance. The macro-averaged (mAP@0.5) achieved by the model with mean ± standard deviation across five independent training runs with different random seeds is demonstrated in Table 3. The highest per-class (mAP@0.5) is achieved for Reefs and other invertebrates (RI) at 89%, followed by Aquatic plants/flora (PF) at 88%, while the lowest is for Wreck_ruins (WR), at 84%. Moreover, the high (mAP@0.5) values across all classes (84%–89%) indicate that the refinement stage effectively localizes objects with high precision.

Table 4 presents the evaluation metrics, comprising mean Recall (mRecall), mean Precision (mPrecision), and mean F1 (mF1Score) across object categories. All results are reported as the mean ± standard deviation across five independent training runs with different random seeds to evaluate the statistical robustness. The results adopt a powerful detection performance across most categories. Both the Fish Vertebrates (FV) and Human Diver (HD) achieve high precision scores of 93.14 ± 0.30% and 92.00 ± 0.28%, respectively, signifying low false-positive rates for these critical classes. Also, the balance between precision and Recall across most classes yields F1-scores in the mid-to-high 80s, indicating minimal false alarms and missed detections. Moreover, the consistently low standard deviations over all metrics confirm the stability and robustness of the proposed multi-stage framework. It is worth noting that the LightGBM refinement stage is the main cause of high precision, which effectively filters out false positives in complex backgrounds.

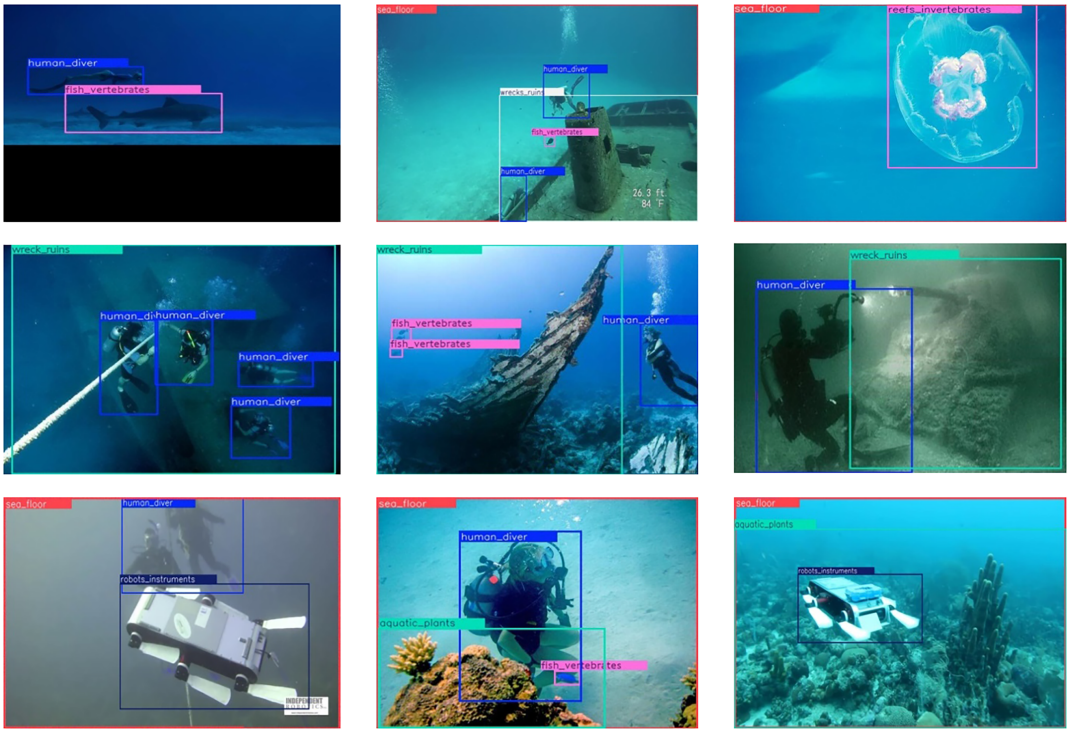

Figure 3: The qualitative detection results of the proposed model on six sample images from the SUIM test set. Bounding boxes show predicted objects with class labels and confidence scores.

Table 3: The per-class detection performance (mAP@0.5) on the SUIM dataset.

Class	mAP@0.5
Fish_vertebrates (FV)	85 ± 0.54
Wreck_ruins (WR)	84 ± 0.41
Human_diver (HD)	86 ± 0.34
Aquatic plants/flora (PF)	88 ± 0.54
Robots and instruments (RO)	86 ± 0.50
Reefs and other invertebrates (RI)	89 ± 0.64
Macro mAP@0.5	86.33 ± 0.49

Table 4: The quantitative performance of the proposed model on the SUIM dataset. Results are reported as mean ± standard deviation over five independent runs.

Class	Precision%	Recall%	F1 Score%
Fish_vertebrates (FV)	93.14 ± 0.30	88.20 ± 0.60	89.20 ± 0.35
Wreck_ruins (WR)	88.60 ± 0.42	86.31 ± 0.55	87.12 ± 0.40
Human_diver (HD)	92.00 ± 0.28	87.24 ± 0.50	89.21 ± 0.30
Aquatic plants/flora (PF)	87.67 ± 0.51	86.00 ± 0.65	88.81 ± 0.45
Robots and instruments (RO)	86.13 ± 0.48	85.45 ± 0.70	88.00 ± 0.50
Reefs and other invertebrates (RI)	87.20 ± 0.39	86.32 ± 0.58	90.11 ± 0.32
Macro-Avg	88.83 ± 0.39	86.33 ± 0.59	88.50 ± 0.38

For fair and direct comparison, we reimplemented Wang et al. [8] and AIT-YOLOv7 [10] on the SUIM detection benchmark using identical training and evaluation protocols. Where all methods were trained on the same SUIM train/test split (80/20), with an input resolution of 640 × 640, a batch size of 16, a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.01, and 300 epochs. Every method was run five times with different random seeds (42, 123, 456, 789, 101,112), and all results are reported as mean ± standard deviation. The YOLOv8-nano baseline represents the standard detector without our proposed enhancements.

Table 5 shows that the proposed method attains the highest mAP@0.5 of 86.33%, outperforming Wang et al. [8] by 2.10 percentage points, AIT-YOLOv7 [10] by 2.93 percentage points, and the baseline YOLOv8-nano by 6.20 percentage points. The improvements are statistically significant over five independent runs. In terms of precision, our method achieves 88.83%, which is 0.33% higher than Wang et al. (88.50%) and substantially higher than AIT-YOLOv7 (82.10%). Also, the recall of the suggested framework (86.58%) exceeds all reimplemented baselines. These results indicate that the synergistic integration of MSRCR enhancement, ResNet-50 with CBAM attention, and LightGBM refinement provides significant improvements over existing YOLO-based underwater detectors when evaluated under identical conditions on the SUIM benchmark.

Table 5: Comparison with reimplemented YOLO-based underwater object detectors on the SUIM detection benchmark. All methods were trained and evaluated under identical conditions. Results reported as mean ± standard deviation over five independent runs.

Method	mAP@0.5 (%)	Precision (%)	Recall (%)
YOLOv8-nano (baseline)	80.13 ± 0.45	83.23 ± 0.63	83.23 ± 0.63
Wang et al. (YOLOv7 + enhancement) [8]	84.23 ± 0.90	88.5 ± 0.42	85.5 ± 0.42
AIT-YOLOv7 [10]	83.40 ± 0.80	82.10 ± 0.90	82.10 ± 0.90
Proposed method	86.33 ± 0.49	88.83 ± 0.39	86.33 ± 0.59

5.1 The Complexity of the Proposed Model

In this section, a comprehensive analysis of model complexity, including parameters and inference time, along with an analysis of the design choices, has been introduced. Table 6 presents the parameter breakdown for each component of our multi-stage framework.

Table 6: The parameter for each component of the proposed multi-stage framework.

Component	Approximate Parameters
MSRCR Enhancement	0
YOLOv8-nano	~3,200,000
ResNet-50	~25,600,000
CBAM modules	~86,000
Feature Fusion Modules (1 × 1 convs)	~412,000
LightGBM	~0.13 million (500 trees × 63 leaves)
TOTAL	~29, 30 million parameters

Table 7 demonstrates the complexity comparison with standard object detectors. The table indicates that the proposed method (29.3 M parameters, 38.5 ms, 26 FPS) occupies a middle ground between lightweight single-stage detectors (YOLOv8-nano: 3.2M, 11.3 ms, 88 FPS) and heavier two-stage detectors

(Faster R-CNN: ~41M, ~50 ms, ~20 FPS). The additional computational cost over YOLOv8-nano comes from three sources: (1) the parallel ResNet-50 branch (+25.6M parameters, +10.2 ms), (2) CBAM and feature fusion modules (+0.5M parameters, +3.3 ms), and (3) LightGBM refinement (+0.13M parameters, +7.5 ms). These overheads are justified by the accuracy gains shown in Table 5 (+6.2% mAP over YOLOv8-nano).

Table 7: Complexity comparison with standard object detectors.

Model	Parameter (M)	Inference Time (ms)	FPS
YOLOv8-nano (baseline)	3.2	11.3	88
YOLO8-small	~11.2	~15	~66
Faster R-CNN (ResNet-50)	~41	~50	~20
Proposed method (total)	29.3	38.5	26

5.2 Inference Time

The primary tests demonstrate that, despite its complexity, the proposed model performs within the reasonable limits of real-time applications. Table 8 shows the Inference Time of each of the stages of the proposed manuscript.

Table 8: The Inference time and FPS for each stage in the proposed manuscript.

Processing Stage	Time (ms)	Frames per Second (FPS)
Stage 1: MSRCR Enhancement	6.2	161
Stage 2: Augmented Detection Network	24.8	40
YOLOv8-nano	11.3	88
ResNet-50 forward pass	10.2	98
CBAM + Feature Fusion	3.3	303
Stage 3: Refinement	7.5	133
Feature vector extraction	5.8	172
LightGBM inference (per proposal)	0.002	500,000
Post-processing (filtering NMS)	1.7	588
Total pipeline	**38.5**	**26.0**

The complete pipeline achieves 26 FPS, which is sufficient for near-real-time underwater applications (typical requirements: 15–30 FPS for ROV navigation, 5–10 FPS for survey missions) [25].

Our multi-stage framework achieves 26 FPS, which is lower than the YOLOv8-nano baseline (88 FPS) but higher than two-stage detectors such as Faster R-CNN (~20 FPS). The additional computational cost comes from three sources: (1) MSRCR enhancement (6.2 ms), (2) parallel ResNet-50 forward pass (10.2 ms), and (3) LightGBM feature extraction and inference (7.5 ms). These overheads are justified by the accuracy gains shown in Table 5 (+6.2% mAP over YOLOv8-nano). For applications that require higher speed, the LightGBM refinement stage can be removed (reducing FPS to ~32) at the cost of lower precision. We do not claim real-time deployment on severely resource-constrained platforms (e.g., low-power AUVs), but 26 FPS is sufficient for many ROV-based survey missions where a standard GPU is available.

Feature vector extraction (5.8 ms) includes: cropping features from YOLOv8 feature maps (P3, P4, P5), cropping from ResNet-50 stage outputs, computing attention statistics from CBAM, and aggregating contextual features. This is implemented efficiently using RoIAlign-style cropping.

5.3 Ablation Study

To verify the effect or contribution of each component to the proposed work, comprehensive excision experiments were performed on the SUIM dataset. All models were trained under identical conditions: the input size was 640 × 640, batch size 16, an SGD optimizer with an initial learning rate of 0.01, and 300 epochs. Also, to account for statistical variance, each case was run 5 times with different random seeds, and the mean and standard deviation across all included measures were reported. Table 9 reveals the results of the ablation study on the SUIM dataset. From Table 9, these gains are not strictly independent, as components may interact (e.g., CBAM attention operates on ResNet-50 fused features, and LightGBM refines detections from the full detector). The sequential addition order shown in Table 9 reflects cumulative improvement, but the precise contribution of each component in isolation cannot be fully disentangled due to these interactions. Therefore, the attributions above should be interpreted as observational rather than causal contributions.

Table 9: Ablation study results on the SUIM dataset (mean ± std over five runs).

Configuration	mAP@0.5%	Precision %	Recall %
BaselineYOLOv8-nano	78.3 ± 1.2	76.8 ± 1.3	74.2 ± 14
MSRCR Enhancement	81.5 ± 0.9	79.4 ± 1.0	77.1 ± 1.1
ResNet-50 feature extraction	84.2 ± 0.7	82.1 ± 0.8	79.8 ± 0.9
CBAM attention modules	85.7 ± 0.5	87.6 ± 0.6	84.3 ± 0.7
LightGBM reninement (full model)	86.33 ± 0.49	88.83 ± 0.39	86.33 ± 0.59

Fig. 4 demonstrates the results of the ablation study. It isolates the contribution of each component in the proposed multi-stage framework in terms of mAP@0.5, Precision, and Recall. Overall, the data show a monotonic improvement in performance as each new module is added to the baseline, providing strong evidence that each component of the proposed model contributes positively to the final result. The following is the analysis of component contributions.

An MSRCR enhancement increases the mAP at 0.50 by 3.2 percentage points (pp) over baseline, mitigating the underwater color distortion and low contrast, and enabling better feature extraction.

The ResNet-50 fusion features illustrate 2.7 pp improvement, confirming that high-level semantic features provided by a trained classification network are complementary to the YOLOv8 backbone, especially for object class differentiation in a cluttered underwater scheme.

The use of CBAM attention modules (1.5 pp) is related to the fact that they draw the network's attention to informative channels and the object's spatial location, while eliminating noise produced by the turbidity and background textures such as sand or aquatic plants.

The LightGBM refinement offers an additional improvement of 0.63 pp, suggesting that the trained 51-dimensional feature representation helps differentiate true detections and false positives.

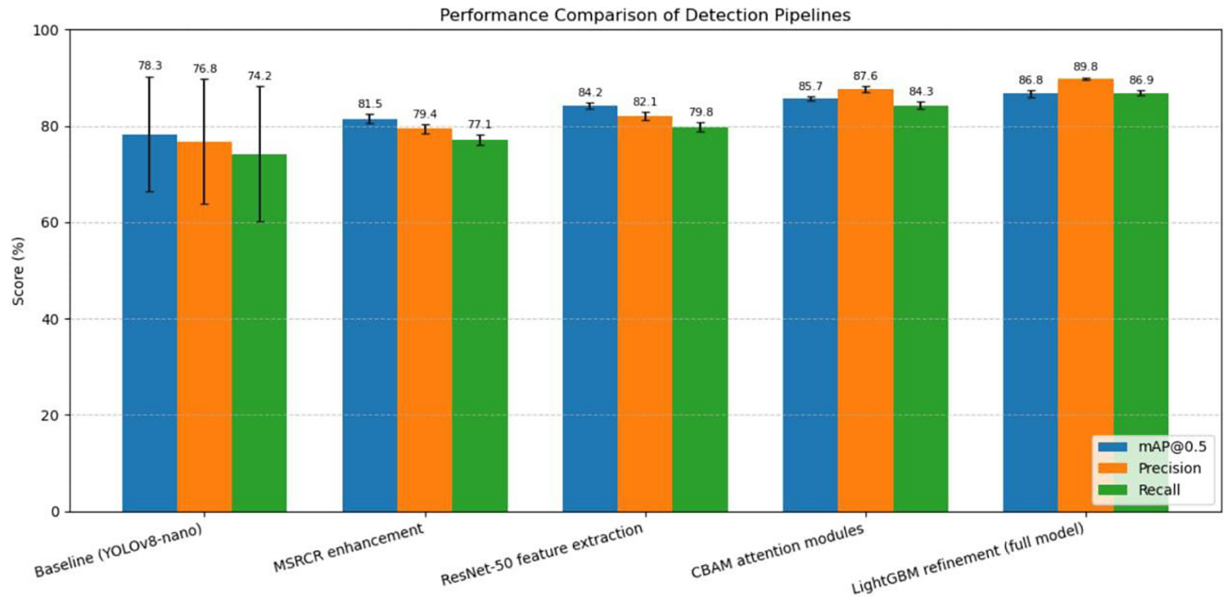

Figure 4: The ablation study results display the progressive improvement in mAP@0.5, precision, and recall as each component (MSRCR, ResNet-50, CBAM, LightGBM) is added to the YOLOv8-nano baseline. Results are mean values over five independent runs.

6 Conclusions and Future Works

This work presented a united multi-stage deep learning architecture that combines adaptive MSRCR enhancement, dual-branch feature extraction with ResNet-50 and CBAM, and LightGBM-based refinement for underwater object detection in challenging visual conditions. An adaptive Multi-Scale Retinex with Color Restoration (MSRCR) image enhancement module, followed by an augmented YOLOv8 detector (supplemented with a ResNet-50 backbone and CBAM attention modules), and a LightGBM classifier for detection refinement, are synergistically integrated into an integrated three-stage pipeline.

The comprehensive tests on the SUIM dataset established the efficiency of the proposed framework, achieving a macro-average mAP@0.5 of 86.33% ± 0.49, alongside a precision of 88.83% ± 0.39 and a recall of 86.33% ± 0.59. Additionally, the consistency and resilience of the suggested technique were verified by results from five independent runs that demonstrated stable performance with low standard deviations across all metrics. Accurate scores for important classes, including fish vertebrates (93.14) and Human divers (92.00), also indicate practical applicability in underwater contexts where false positives are costly.

Under identical reimplementation conditions on the SUIM dataset, the proposed model outperforms the state-of-the-art, confirming the effectiveness of the multi-stage framework.

The individual components of the suggested framework are established techniques in computer vision. So, the contribution lies in the integration of these techniques into a three-stage pipeline designed specifically for underwater imagery, using the systematic adaptation of the SUIM dataset for detection evaluation. The observed demonstration of improved performance over reimplemented YOLO-based baselines.

In spite of the powerful quantitative results, the study has several limitations that should be discussed. One of them is Computational Complexity: Although the multi-stage architecture is efficient and effective, it adds computational overhead due to the sequential processing of image enhancement, feature extraction, fusion, and LightGBM refinement, thus increasing the inference time. Therefore, it may be limited to real-time implementation on platforms with constrained resources, such as autonomous underwater vehicles

(AUVs) or remotely operated vehicles (ROVs). Also, although the SUIM dataset encompasses diverse underwater scenes, it represents a specific distribution of marine environments. A key limitation of this study is the absence of cross-dataset validation where the experiments were conducted on the SUIM dataset (which was adapted by adapted for detection), and we have not evaluated on other underwater benchmarks such as URPC or TrashCan. Therefore, claims about generalization to real-world deployment environments are preliminary. Second-dataset validation remains an important direction for future work.

For future work, a number of directions will be pursued to enhance practicality and performance. First, lightweighting techniques (such as network pruning or knowledge distillation) may be explored to reduce the computational cost of deploying the model in real time on autonomous underwater vehicles (AUVs). Second, more sophisticated schemes for fusing the YOLOv8 and ResNet-50 streams can be examined, which might require changing or attention-based weighting. Finally, it is possible to test the system's generalizability across a wider range of underwater datasets and in more extreme environments, such as deep-sea or turbid-river environments, to ensure its soundness further.

Acknowledgement: The author (Asmaa Sadiq) thanks the Department of Computer Science, College of Science, Mustansiriyah University, for supporting this work.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Conceptualization, Rana Lateef and Asmaa Abdul Jabbar; methodology, Rana Lateef and Asmaa Abdul Jabbar; software, Rana Lateef; validation, Asmaa Abdul Jabbar; formal analysis, Rana Lateef; investigation, Rana Lateef; resources, Asmaa Abdul Jabbar; data curation, Asmaa Abdul Jabbar; writing—original draft preparation, Asmaa Abdul Jabbar; writing—review and editing, Asmaa Abdul Jabbar and Rana Lateef; visualization, Rana Lateef and Asmaa Abdul Jabbar; supervision, Rana Lateef; project administration, Rana Lateef. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The SUIM dataset used in this study is available at: 1. https://github.com/xahidbuffon/SUIM. 2. https://huggingface.co/datasets/SatwikKambham/suim.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang M, Wang Z, Song W, Zhao D, Zhao H. Efficient small-object detection in underwater images using the enhanced YOLOv8 network. Appl Sci. 2024;14(3):1095. doi:10.3390/app14031095.
2. Zhou H, Kong M, Yuan H, Pan Y, Wang X, Chen R, et al. Real-time underwater object detection technology for complex underwater environments based on deep learning. Ecol Inform. 2024;82:102680. doi:10.1016/j.ecoinf.2024.102680.
3. Safa S, Sadiq A. Enhancement of underwater images using color correction and weight maps. Fusion Pract Appl. 2025;19(1):235–47. doi:10.54216/fpa.190118.
4. A survey of restoration and enhancement for underwater images. IEEE Access. 2019;7:182259–79.
5. Ji X, Chen L, Li X, Liu W. Multi-stage differential-aware attention network for real-time underwater salient object detection. J Real Time Image Process. 2025;23(1):18. doi:10.1007/s11554-025-01814-8.
6. Akhtarshenas A, Toosi R. An open-set framework for underwater image classification using autoencoders. SN Appl Sci. 2022;4(8):229. doi:10.1007/s42452-022-05105-w.
7. Yang J, Cai M, Yang X, Zhou Z. Underwater image classification algorithm based on convolutional neural network and optimized extreme learning machine. J Mar Sci Eng. 2022;10(12):1841. doi:10.3390/jmse10121841.

8. Wang Z, Zhang G, Luan K, Yi C, Li M. Image-fused-guided underwater object detection model based on improved YOLOv7. Electronics. 2023;12(19):4064. doi:10.3390/electronics12194064.
9. Dakhil RA, Khayeat ARH. Deep learning for enhanced marine vision: object detection in underwater environments. Int J Electr Electron Res. 2023;11(4):1209–18. doi:10.37391/ijeer.110443.
10. Pachaiyappan P, Chidambaram G, Jahid A, Alsharif MH. Enhancing underwater object detection and classification using advanced imaging techniques: a novel approach with diffusion models. Sustainability. 2024;16(17):7488. doi:10.3390/su16177488.
11. Zuo X, Jiang J, Shen J, Yang W. Improving underwater semantic segmentation with underwater image quality attention and muti-scale aggregation attention. Pattern Anal Appl. 2025;28(2):80. doi:10.1007/s10044-025-01460-7.
12. Wang H, Zhang Y, Hong Y, Wang X. Underwater object segmentation using improved DeepLabv3+. In: Proceedings of the Fourth International Conference on Machine Vision, Automatic Identification, and Detection (MVAID 2025); 2025 May 23–25; Wuhan, China. doi:10.1117/12.3077389.
13. Chen B, Zhao W, Zhang Q, Li M, Qi M, Tang Y. Semantic segmentation of underwater images based on the improved SegFormer. Front Mar Sci. 2025;12:1522160. doi:10.3389/fmars.2025.1522160.
14. Petro AB, Sbert C, Morel JM. Multiscale retinex. Image Process Line. 2014;4:71–88. doi:10.5201/ipol.2014.107.
15. Chen J, Gao Z, Huang C, Yang L. Underwater image enhancement algorithm based on Retinex and wavelet fusion. IOP Conf Ser Earth Environ Sci. 2020;615(1):012120. doi:10.1088/1755-1315/615/1/012120.
16. Parthasarathy S, Sankaran P. An automated multi Scale Retinex with color restoration for image enhancement. In: Proceedings of the 2012 National Conference on Communications (NCC); 2012 Feb 3–5; Kharagpur, India. p. 1–5. doi:10.1109/NCC.2012.6176791.
17. Huang J, Fang C, Zheng X, Liu J. YOLOv8-UC: an improved YOLOv8-based underwater object detection algorithm. IEEE Access. 2024;12:172186–95. doi:10.1109/ACCESS.2024.3496925.
18. Ding J, Hu J, Lin J, Zhang X. Lightweight enhanced YOLOv8n underwater object detection network for low light environments. Sci Rep. 2024;14:27922. doi:10.1038/s41598-024-79211-7.
19. Islam MJ, Edge C, Xiao Y, Luo P, Mehtaz M, Morse C, et al. Semantic segmentation of underwater imagery: dataset and benchmark. In: Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2021 Oct 24–Jan 24; Las Vegas, NV, USA. p. 1769–76. doi:10.1109/IROS45743.2020.9340821.
20. Yerram V, Takeshita H, Iwahori Y, Hayashi Y, Bhuyan MK, Fukui S, et al. Extraction and calculation of roadway area from satellite images using improved deep learning model and post-processing. J Imag. 2022;8(5):124. doi:10.3390/jimaging8050124.
21. Adam JM, Liu W, Zang Y, Afzal MK, Bello SA, Muhammad AU, et al. Deep learning-based semantic segmentation of urban-scale 3D meshes in remote sensing: a survey. Int J Appl Earth Obs Geoinf. 2023;121:103365. doi:10.1016/j.jag.2023.103365.
22. Xu D, He Y, Su J, Qiu L, Lin L, Zheng J, et al. An ultra-lightweight and high-precision underwater object detection algorithm for SAS images. Remote Sens. 2025;17(17):3027. doi:10.3390/rs17173027.
23. Mahara A, Khan MRK, Deng L, Rishe N, Wang W, Sadjadi SM. Automated road extraction from satellite imagery integrating dense depthwise dilated separable spatial pyramid pooling with DeepLabV^{3+}. Appl Sci. 2025;15(3):1027. doi:10.3390/app15031027.
24. Kumar S, Sur A, Baruah RD. DatUS: data-driven unsupervised semantic segmentation with pretrained self-supervised vision transformer. IEEE Trans Cogn Dev Syst. 2024;16(5):1775–88. doi:10.1109/TCDS.2024.3383952.
25. Li W, Zhang F. Real-time vision–language analysis for autonomous underwater drones: a cloud–edge framework using Qwen2.5-VL. Drones. 2025;9(9):605. doi:10.3390/drones9090605.