Improving YOLOv8m with Neck-Integrated Atrous Spatial Pyramid Pooling for Enhanced Detection of Small Fish and Jellyfish

Shaymaa K. Hussein, Zainab N. Nemer
College of Computer Science and Information Technology, University of Basrah, Iraq
E-Mail: shaimaa.khudher@uobasrah.edu.iq, Zainab.nemer@ uobasrah.edu.iq

Keywords: YOLOV8M, ASSP, underwater object detection

Received: March 19, 2025

The ocean depths are vital for biodiversity, as they host numerous marine species essential for maintaining ecosystem health. Accurate identification of aquatic creatures is critical for developing effective conservation strategies and sustainable marine resource management. However, aquatic environments pose distinct challenges, including light scattering, occlusions, and turbidity. This paper offers an improved YOLOv8m architecture that integrates ASPP module (Atrous Spatial Pyramid Pooling) that enhances multi scale feature extraction. Our proposed model, YOLOv8m-ASPP, was evaluated against the baseline YOLOv8m on the Brackish dataset, which contains 8,417 annotated images across six marine categories ("crab, jellyfish, fish, shrimp, small_fish, and starfish").
The key architectural innovation involves integrating the ASPP module, with dilation rates of [2, 4, 6], into YOLOv8m's neck, specifically after the SPPF layer. This placement allows the ASPP module to process rich contextual features from the backbone, improving the ability of the model to capture objects at various scales. The YOLOv8m-ASPP model achieved an overall mAP@50 of 0.991 (a +0.002 increase) and a mAP@50-95 of 0.832 (a +0.004 increase) compared to the baseline YOLOv8m's 0.989 mAP@50 and 0.828 mAP@50-95. The modified model showed a precision of 0.980 and recall of 0.979., operating at approximately 60 FPS. Performance notably improved for challenging classes: the 'jellyfish' class mAP@50-95 rose to 0.757 (from the baseline's 0.730). Furthermore, robustness in small object detection was evident, with the 'small_fish' class achieving 0.970 mAP@50 (up from the baseline's 0.960 mAP@50). The findings demonstrate the effectiveness of the YOLOv8m-ASPP model for underwater ecological monitoring, successfully maintaining both detection accuracy and real-time processing capabilities. Future research could explore improved detection methods for small objects in environments with high turbidity.

Povzetek: Članek izboljšuje YOLOv8m z integracijo ASPP za izboljšano zaznavanje majhnih rib in meduz v podvodnih okoljih. Model ADMT doseže boljšo točnost in robustnost ob hkratnem ohranjanju obdelave v realnem času.

1 Introduction

The underwater object detection (UOD) Become a crucial computer vision technology for marine exploration and monitoring[1]. Its applications are diverse, ranging from ecological research and biodiversity protection to enhancing security, aiding autonomous underwater vehicle (AUV) navigation, and supporting underwater archaeology and search and rescue operations[2, 3]. UOD offers a non-invasive means of gathering crucial data from marine ecosystems, minimizing anthropogenic impact[4] However, the efficacy of UOD systems is often hindered by significant underwater challenges, including light attenuation, turbidity-induced visual degradation, and dynamic environmental conditions that impair detection accuracy[5, 6].

Considering these restrictions, this study suggests and evaluates a novel hybrid deep learning architecture that strategically integrates the ASPP module, known for its effectiveness in multi-scale feature extraction[7], with the robust YOLOv8 m detection framework. The primary research objective is to investigate whether this integration can achieve measurable improvements in detection accuracy and robustness, particularly under conditions of poor visibility, while striving to maintain real-time processing capabilities (e.g., targeting >30 FPS). Specifically, this work aims for a notable enhancement mean average precision (mAP@50-95) compared to the traditional Yolov8m baseline. Emphasis is placed on scenarios demanding both high accuracy (e.g., targeting >95% recall for critical detections) and low latency (e.g., <50ms inference time), pertinent to applications such as AUV path planning and the detection of illegal fishing activities [8].

The proposed architecture is designed to overcome key limitations in existing UOD systems. Utilizing ASPP with its multi-scale feature analysis capabilities, the model seeks to make it easier to identify partially occluded objects or objects with variable sizes commonly found in underwater turbid scenes. In addition, inspired by the efficient design in YOLOv8m, architecture tries to maintain computational efficiency for real-time running on underwater robotic platforms. A systematic evaluation of the model's performance will be assessed. consistency across varied underwater conditions. This investigation is particularly focused on enhancing the detection of small targets, a persistent challenge in UOD.

This work has the following contributions: (1) It proposes a new hybrid structure that combines the multi-scale feature learning ability of ASPP with the efficient detection framework of YOLOv8m to the UOD problem. (2) We can present an in-depth performance evaluation of this architecture over the baseline systems on the Brackish dataset by showing improvements in standard metrics. (3) It studies the real-time processing of this model. (4) It offers insight into the model's efficacy in detecting hard-to-detect objects (e.g., small or partially occluded marine life), and then directs future work for UOD. The results are expected to be highly applicable to operational needs in marine conservation, underwater structure survey, and autonomous navigation, where robust object recognition in unfavorable conditions is vital.

2 Related work

2.1 Introduction

This section highlights the primary challenges facing UOD while discussing related work.It also investigates the development of object detection approaches, particularly deep learning-based methods, such as the You Only Look Once (YOLO) family of methods, analyses the ways of improving feature representation and performance in real time. In this research, we concentrate on the enhancement of YOLOv8m for UOD using ASPP.

2.2 Key challenges in underwater object detection (UOD)

Underwater habitats bring distinct challenges, set of conditions for object detection. Poor visibility, stemming from light attenuation and scattering, significantly degrades image quality. This is often compounded by issues including low contrast and color distortion. which traditional image enhancement techniques attempt to address [9, 10]. Furthermore, object ambiguity due to marine organisms blending with complex backgrounds, detecting small or distant objects can be challenging, especially in the presence of turbidity or marine particulates further complicate reliable detection[11].Beyond these visual impediments, UOD efforts are also constrained by dataset limitations, as publicly available underwater datasets are often small, imbalanced, or lack diverse annotations, hindering the advancement of resilient and universally applicable models [12]. Finally, the computational resources available on autonomous underwater platforms are typically restricted, necessitating efficient algorithms that can perform accurately in real-time without excessive power consumption[13].

2.3 Object detection's evolution techniques and their application in UOD

2.3.1 Traditional object detection

Traditional object detection frameworks, such as the Viola-Jones Detector and feature extraction, often paired with machine learning techniques like SVM and Random Forests for high accuracy. In underwater image enhancement, histogram-based methods (e.g., HE, A HE, CLAHE) and Retinex-based methods (e.g., MSR, MSRCR) address challenges like color distortion and low contrast but face trade-offs in computational complexity and real-time performance. While these methods improve image quality, further advancements are needed to enhance robustness and efficiency in challenging underwater environments.[9, 10].

2.3.2 The development of deep learning at UOD

Deep learning-based object detection methodologies employ Convolutional Neural Networks (CNNs) to autonomously extract features from images, replacing traditional methods and improving generalization and robustness[10]. Advances in hardware, such as GPUs, have addressed challenges like large data demands and long training times. Object detection involves classifying and localizing objects using bounding boxes. Detectors can be either two or single stage. Two-stage detectors, such as R-CNN and Faster R-CNN, create regions of Interest (ROIs) and then classify, offering higher accuracy but with complex architectures and slower inference. Detectors like YOLO and SSD predict class and bounding box probabilities in a single step, providing simpler, faster, and more efficient solutions, albeit with slightly lower accuracy. YOLO is popular for its simplicity, speed, and real-time capabilities, balancing accuracy and efficiency for practical applications[14].

2.3.3 The YOLO series and its application in UOD

Redmon et al.'s 2016 introduction transformed real time object detection into a regression problem, revolutionizing the field. Divide the input image into a grid, then estimate bounding boxes and probability of classes for each cell. (see Figure 1).

.

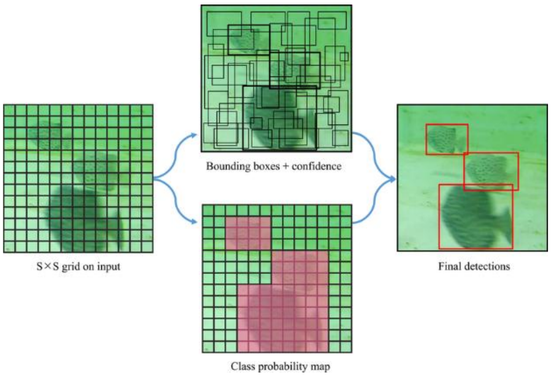


Figure 1: Principle of the YOLO algorithm [15]

Subsequent versions introduced significant enhancements: YOLOv2 enhances accuracy and recall using anchor boxes and multi-scale training; YOLOv3 incorporated multi-scale predictions via a deeper backbone and a Feature Pyramid Network (FPN)-like structure; YOLOv4 and YOLOv5 focused on optimizing network architecture, data augmentation, and training techniques for better performance and efficiency. More recent iterations like YOLOv6 and YOLOv7 further refined efficiency with concepts such as task-aligned heads and “Extended Efficient Layer Aggregation Networks (E-ELAN)”, while approaches like DAMO-YOLO explored Neural Architecture Search (NAS) for dynamic structures. This has been marginalized in YOLOv8, the baseline for our study, which featured an anchor-free design, an optimized C2f module and decoupled head while extending from single-task learning such as translation to multi-task learning from segmentation and pose estimation, further establishing YOLO as the de-facto framework for real-time object detection[16].

YOLO versions There have been adapted several YOLO versions in the UOD domain. For example, one of such adaptations which relied on YOLOv4 architecture to balance detection accuracy and computational speed was able to achieve competitive results on the challenging underwater datasets. This specific model achieved a Mean Average Precision (mAP) equal to 81.67% on PASCAL VOC dataset and 92.65% on brackish water datasets. It also retained a processing speed that is appropriate for near real-time application, clocking at 44.22 frames per second (FPS) and succeeded in detecting underwater objects even when faced with low visibility and color lied [17] Acknowledging that, for resource-limited devices, even faster inference may be required, lightweight models such as YOLOv4-tiny are also investigated for UOD. But, despite their speed, YOLOv4-tiny is not perfect with detection accuracy. To overcome this, we propose an approach that leverages YOLO-UOD on top of YOLOv4-tiny. YOLO-UOD is reported to achieve a dramatically higher mAP of 87.88% on the Brackish underwater dataset than the 77.38% of YOLOv4-tiny and even more than YOLOv5s and YOLOv5m with faster speeds and thus balancing the speed and accuracy well[18]. In the same veins, there are also attempts to improve the previous foundational methods YOLOv3 for this specific domain. in proceedings was used as a base configuration since YOLOv3 had been previously demonstrated to be stable in a wide range of Open Images and was selected as one of the strongest single object detection model for the most difficult object classes. This work specifically tuned U-YOLOv3 for underwater object detection and reported a 2-10% higher mean Average Precision (mAP) than the original YOLOv3 on both the Brackish and Trash ICRA19 datasets[19]. Other approaches have focused on newer variants like YOLOv8 or specialized models such as LFN-YOLO, which is optimized for underwater environments. The LFN-YOLO model, for example, achieved a mean Average Precision (mAP) of 82.1% on the URPC dataset and an impressive 97.5% on the Brackish dataset, indicating its high detection accuracy in these specific contexts[20]. Similarly, YOLOv7 has been selected as a base for underwater target detection enhancements due to its advanced capabilities in Real-time applications require both accuracy and speed is the YOLOv7-AC mode, leveraged YOLOv7's architecture to improve feature extraction and overall performance in challenging underwater conditions. The YOLOv7-AC model attained an average precision (mAP) of 89.6% on the URPC dataset.and 97.4% on the Brackish dataset. Notably, it demonstrated improved detection efficiency, particularly in specific categories like echinus, which reached the average precision (AP) of 92.2%. This enhanced model reportedly outperformed other popular target detection models, showing an improvement of 1.1% over the original YOLOv7 and even greater gains when compared to YOLOv6, YOLOv5s, and SSD [21].

2.4 Addressing specific UOD challenges with advanced techniques

2.4.1 Enhancing feature representation and multi-scale detection

The variable sizes of underwater objects and visibility degradation necessitate robust multi-scale feature representation. Feature Pyramid Networks (FPNs) are widely adopted for this. Zhang et al. (2021) proposed a Modified Attentional Feature Fusion (AFFM) for the FPN structure in their lightweight YOLOv4, based on a" Multi-Scale Channel Attention Module (MS-CAM)”, to better fuse semantic and scale-inconsistent features for small underwater targets[17]. Alongside these CNN-based enhancements, alternative architectures such as DEtection TRansformers (DETR) have been investigated for their ability to utilize global contextual information in UOD. For instance, Ali et al. employed DETR for marine object detection, utilizing a ResNet50 backbone to produce multi-resolution feature maps, thereby enhancing the model's capability to detect objects in complex underwater environments. Their DETR model was fine-tuned on the Brackish dataset to specifically address challenges like low visibility and complex backgrounds. This transformer-based approach achieved a significant mean Average Precision (mAP) of 0.648 on the test set, reportedly outperforming other detectors like YOLOv3 (which achieved mAP of 0.3893 on the same dataset) in their study. Such findings suggest that transformer-based models offer considerable potential for enhancing detection accuracy in challenging underwater scenarios, paving the way for further advancements in marine object detection technologies by leveraging their distinct approach to feature representation and global context understanding.[1] Further advancing feature learning and fusion, Tian et al. (2024) utilized YOLOX as a baseline and introduced a novel Level, Channel, Spatial (LCS) multi-attention module. The core innovation of their work lies in the LCS module's approach to feature fusion, which is designed to significantly improve the model's learning ability by compelling it to focus on critical scale, channel, and spatial aspects of the features. This enhanced feature representation and contextual understanding resulted in their model achieving a mean Average Precision (mAP) of 77.32%[22]. Similarly, Liu et al. (2025), they have also designed their LFN-YOLO model based on the YOLOv8n model to concentrate on robust feature fusion for accommodating the different scales of objects under water. In the neck part of the model, the shape decoder used a GFPN to powerfully fuse the geometric details combining low-level feature maps and the abundant semantic information derived from high-level feature maps. The aim of this GFPN embedding was to enhance the network's adaptability to features of varying scales and small object detection by efficiently propagating information across different levels of features using reparametrized elements. Their LFN-YOLO obtains a mAP@0. 5 of 97.5% on the Brackish dataset, demonstrating the effectiveness of their feature fusion scheme for UOD[20]. Atrous Spatial Pyramid Pooling (ASPP) is a special powerful tool to capture long-range context with multi-scale information and no resolution reduction, which is very important for both semantic segmentation and object detection. The ASPP module which was inspired in DeepLab series exploits parallel atrous (dilated) convolutions with different rates to cover multi-scale context and scale of objects to enlarge the receptive field and gather various contextual information. It is an essential ability-skill in remote sensing and consequently in demanding UOD scenarios, where objects can be seen in very different sizes and environments. Motivated by these advantages, Hu et al. (2024) published ASPP+-LANet for high-resolution remote sensing image segmentation, in which an advanced ASPP module (denoted ASPP+) is proposed according to the original ASPP by adding on an another feature extraction path, reconsidering the dilation rates, and integrating a Coordinate Attention (CA) mechanism. Their ablation studies showed that including their ASPP+ module dramatically enhanced segmentation results of ground object targets in different scales[23]. Further, Sivanpillai et al. (2023) indicated that ASPP module in DeepLabV3+ can remarkably improve the performance of water body segmentation, even for the small and partly occluded ones. Their research suggested that multi-scale feature extraction of ASPP significantly satisfied the requirements for the model to predict the water boundaries and details of water bodies with different sizes since DeepLabV3+ with ASPP far outperformed versions without it. These results together confirm the performance applicability of ASPP to complex tasks containing multi-scale features and occlusions, and similar phenomena appear to exist in the underwater object detection, which constitutes a strong foundation of our proposed architectural modification [7].

2.4.2 Lightweight models and real-time performance in UOD

Operational constraints of AUVs necessitate computationally efficient models. Zhang et al. (2021) focused on this by adapting YOLOv4 with a MobileNetV2 backbone and depth-wise separable convolutions, achieving over 44 FPS with a significantly reduced model size[17]. LFN-YOLO (Liu et al., 2025), based on YOLOv8n, used RepGhost and SPD-Conv, resulting in a 5.9MB model with 58-63 FPS on Brackish[20]. Tian et al. (2024) used YOLOX with a GhostNet backbone and an LCS attention module, achieving a 18.5MB model and 55.54 FPS on URPC2021[22]. These efforts highlight the ongoing pursuit of balancing detection accuracy with computational feasibility.

The reviewed literature demonstrates significant advancements in Underwater Object Detection (UOD), largely driven by the growth of deep learning approaches. Convolutional Neural Networks, notably the You Only Look Once (YOLO) series, have become foundational because of their mix of speed and accuracy, with various iterations being adapted for the challenging aquatic domain. Researchers have explored various strategies to mitigate the adverse effects of underwater conditions, including poor visibility, color distortion, and turbidity. These include advanced image enhancement pre-processing, specialized backbone networks (e.g., MobileNetV2, RepGhost, and GhostNet,) various feature fusion mechanisms, such as enhanced Feature Pyramid Networks (FPNs) and attention modules (e.g., GAM, AFFM), to improve multi-scale detection and robustness. The pursuit of lightweight models for deployment on resource-constrained underwater platforms has also been a prominent theme.

Despite these efforts, persistent challenges remain. Many existing models, while improving general UOD performance, still struggle with the reliable detection of small, occluded, or camouflaged objects, particularly under varying turbidity and illumination levels. For instance, LFN-YOLO, despite its lightweight design and high accuracy, can still face issues with false positives and missed detections in highly variable environments and with significant object scale changes. Similarly, models like YOLOv7-AC may exhibit errors in highly complex underwater scenes. While techniques like specialized detection heads and dedicated feature extractors for small objects show promise, achieving consistent high performance across diverse underwater conditions remains an objective. Furthermore, while context aggregation modules like Atrous Spatial Pyramid Pooling (ASPP) have proven highly effective for capturing multi-scale information in fields like semantic segmentation, and can enhance object detectors, their optimal integration and impact within the neck of recent, efficient YOLO architectures like YOLOv8m—specifically for enhancing the detection of problematic marine classes such as 'small_fish', 'jellyfish', while maintaining robust real-time processing capabilities—are not yet fully explored in the UOD literature. The trade-off between added Computational complexity from such modules and the achievable accuracy gains, especially for real-world deployment on AUVs, also warrants careful investigation. This study, therefore, aims to address this gap by proposing and evaluating a YOLOv8m architecture enhanced with a strategically integrated ASPP module.

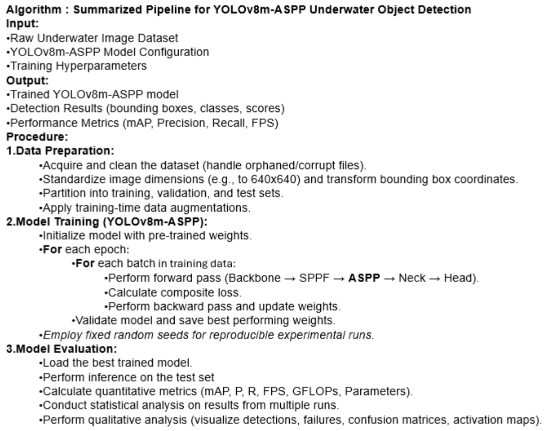
The research investigates the efficacy of this approach in improving multi-scale feature representation and contextual understanding, with a specific focus on enhancing detection accuracy and robustness for challenging underwater targets, while critically assessing the impact on real-time performance. To further consolidate the reviewed literature and provide a direct comparative overview, Table 1 summarizes key characteristics and performance metrics of several prominent object detection models discussed, particularly those adapted for or relevant to the underwater domain.

Table 1: Summary of selected underwater object detection models from related work (primarily on brackish dataset)

Source (Placeholder)	Model	Base Architecture	Key Modifications/Focus	Dataset(s) Used	Precision	Recall	mAP@0.5	mAP@0.5:0.95	FPS	Key Limitations Highlighted
Ali et al. [1]	YOLOv4 (general reference)	YOLOv4	Standard architecture	Brackish	-	-	0.9356	(not specified with mAP50 in Ali et al.)	-	(General challenges of UOD, less optimized than specific UOD variants)
Zhao et al. [18]	YOLO-UOD	YOLOv4-tiny	Symmetric dilated convolution module, Symmetric FPN-Attention module, Label smoothing	Brackish	-	-	-	0.8788	9.24 (Jetson Nano)	Some inference speed loss vs YOLOv4-tiny; needs more scene validation.
Sarkar et al [19]	U-YOLOv3	YOLOv3	MIRNet enhancement, K-means++ anchors, SPP layer, modified classification layers.	Brackish, Trash	0.88 (Brackish, 8k iter.)	0.87 (Brackish, 8k iter.)	-	~0.80 (Brackish, inferred from 10% gain over YOLOv3's ~0.72)	- (YOLOv3 was ~104 FPS on server)	Higher training time; 63.25M parameters. Lacks in small/dense objects compared to its own enhancements.
Liu et al [20]	LFN-YOLO	YOLOv8n	RepGhost, SPD-Conv, Generalized FPN (GFPN), CLLAHead, DFL loss. Focus on lightweight, small object detection, feature fusion.	Brackish, URPC	0.974	0.954	0.975	0.798	63 (Jetson AGX Orin implied)	Faces FPs/FNs in highly variable environments; issues with very small objects in complex backgrounds or significant scale changes.
Liu et al. [21]	YOLOv7-AC	YOLOv7	E-ELAN Enhanced, ResNet-ACmix Added, GAM Incorporated, K-means++ Anchors	Brackish,	98.2	95.2	97.4	73.7	-	not suitable for real-time, Robust multi-scale feature representation is needed
Ali et al. [1]	DETR	Transformer	ResNet50 backbone, fine-tuned for UOD. Focus on global contextual information.	Brackish	-	-	0.951	0.648	-	Evaluated on limited classes; transformers can be computationally heavy.

3 Methodology

This section delineates the research methodology employed to develop and evaluate the proposed enhanced object detection model for underwater environments. It details the baseline architecture, the proposed architectural modifications involving Atrous Spatial Pyramid Pooling (ASPP), the dataset utilized, experimental setup, training procedures, and the metrics employed for performance evaluation. The overall pipeline of our proposed approach, from data input to final evaluation, is summarized in Algorithm as illustrated in **Figure 2.**



```
Algorithm : Summarized Pipeline for YOLOv8m-ASPP Underwater Object Detection
Input:
•Raw Underwater Image Dataset
•YOLOv8m-ASPP Model Configuration
•Training Hyperparameters
Output:
•Trained YOLOv8m-ASPP model
•Detection Results (bounding boxes, classes, scores)
•Performance Metrics (mAP, Precision, Recall, FPS)
Procedure:
1.Data Preparation:
    •Acquire and clean the dataset (handle orphaned/corrupt files).
    •Standardize image dimensions (e.g., to 640x640) and transform bounding box coordinates.
    •Partition into training, validation, and test sets.
    •Apply training-time data augmentations.
2.Model Training (YOLOv8m-ASPP):
    •Initialize model with pre-trained weights.
    •For each epoch:
        •For each batch in training data:
            •Perform forward pass (Backbone → SPPF → ASPP → Neck → Head).
            •Calculate composite loss.
            •Perform backward pass and update weights.
        •Validate model and save best performing weights.
    •Employ fixed random seeds for reproducible experimental runs.
3.Model Evaluation:
    •Load the best trained model.
    •Perform inference on the test set
    •Calculate quantitative metrics (mAP, P, R, FPS, GFLOPs, Parameters).
    •Conduct statistical analysis on results from multiple runs.
    •Perform qualitative analysis (visualize detections, failures, confusion matrices, activation maps).
```

Figure 2: Summary algorithm of the model pipeline.

3.1 Overview of the proposed approach

The primary aim of this research is to improve the performance of the YOLOv8m model for UOD, focusing particularly on hard-object detection in varying visibility conditions. The fundamental approach is to incorporate an ASPP module in the YOLOv8m pipeline. This adaptation aims to improve the ability of the model to extract features at various scales and to understand context, thereby improving detection accuracy and robustness without unduly compromising its real-time processing capabilities.

3.2 Baseline architecture: YOLOv8m

The YOLOv8m model, developed by Ultralytics, serves as the baseline for this study. This single-stage object detector is highly accurate and fast, making it a cutting-edge technology. Key architectural components of YOLOv8m include:

- A backbone network, drawing inspiration from CSPDarknet, which utilizes C2f (CSP Bottleneck with 2 convolutions) modules for efficient feature extraction at various scales.
- A Spatial Pyramid Pooling Fast (SPPF) module located at the termination of the backbone, designed to aggregate contextual information by pooling features at different scales with minimal computational overhead.[24]
- A Path Aggregation Network (PANet)-inspired neck structure that facilitates effective fusion of features from different backbone levels (e.g., P3, P4, P5) to enhance multi-scale feature representation.

An anchor-free, the decoupled Bounding boxes and class are predicted by the detective head independently, which contributes to Faster convergence and better performance [25]

3.3 Proposed YOLOv8m-ASPP architecture

To address the challenges of UOD, particularly the detection of objects of varying sizes and those in cluttered or low-visibility scenes, we propose an enhanced architecture, hereafter referred to as YOLOv8m-ASPP. This architecture incorporates an ASPP module, a technique proven effective in semantic segmentation for capturing rich contextual information at multiple scales without significant loss of spatial resolution.

3.3.1 Atrous Spatial Pyramid Pooling (ASPP) Module

Integrated the ASPP module into our model is designed to Consider The entering convolutional feature layer includes numerous parallel atrous (dilated) convolutions with Various dilation rates. This allows the model to capture contextual information via a broader range of sources.the ASPP block comprises:

(1) One 1x1 convolutional branch. (2) Three 3x3 atrous convolutional branches with dilation rates of 2, 4, and 6, respectively. Each of these parallel branches processes the input feature map generates an intermediate feature map with [Specify, e.g., 192] channels. The outputs from these four branches are combined along a channel dimension. Finally, the combined features are fused the multi-scale information and refine the output using a 1x1 convolution., resulting in a feature map with [Specify, e.g., 768] channels, consistent with the input channel dimension to this block, as illustrated in **Figure 3.**

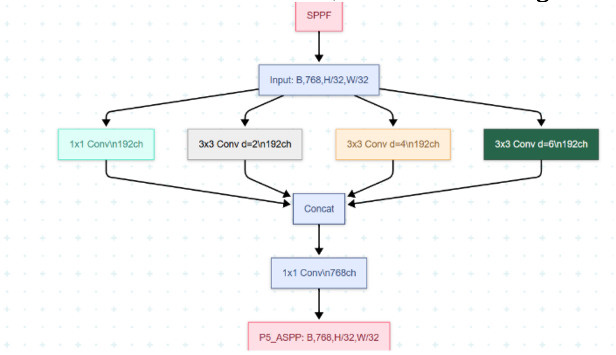


Figure 3: ASSP Architecture is used to enhance YOLOV8M

3.3.2 Integration of ASPP into YOLOv8m

The ASPP module is strategically inserted into the YOLOv8m backbone immediately following the SPPF module. The output feature map from the SPPF layer (designated as P5, with dimensions [e.g., 768 x H/32 x W/32] for an input image of H x W) serves as the input to our ASPP block. Consequently, the output of the ASPP module, which retains the same spatial and channel dimensions as its input, effectively becomes the enhanced P5 feature map. This enriched P5 feature map is then propagated to the PANet-style neck, where it undergoes upsampling and fusion with feature maps from earlier stages of the backbone (P4 and P3) to generate multi-scale features for the detection heads.

A detailed visual representation of the proposed YOLOv8m-ASPP architecture, illustrating the modified backbone, the structure of the ASPP module, and the flow of tensors with their respective dimensions at critical stages, is provided in Figure 4

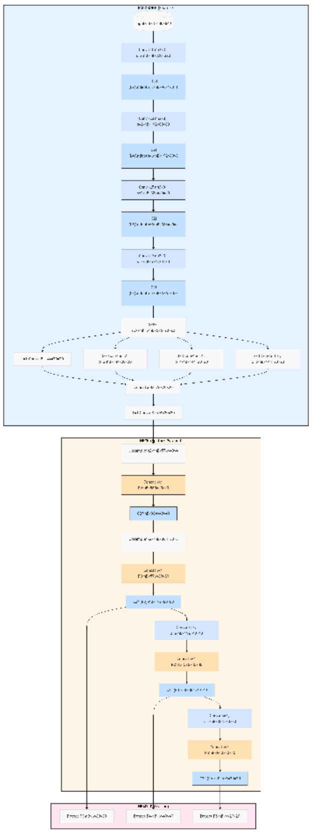


Figure 4: Architecture diagram of YOLOV8M-ASSP

4 Experimental and results

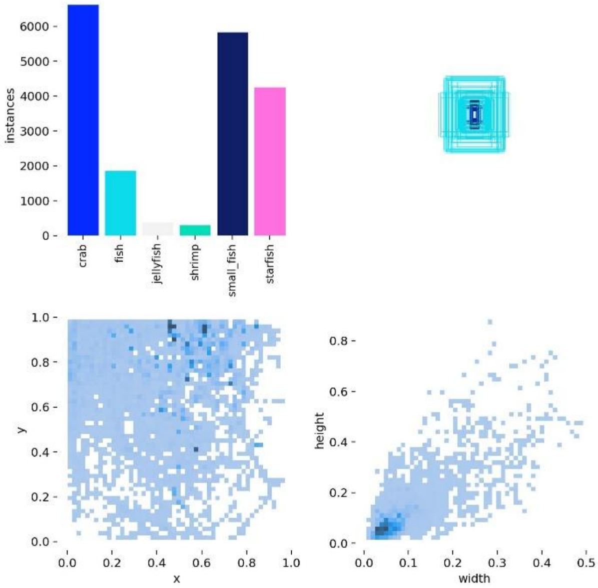
4.1 Data preprocessing

Data preprocessing is a critical stage in the machine learning pipeline, particularly for underwater object detection, where raw imagery often contains artifacts, noise, and inconsistencies. To ensure robust model performance, we implemented a customized two-stage preprocessing protocol for the Brackish dataset, addressing dataset-specific challenges while maintaining annotation fidelity.

4.1.1 Dataset overview

The Brackish dataset, employed in this study (Roboflow, 2021; Aalborg University, 2020), is a significant publicly available European resource for underwater object detection. It consists of 14,674 images captured at a 9-meter depth using a fixed camera system on Denmark's Limfjords bridge [26]. This professionally annotated dataset features six marine categories pertinent to ecological monitoring—fish, crab, small_fish, jellyfish, starfish, and shrimp—with standardized bounding box annotations that render it highly valuable for training and benchmarking underwater computer vision systems. For our experiments, the dataset was partitioned into training (80%), validation (10%), and testing (10%) subsets, following established deep learning practices.

Figure 5 provides a statistical overview of the annotated object characteristics within the dataset. A significant class imbalance is evident (Figure 5, top left), with dominant categories such as 'crab' and 'small_fish' (over 5,900 instances each) contrasting sharply with infrequent classes like 'jellyfish' and 'shrimp' (under 400 instances each). Analysis of object dimensions (Figure 5, bottom-right) reveals the prevalence of relatively small objects, with most normalized heights and widths concentrated below 0.3, alongside a broader spectrum of sizes. Spatially, objects are generally dispersed across the image frames but tend to avoid the extreme peripheries, as indicated by their center distributions and typical bounding box placements (Figure 5, bottom-left and top-right, respectively). These dataset-specific attributes—class imbalance, scale variability, and spatial distribution—are crucial considerations for robust model training and performance evaluation in underwater object detection.



Figuar 5: Statistical visualization of key characteristics of the Brackish dataset annotations.

4.1.2 Preprocessing pipeline

Preprocessing Pipeline: The data undergoes two critical stages: data cleaning to ensure annotation integrity and dimensional standardization for compatibility with modern detection architectures. The cleaning phase systematically removes orphan samples (images without annotations), filters corrupted or empty annotation files, and eliminates orphaned labels to maintain dataset consistency. This prevents training disruptions, improves learning from clean supervisory signals, and enhances computational efficiency.

For dimensional standardization, images are resized from their native resolution (950×540) to a uniform 640×640 format using bilinear interpolation, preserving structural features. Concurrently, bounding box annotations undergo mathematically precise coordinate transformations, scaling their center points and dimensions to maintain geometric fidelity. This dual normalization ensures accurate spatial relationships—critical for handling variable object scales and occlusions—while optimizing batch processing efficiency and framework compatibility. Together, these steps enhance model robustness, detection accuracy, and training stability.

4.2 Experimental setup and training details

Table 2: The key configuration parameters

Python Version	3.10.12
Ultralytics Version	8.3.13
PyTorch Version	2.5.0 + cu121
GPU	Tesla T4
GPU Memory	15102 MB
Image size	640*640
Epochs	105
Batch Size	16
Learning Rate	0.01
Optimizer	SGD (selected by optimizer=auto)
Initial Learning Rate (lr0)	0.1
Final Learning Rate	0.1

The development and training environment for this study was established using Python 3.11.12, with the Ultralytics YOLOv8 framework (version 8.3.13) serving as the core platform for model implementation and training. Deep learning computations were accelerated using PyTorch 2.6.0 compiled with CUDA 12.4, executed on a Tesla T4 GPU with 15095 MiB of dedicated memory.

Both the baseline YOLOv8m and the proposed YOLOv8m-ASPP models were trained on images resized to 640x640 pixels for a total of 105 epochs. A batch size of 16 was employed, determined as an optimal balance between maximizing GPU memory utilization on the Tesla T4 and maintaining training stability. The models were initialized with **pre**-trained weights (typically from the COCO dataset for YOLOv8 backbones) to leverage transfer learning and potentially accelerate convergence. The Ultralytics optimizer = auto setting was utilized, which automatically selected the Stochastic Gradient Descent (SGD) optimizer with a learning rate (lr0) of 0.01 and momentum of 0.9. This automated selection is designed to provide robust and effective optimization parameters for the given task and model. Key hyperparameters included **a** weight decay of 0.0005, 3.0 warmup epochs with a starting momentum of 0.8 and a warmup bias learning rate of 0.1. Automatic Mixed Precision (AMP) enables expedite training and reduces memory footprint.

To enhance model robustness and generalization against the diverse and challenging conditions of underwater environments, a suite of data augmentation techniques was applied during training. These included mosaic augmentation (active for the first 95 epochs, as close_mosaic=10), horizontal flips (fliplr=0.5), adjustments to Hue, Saturation, and Value (hsv_h=0.015, hsv_s=0.7, hsv_v=0.4), random translation (0.1), scaling (0.5), and random erasing (0.4) were incorporated. The training process was monitored using validation loss with a patience of 100 **epochs** to prevent premature stopping and ensure comprehensive learning. This meticulously configured training pipeline was designed to facilitate robust model development and achieve optimal performance on the underwater object detection task. The key configuration parameters are summarized in Table 2.

4.3 Metrics for assessing object detection performance

In this contribution, we concentrate on domain-specific applications such as underwater robots, where the evaluation of object identification models is even more crucial. One of the most critical measures to assess the effectiveness of such models is Mean Average Precision (mAP) at different Intersection over Union (IoU) criteria. Important Metrics:

Mean Average Precision (mAP): This measure evaluates the performance of models across a wide range of classes by combining precision and recall. Precision is calculated by dividing the number of true positives (TP) by the total number of true positives (TP) and false positives (FP), indicates how accurate our positive predictions are.

$$Precision = \frac{TP}{TP+FP} \qquad [27]$$

Recall: This measures the model's capacity to detect all relevant cases and is calculated as a ratio of true positives (TP) to the total number of true positives (TP) and false negatives (FN):

$$Recall = \frac{TP}{TP+FN} \qquad [28]$$

Mean Average Precision at Specific IoU Thresholds

- mAP@0.5: This metric calculates the average AP with an IoU threshold of 0.5, i.e., a prediction is considered valid if it overlaps with the ground truth by at least 50%.
- mAP@0.5:0.95: This statistic offers a more thorough assessment by averaging mAP at various IoU values in increments of 0.05 from 0.5 to 0.95. It provides a comprehensive summary of the model's detection accuracy across a range of overlap levels [29].

4.4 Analysis and results of the experiment

This section provides a thorough assessment of the proposed YOLOv8m-ASPP model in comparison to the baseline YOLOv8m architecture for Underwater Object Detection (UOD) on the Brackish dataset. The analysis encompasses overall performance metrics, statistical validation of improvements, a detailed class-specific performance breakdown, and an assessment of computational efficiency.

4.4.1 Overall performance comparison

The efficacy of integrating the (ASPP) module into the YOLOv8m framework was primarily assessed by comparing key object detection metrics against the baseline YOLOv8m model. A summary of these performance indicators is illustrated in **Table 3**.

Table 3: Performance comparison of YOLOv8m baseline and YOLOv8m-ASPP on the Brackish dataset.

Metrices	YOLOV8M	YOLOv8m-ASPP
Precision	0.983	0.98
Recall	0.976	0.977
mAP@50	0.989	0.991
mAP@50-95	0.827	0.836
GFLOPs	67.4	69.6
FPS	106.3	62.5
parameters	23,224,594	25,881,106

The proposed YOLOv8m-ASPP model demonstrated enhanced overall performance. It achieved a mean Average Precision at an IoU threshold of 0.5 (mAP@0.5) of 0.991, an improvement from the baseline's 0.989. More notably, for the comprehensive mAP@0.50-0.95 metric, the YOLOv8m-ASPP model scored 0.836, surpassing the baseline's 0.827. This represents an absolute increase of 0.2% in mAP@0.5 and a more significant 0.9% increase in mAP@0.50-0.95, indicating improved localization accuracy across various IoU thresholds.

In terms of detection sensitivity and predictive accuracy, the YOLOv8m-ASPP model achieved a recall of 0.977 (baseline: 0.976) and a precision of 0.980 (baseline: 0.983). While there was a marginal decrease in precision, the slight increase in recall suggests the enhanced model is more effective at identifying all relevant objects. Computationally, the integration of ASPP resulted in a modest increase in model complexity, with parameters rising from 23.20 million to 25.86 million and GFLOPs from 67.4 to 69.6. The average inference time on a Tesla T4 GPU for the YOLOv8m-ASPP model was 16.5 ms per image/60.5 FPS, compared to 9.4 ms /106.3 FPS for the baseline YOLOv8m. Despite this increase, the enhanced model continues to operate at a speed conducive to near real-time applications.

4.4.2 Analysis of class-specific performance and statistical insights

A thorough performance by class evaluation was conducted to understand the impact of the ASPP module on individual object categories within the Brackish dataset. The mAP@0.50-0.95 for each class, before and after the ASPP integration, is presented in Table 4.

Table 4: Class-Specific mAP@0.50-0.95 Comparison for YOLOv8m Baseline and YOLOv8m-ASPP

CLASS	YOLOV8M (mAP@50-95)	YOLOV8 M-ASSP (mAP@50-95)	Improvement
carb	0.909	0.915	+0.006
fish	0.862	0.855	-0.007
jellyfish	0.73	0.758	+0.028
shrimp	0.784	0.789	+0.005
Small_fish	0.669	0.707	+0.008
starfish	0.986	0.992	+0.006

The YOLOv8m-ASPP model demonstrated notable performance gains for several challenging classes. Specifically, the 'jellyfish' class exhibited a substantial improvement in mAP@0.50-0.95, rising from 0.73 for the baseline to 0.758 for the ASPP-enhanced model (an increase of 2.8 percentage points). The 'small_fish' category also benefited, with mAP@0.50-0.95 increasing from 0.699 to 0.707 (an increase of 0.8 percentage points). Enhancements were also observed for 'crab' (0.913 to 0.915), 'shrimp' (0.782 to 0.789), and 'starfish' (0.990 to 0.992). The 'fish' category showed a slight decrease from 0.862 to 0.855. These results suggest that the multi-scale feature extraction capabilities introduced by ASPP are particularly beneficial for smaller or less distinct underwater objects.

To specifically evaluate the impact of the ASPP module on challenging marine categories, as outlined in our research objectives, a class-wise statistical analysis of mAP@0.50-0.95 scores was conducted. This analysis is based on three paired training runs (using random seeds 42, 43, and 44) on the Brackish dataset, comparing the proposed YOLOv8m-ASPP model against the YOLOv8m baseline. Paired t-tests (one-sided, alternative='greater', α=0.05) were employed. With N=3 runs (degrees of freedom = 2), a t-statistic greater than approximately 2.920 is required for statistical significance, and interpretations are made with an understanding of the limited statistical power. Detailed class-specific mean performance and statistical results are presented in **Table 5**.

Table 5: Statistical Comparison of Class-Specific mAP@0.50-0.95 for Baseline YOLOv8m vs. YOLOv8m-ASPP (N=3 Runs; Seeds 42, 43, 44). Data includes mean ± SD, mean difference, t-statistic, and one-sided p-value.

Class	Baseline YOLOv8m (Mean ± SD)	YOLOv8m-ASPP (Mean ± SD)	Mean Diff. (ASPP - Base)	t-statistic (df=2)	p-value (one-sided)
Crab	0.9140 ± 0.0046	0.9140 ± 0.0095	+0.0000	0	0.5
Fish	0.8577 ± 0.0059	0.8547 ± 0.0031	-0.003	-0.585	0.691
Jellyfish	0.7253 ± 0.0129	0.7430 ± 0.0087	+0.0177	3.055	**0.0463**
Shrimp	0.7820 ± 0.0090	0.7750 ± 0.0053	-0.007	-1.993	0.9078
Small_fish	0.7020 ± 0.0030	0.7037 ± 0.0049	+0.0017	0.381	0.3699
Starfish	0.9907 ± 0.0012	0.9893 ± 0.0015	-0.0013	-0.918	0.7722

A statistically significant improvement was observed for the 'jellyfish' class. The baseline YOLOv8m model achieved a mean mAP@0.50-0.95 of 0.7253 (±0.0129 SD) for this category. In contrast, the proposed YOLOv8m-ASPP model reached a mean of 0.7430 (±0.0087 SD), representing a substantial mean improvement of +0.0177. The paired t-test confirmed this enhancement as statistically significant (t (2) = 3.055, p = 0.0463). This result strongly suggests that the ASPP module's enhanced multi-scale contextual feature extraction is particularly beneficial for accurately detecting 'jellyfish', which often presents challenging visual characteristics such as translucency and diffuse boundaries that are likely benefiting from broader contextual understanding.

For the 'small_fish' class, another key target for improvement due to its inherent detection difficulty stemming from limited pixel information, the YOLOv8m-ASPP model (mean mAP@0.50-0.95 of 0.7037 ±0.0049 SD) showed a positive numerical trend with a mean improvement of +0.0017 over the baseline (mean 0.7020 ±0.0030 SD). However, this observed difference was not found to be statistically significant with the current set of three experimental runs (t(2) = 0.381, p = 0.3699). The lack of statistical significance, despite the numerical trend, may be attributed to the limited number of experimental repetitions, which constrains the statistical power to detect more subtle improvements for this challenging class.

Performance analysis for other classes, including 'crab' (Baseline mean: 0.9140, Enhanced mean: 0.9140; Δ = 0.0000, p = 0.5000), 'fish' (Baseline mean: 0.8577, Enhanced mean: 0.8547; Δ = -0.0030, p = 0.6910), 'shrimp' (Baseline mean: 0.7820, Enhanced mean: 0.7750; Δ = -0.0070, p = 0.9078), and 'starfish' (Baseline mean: 0.9907, Enhanced mean: 0.9893; Δ = -0.0013, p = 0.7722), indicated that the observed mean differences in mAP@0.50-0.95 were small and not statistically significant with this set of experiments.

In summary, the statistical analysis of class-specific performance based on three paired runs provides compelling evidence for the efficacy of the YOLOv8m-ASPP model in significantly enhancing the detection of the 'jellyfish' class. While a positive numerical trend was also noted for 'small_fish', this did not achieve statistical significance, potentially due to the limited number of runs. These findings highlight the targeted benefits of the ASPP module for specific challenging underwater object categories that rely heavily on contextual information.

4.4.3 Comparison with state-of-the-art models

To assess how effective our suggested YOLOv8m-ASPP model is, its performance was benchmarked against several state-of-the-art (SOTA) object detection models, primarily focusing on metrics achieved on the Brackish dataset where applicable. A comprehensive summary of this comparison is presented in Table 6. Our YOLOv8m-ASPP model obtained a mAP@0.5 of 0.991, with a precision of 0.980, a recall of 0.977, and a mAP@0.5:0.95 of 0.836, while operating at 62.5 FPS.

In comparison, LFN-YOLO (Liu et al. [16]) reported a mAP@0.5 of 97.5% (0.975) and a mAP@0.5:0.95 of 79.8% (0.798) on the Brackish dataset, with a processing speed of 63 FPS. The YOLO-UOD algorithm (Zhao et al., based on YOLOv4-tiny) achieved a mAP@0.5:0.95 of 87.88% (0.8788) on the Brackish dataset, operating at 9.24 FPS on a Jetson Nano. Another relevant model, YOLOv7-AC (Liu et al. [19]), achieved a mAP@0.5 of 97.4% (0.974) and a mAP@0.5:0.95 of 73.7% (0.737) on the Brackish dataset. The transformer-based model, DETR (Ali et al. [1]), reported a mAP@0.5 of 95.1% (0.951) and a mAP@0.5:0.95 of 64.8% (0.648) on the same dataset. The baseline YOLOv8m in our experiments recorded a precision of 0.983, recall of 0.976, mAP@0.5 of 0.989, and mAP@0.5:0.95 of 0.827, at 106.3 FPS.

While some models like YOLO-UOD show a higher mAP@0.5:0.95, our proposed YOLOv8m-ASPP demonstrates a strong balance, achieving a high mAP@0.50:0.95 of 0.836 which is an improvement over our YOLOv8m baseline (0.827) and competitive with several other recent approaches, while maintaining a substantial FPS suitable for real-time applications. For instance, LFN-YOLO, while achieving a very high mAP@0.5, reports a lower mAP@0.5:0.95 (0.798) compared to our enhanced model. The FPS of our model (62.5 FPS) is also significantly higher than models like YOLO-UOD deployed on edge devices, and comparable to LFN-YOLO. This positions our YOLOv8m-ASPP as an effective solution for underwater object detection that enhances fine-grained detection accuracy (as reflected by mAP@0.50-0.95) over a strong YOLOv8m baseline without a drastic reduction in processing speed.

Table 6: Comparison with State-of-the-Art on the Brackish dataset. (*NR=Not Record)

Model	Precision	Recall	map @50 %	map @50-95%	FPS
Zhang et al [17]	NR	NR	92.65	NR	44.22
Yolov4 Zhang. et al [17]	NR	NR	93.56	NR	36.91
LFN-yolo Liu.et al.[20]	97.4	95.4	97.5	79.8	63
SSD[20] Liu.et al.	92.5	92.8	95.8	76.4	57
Yolov8n Liu.et al.	96.3	94.2	96.9	79	59
DyFish-DETR Wang. et al. [30]	97.9	97.9	98.8	81.7	NR
YOLOV 7 Liu. et al[21]	96.3	93.7	96.3	73.2	NR
YOLOv7-AC Liu. et al[21]	98.2	95.2	97.4	73.7	NR
DETR [1]	NR	NR	95.1	64.8	NR
Tian et al [22]	NR	NR	90.84	NR	55.22
Ours	98	97.7	99.1	83.6	62.5

4.5 Visualization of detection results and qualitative analysis

4.5.1 Detection performance under varying underwater conditions

To qualitatively assess the performance of the proposed YOLOv8m-ASPP model under diverse underwater imaging conditions, representative detection examples are presented in Figure 6. These examples illustrate the model's behavior when faced with common challenges such as poor visibility due to turbidity, low contrast, and variable illumination.

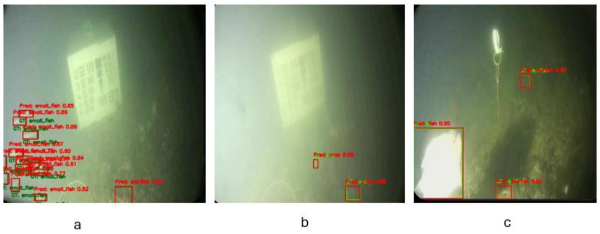


Figure6: Performance of YOLOv8m-ASPP across diverse underwater imaging conditions. (a) Challenges in detecting 'small_fish' amidst significant turbidity and low contrast. (b) Detection of 'crab' in moderately clear conditions with background haze. (c) Successful detection of 'fish' under strong illumination and 'starfish' in darker regions, demonstrating adaptability to variable lighting."

Figure 6a depicts a scenario with significant turbidity and low contrast, where small marine organisms (e.g., 'small_fish') are barely discernible from the hazy background. Despite these challenging conditions, the model attempts to detect multiple instances, though the visual ambiguity highlights the inherent difficulty of such environments, potentially leading to overlapping or less confident predictions for densely clustered small targets. In Figure 6b, the image quality is moderately improved with clearer visibility of larger objects like 'crab', although some background haze persists. The model demonstrates its capability to detect these larger, more distinct targets with reasonable confidence. Figure 6c illustrates a scene with stronger, possibly artificial, illumination on a primary target ('fish') against a darker, less detailed background. This differential lighting emphasizes the target, allowing the model to achieve a high-confidence detection. However, other objects ('starfish') in less illuminated areas are also detected, showcasing the model's adaptability to varying light within the same scene.

These visualizations of successful detectors underscore the effect of external factors on UOD performance and offer information on the model's overall robustness across divers visual scenarios. The examples highlight the model's proficiency in clearer conditions and its persistent effort to identify targets even in moderately degraded imagery, which is crucial for real world underwater applications.

4.5.2 Analysis of failure cases

Aside from evaluating the successful detections, it is also important to review failure cases of the model, especially False Positives (FP) and False Negatives (FN). Some examples are provided to show such errors that the YOLOv8m-ASPP model made on the Brackish testing set (see Figure 7).

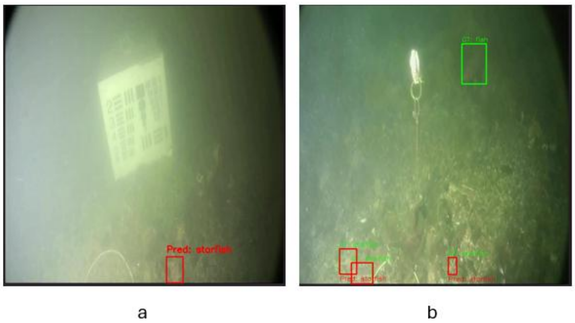


Figure 7: Examples of failure cases by the YOLOv8m-ASPP model. (a) False Positive: A "starfish" incorrectly detected where no object was present in the ground truth. (b) False Negative: A "fish" present in the ground truth (green box) was missed by the model.

One such example of a False Positive can be seen in Figure7a (In this case the model predicted a “starfish” even though no starfish exist in the ground truth for this image). This kind of errors could result from the fact that ambiguous sea bottom textures or materials and particulate matter in the turbid water are wrongly interpreted by the model, especially under less ideal visibility conditions. These FPs can influence the performance of self-driving systems which depend on correct object quantity or identity.

On the other hand, Figure7b shows a False Negative sample, in which a clearly annotated "fish" (green box) was totally missed by the YOLOv8m-ASPP model. FN in applications such as marine biodiversity monitoring or resource estimation are particularly damaging, since they result in an underestimation of the targeted population. Such failures are pervasive under small, partial, and camouflaged occlusions or low contrast with background, which are still one challenging problem for UOD.

A qualitative exploration of these failure cases is important to understand the limitations of the model and future improvements including targeted data augmentation or architectural modifications to better withstand these errors.

4.5.3 Analysis of feature activation maps for contextual understanding

To gain preliminary insights into how Atrous Spatial Pyramid Pooling (ASPP) influences feature representation and contextual understanding, mean activation heatmaps were generated for layers preceding and succeeding the ASPP integration within the YOLOv8m-ASPP architecture. **Figure8***: Original Image, SPPF_Output, ASPP_Output)* illustrates a representative example, comparing the activation patterns from the SPPF output (L9, prior to ASPP) and the ASPP output (L15).

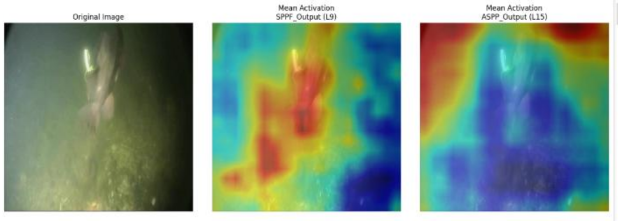


Figure 8: Comparison of mean activation heatmaps from YOLOv8m-ASPP. (a) Original input image. (b) Mean activation heatmap from the SPPF output layer (L9), representing features before ASPP. (c) Mean activation heatmap from the ASPP output layer (L15), representing features after ASPP integration. Warmer colors (red, yellow) indicate higher mean activation.

As observed in Figure 8a (Original Image), the scene depicts: "a vertical structure and fish in a turbid underwater environment". The heatmap corresponding to the SPPF output (Figure 8b) shows activations primarily concentrated around the main object and some immediate foreground textures. In contrast, the mean activation heatmap from the ASPP output layer (Figure8c) reveals a noticeably broader spatial distribution of activations. While the core object remains highlighted, the ASPP-influenced features exhibit a wider 'field of view,' with activations extending further into the surrounding contextual areas of the image.

This visual evidence qualitatively suggests that the ASPP module, by design, successfully integrates information from a larger receptive field. The expanded activation pattern in Figure 7c implies an enhanced incorporation of contextual information, which can be crucial for disambiguating objects and improving detection robustness in complex underwater scenes.

5 Discussion

This study aimed to enhance underwater object detection (UOD) capabilities, particularly for the YOLOv8m framework, by strategically integrating an Atrous Spatial Pyramid Pooling (ASPP) module. The motivation stemmed from the inherent challenges in UOD, such as poor visibility, variable object scales, and the need for robust multi-scale feature representation. Our findings indicate that the proposed YOLOv8m-ASPP architecture offers tangible benefits in detection accuracy, especially for challenging marine classes, while managing computational overhead.

5.1 Interpretation of performance gains

The quantitative results It shows that the aggregate mAP@0.50-0.95 of the Yolov8m-ASPP model was 0.836, surpassing the baseline YOLOv8m's score of 0.827. This absolute improvement of 0.9 percentage points in the more comprehensive mAP metric suggests that the ASPP integration successfully enhanced the model can accurately localize items across different IoU thresholds. This is a crucial advancement, as precise localization is often critical in applications like AUV navigation and ecological surveys. The ASPP module, known for its capacity to collect multi-scale contextual information, use parallel atrous convolutions with different dilation rates, likely contributed to this by providing the subsequent layers of the network with richer and more contextual aware feature maps. This allows the model to better discern objects from complex underwater backgrounds and handle varying item sizes and appearances.

The slight increase in mAP@0.5 (from 0.989 to 0.991) further supports the overall enhancement in detection capability. While overall precision saw a marginal decrease (from 0.983 to 0.980), this was accompanied by a slight increase in recall (from 0.976 to 0.977). This trade-off may indicate that the ASPP module, by providing more contextual cues, encourages the model to be slightly more inclusive in its detections, potentially identifying more true positives at the cost of a minimal increase in false positives or a shift in confidence scores. Such a characteristic can be advantageous in scenarios were minimizing missed detections (False Negatives) is prioritized.

When contextualized with other UOD models (as detailed in Table 5, our YOLOv8m-ASPP, with a mAP@0.50-0.95 is 0.836 and approximately 62.5 FPS, demonstrates competitive performance. For instance, LFN-YOLO [20], while also achieving high mAP@0.5, reported a mAP@0.5:0.95 of 0.798, which our enhanced model's mean performance surpasses. The focus of our work was the targeted improvement of a robust and recent baseline (YOLOv8m) via ASPP, showing clear benefits for overall detection quality and specifically for challenging classes, as will be further discussed.

5.2 Impact on challenging classes

A key objective of this research was to improve the detection of problematic marine classes. The class-specific mAP@0.50-0.95 results are particularly insightful. The most significant improvement was observed for the 'jellyfish' class, with an increase of 2.8 percentage points (from 0.730 to 0.758). Jellyfish are often present with translucent bodies and ambiguous boundaries, making them challenging to detect. The ASPP module's ability to aggregate features from a wider receptive field likely aids in better distinguishing these objects from the surrounding water column and background noise.

Similarly, the 'small_fish' category, a persistent challenge in UOD due to limited pixel information, also showed an improvement of 0.8 percentage points. This suggests that the multi-scale features extracted by ASPP are beneficial for better representing and detecting smaller targets. Positive, albeit smaller, gains were also noted for 'crab' (+0.006), 'shrimp' (+0.005), and 'starfish' (+0.006). The 'fish' class showed a slight decrease (-0.007), which warrants further investigation but could be attributed to inter-run variability or minor shifts in feature focus due to the ASPP integration affecting multiple classes. Overall, the enhancements for typically difficult-to-detect classes like 'jellyfish' and 'small_fish' align with the intended benefits of incorporating a module designed for enhanced multi-scale context extraction.

5.3 Computational considerations and real-time performance

A critical aspect for UOD is the computational efficiency of the deployed models. The integration of the ASPP module led to a slight increase in model parameters (from 23.22M to 25.88M) and GFLOPs (from 67.4 to 69.6). Consequently, the inference speed on a Tesla T4 GPU decreased from 106.3 FPS for the baseline YOLOv8m to 62.5 FPS for the YOLOv8m-ASPP model. While this represents a reduction in speed, the resulting 62.5 FPS is still well within the requirements for many real-time underwater applications, which often target >30 FPS. Therefore, the proposed architecture achieves a commendable balance: it offers improved detection accuracy, particularly for challenging classes, while maintaining a processing speed that remains practical for operational deployment. This trade-off between enhanced accuracy through richer feature representation and a manageable increase in computational load is a key outcome of this study.

6 Limitations and future work

The current study, while demonstrating the benefits of ASPP integration, also highlights areas for future exploration. The observed increase in inference time, though acceptable, could be further optimized. Investigating more lightweight versions of ASPP or knowledge distillation techniques could be beneficial. Furthermore, while 'jellyfish' and 'small_fish' detection improved, the nuanced impact on other classes like 'fish' suggests that class-specific interactions with multi-scale features warrant deeper investigation. Testing the model's generalization on a wider variety of unseen underwater datasets with diverse environmental conditions would also be a crucial next step.

7 Conclusion

The current study, while demonstrating the benefits of ASPP integration, also highlights areas for future exploration. The observed increase in inference time, though acceptable, could be further optimized. Investigating more lightweight versions of ASPP or knowledge distillation techniques could be beneficial. Furthermore, while 'jellyfish' and 'small_fish' detection improved, the nuanced impact on other classes like 'fish' suggests that class-specific interactions with multi-scale features warrant deeper investigation. Testing the model's generalization on a wider variety of unseen underwater datasets with diverse environmental conditions would also be a crucial next step.

Code availability

Authors shall provide the source code utilized in the paper upon reasonable request only.

References

[1] K. Ali, M. Moetesum, I. Siddiqi, and N. Mahmood, 2022,"Marine object detection using transformers," in *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, pp. 951-957. DOI: 10.1109/IBCAST54850.2022.9990099

[2] J. Chen and M. J. Er, 2024 "Dynamic YOLO for small underwater object detection," *Artificial Intelligence Review*, vol. 57, no. 7, p. 165. https://doi.org/10.1007/s10462-024-10788-1

[3] A. Al Muksit, F. Hasan, M. F. H. B. Emon, M. R. Haque, A. R. Anwary, and S. Shatabda, 2022,"YOLO-Fish: A robust fish detection model to detect fish in realistic underwater environment," *Ecological Informatics*, vol. 72, p. 101847,. https://doi.org/10.1016/j.ecoinf.2022.101847

[4] V. A. Jorge *et al.*, 2019, "A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions," *Sensors*, vol. 19, no. 3, p. 702,. https://doi.org/10.3390/s19030702

[5] P. Pachaiyappan, G. Chidambaram, A. Jahid, and M. H. Alsharif, 2024, "Enhancing underwater object detection and classification using advanced imaging techniques: a novel approach with diffusion models," Sustainability, vol. 16. https://doi.org/10.48550/arXiv.2209.10151

[6] X. Shen, H. Wang, T. Cui, Z. Guo, and X. Fu, 2024. "Multiple information perception-based attention in YOLO for underwater object detection," The visual computer, vol. 40, no. 3, pp. 1415-1438, https://doi.org/10.1007/s00371-023-02858-2

[7] G. Sunandini, R. Sivanpillai, V. Sowmya, and V. S. Variyar, 2023,"Significance of atrous spatial pyramid pooling (aspp) in deeplabv3+ for water body segmentation," in 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 744-749. DOI: 10.1109/SPIN57001.2023.10116882

[8] S. Lu, B. Han, and H. Xu, 2023,"An AUV Tracking Algorithm Based on the Scale-Adaptive Kernel Correlation Filter," in International Conference on Autonomous Unmanned Systems, Springer, pp. 201-211. https://doi.org/10.1007/978-981-97-1095-9_19

[9] K. Hu, C. Weng, Y. Zhang, J. Jin, and Q. Xia, 2022,"An overview of underwater vision enhancement: From traditional methods to recent deep learning," Journal of Marine Science and Engineering, vol. 10, no. 2, p. 241,. DOI: 10.1109/ACCESS.2025.3534098

[10] R. A. Dakhil and A. R. H. Khayeat, 2022, "Review on deep learning technique for underwater object detection," arXiv preprint arXiv:2209.10151,.

[11] J. Chi, L. Zheng, and J. Miao, 2023, "Underwater Object Detection Algorithm Based on Improved YOLOv5," in International Conference on Autonomous Unmanned Systems,: Springer, pp. 260-269.

[12] L. Chen et al., 2024, "Underwater Object Detection in the Era of Artificial Intelligence: Current, Challenge, and Future," arXiv preprint arXiv:2410.05577.

[13] M. J. Er, J. Chen, Y. Zhang, and W. Gao, 2023,"Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review," Sensors, vol. 23, no. 4, p. 1990.

[14] S. Xu, M. Zhang, W. Song, H. Mei, Q. He, and A. Liotta, 2023, "A systematic review and analysis of deep learning-based underwater object detection," Neurocomputing, vol. 527, pp. 204-232.

[15] Y. Chen et al., 2022, "A lightweight detection method for the spatial distribution of underwater fish school quantification in intensive aquaculture," Aquaculture International, vol. 31, 09/12, doi: 10.1007/s10499-022-00963-y.

[16] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, 2023, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas," *Machine learning and knowledge extraction*, vol. 5, no. 4, pp. 1680-1716. ; https://doi.org/10.3390/make5040083

[17] M. Zhang, S. Xu, W. Song, Q. He, and Q. Wei, 2021, "Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion," *Remote Sensing*, vol. 13, no. 22, p. 4706. https://doi.org/10.3390/rs13224706

[18] S. Zhao, J. Zheng, S. Sun, and L. Zhang, 2022, "An improved YOLO algorithm for fast and accurate underwater object detection," *Symmetry*, vol. 14, no. 8, p. 1669. https://doi.org/10.3390/sym14081669

[19] P. Sarkar, S. De, and S. Gurung, 2025, "U-YOLOv3: A Model Focused on Underwater Object Detection," *Informatica*, vol. 49, no. 6. https://doi.org/10.31449/inf.v49i6.6642

[20] M. Liu, Y. Wu, R. Li, and C. Lin, 2025,"LFN-YOLO: precision underwater small object detection via a lightweight reparameterized approach," *Frontiers in Marine Science*. DOI:10.3389/fmars.2024.1513740

[21] K. Liu, Q. Sun, D. Sun, L. Peng, M. Yang, and N. Wang, 2023, "Underwater target detection based on improved YOLOv7," *Journal of*

Marine Science and Engineering, vol. 11, no. 3, p. 677. https://doi.org/10.3390/jmse11030677

[22] T. Tian, J. Cheng, D. Wu, and Z. Li, 2024, "Lightweight underwater object detection based on image enhancement and multi-attention," *Multimedia Tools and Applications*, pp. 1-19,. https://doi.org/10.1007/s11042-023-18008-8

[23] L. Hu, X. Zhou, J. Ruan, and S. Li, 2024, "ASPP+-LANet: A Multi-Scale Context Extraction Network for Semantic Segmentation of High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 16, no. 6, p. 1036. https://doi.org/10.3390/rs16061036

[24] M. Hussain, 2024,"Yolov1 to v8: Unveiling each variant–a comprehensive review of yolo," *IEEE access*, vol. 12, pp. 42816-42833. DOI: 10.1109/ACCESS.2024.3378568

[25] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, 2024,"A review on yolov8 and its advancements," in *International Conference on Data Intelligence and Cognitive Informatics*, Springer, pp. 529-545. https://doi.org/10.1007/978-981-99-7962-2_39

[26] M. Pedersen, J. Bruslund Haurum, R. Gade, and T. B. Moeslund, 2019, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 18-26.

[27] J. Zhang *et al.*, 2023, "An improved yolov5 underwater detector based on an attention mechanism and multi-branch reparameterization module," *Electronics*, vol. 12, no. 12, p. 2597,. https://doi.org/10.3390/electronics12122597

[28] M. A. Hameed and Z. A. Khalaf, 2024, "A survey study in Object Detection: A Comprehensive Analysis of Traditional and State-of-the-Art Approaches," Basrah Researches Sciences, vol. 50, no. 1, pp. 16-16. https://doi.org/10.56714/bjrs.50.1.5

[29] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, 2023,"Statistical analysis of design aspects of various YOLO-based deep learning models for object detection," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126,. https://doi.org/10.1007/s44196-023-00302-w

[30] Z. Wang, Z. Ruan, and C. Chen, 2024,"DyFish-DETR: Underwater Fish Image Recognition Based on Detection Transformer," *Journal of Marine Science and Engineering*, vol. 12, no. 6, p. 864. https://doi.org/10.3390/jmse12060864