frontiers | Frontiers in Marine Science

TYPE Original Research
PUBLISHED 10 September 2025
DOI 10.3389/fmars.2025.1658205

OPEN ACCESS

EDITED BY
Katelyn Lawson,
Auburn University, United States

REVIEWED BY
Ana Carolina Luz,
Instituto de Estudos do Mar Almirante Paulo Moreira, Brazil
Ranjith Kumar Dinakaran,
Teesside University, United Kingdom

*CORRESPONDENCE
S. Pavithra
pavithra.sekar@vit.ac.in

RECEIVED 02 July 2025
ACCEPTED 21 August 2025
PUBLISHED 10 September 2025

CITATION
Jyothimurugan M, Pavithra S and Deepika Roselind J (2025) Efficient underwater ecological monitoring with embedded AI: detecting Crown-of-Thorns Starfish via DCGAN and YOLOv6.
Front. Mar. Sci. 12:1658205.
doi: 10.3389/fmars.2025.1658205

Efficient underwater ecological monitoring with embedded AI: detecting Crown-of-Thorns Starfish via DCGAN and YOLOv6

Mohan Jyothimurugan, S. Pavithra* and J. Deepika Roselind

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India

Introduction: Coral reefs are among the most vital and diverse ecosystems on the planet, providing habitats for marine life, supporting fisheries, and protecting coastlines. However, they are increasingly threatened by outbreaks of Crown-of-Thorns Starfish (COTS), a coral-eating predator capable of causing large-scale reef destruction. Traditional monitoring methods rely on manual diver surveys, which are time-consuming, labour-intensive, and unsuitable for rapid large-scale assessments.

Methods: To address these limitations, this study proposes an AI-powered framework for detecting COTS in underwater imagery. The system integrates advanced deep learning object detection techniques with synthetic data augmentation to improve model robustness and adaptability under complex underwater conditions. Synthetic training images were generated to expand dataset variability, while optimized detection models were designed for high accuracy and real-time inference.

Results: The final detection model demonstrated strong performance, achieving a precision of 0.927, recall of 0.903, and mAP@50 of 0.938. These results indicate the effectiveness of the framework in accurately identifying COTS across diverse underwater environments.

Discussion and Conclusion: The proposed solution is designed for deployment on embedded systems, ensuring practical, scalable, and efficient monitoring of coral reef ecosystems. By enabling real-time and high-accuracy detection of COTS, this framework supports timely interventions and contributes to the conservation and ecological resilience of coral reefs, particularly in vulnerable regions such as the Great Barrier Reef.

KEYWORDS

Crown-of-Thorns Starfish (COTS), underwater object detection, YOLOv6, faster R-CNN, generative adversarial network, embedded AI, coral reef monitoring

1 Introduction

Coral reefs, often referred to as the "rainforests of the sea," are among the most diverse and valuable ecosystems on Earth. They support a vast array of marine life, provide coastal protection, and contribute significantly to the economy through tourism and fisheries. Despite their ecological and economic importance, coral reefs face increasing threats from climate change, pollution, and biological disturbances. One of the most destructive biological threats is the Crown-of-Thorns Starfish (COTS) (Acanthaster planci), a coral-eating echinoderm that can rapidly degrade reef structures by consuming living coral polyps. COTS outbreaks have been recorded across the Indo-Pacific region, with the Great Barrier Reef (GBR) in Australia being one of the most severely affected areas. Each starfish can consume large areas of coral cover, and when populations surge uncontrollably, the resulting damage can outpace coral regeneration. The complexity and vastness of reef ecosystems make manual COTS monitoring challenging, time-consuming, and often inaccurate. Traditional methods involving human divers to visually identify and count COTS are not scalable, provide limited coverage, and are prone to fatigue and bias. These limitations highlight the need for automated, intelligent, and scalable solutions capable of accurately detecting and localizing COTS in real time under diverse underwater conditions.

Existing approaches to underwater object detection, particularly for identifying Crown-of-Thorns Starfish (COTS), face several significant limitations that hinder their applicability in realworld scenarios. One of the primary challenges is environmental complexity underwater imagery is often affected by low contrast, variable lighting conditions, turbidity, and occlusions caused by surrounding marine life such as corals and algae. These factors reduce visual clarity, making it difficult for conventional object detection algorithms to perform effectively. Data scarcity further compounds this challenge. Deep learning models require large volumes of well-labelled data for effective training; however, publicly available annotated datasets for COTS are limited in size and lack diversity. This often leads to overfitting, causing models to perform poorly when applied to different underwater environments. Moreover, high false-positive rates and missed detections remain common in current methods, particularly when class imbalance is not adequately addressed.

Smaller or partially occluded starfish often go undetected, while false alarms increase due to background noise. Another major challenge is the incompatibility of many existing models with real time and embedded deployments. For example, although models like Faster R-CNN achieve high accuracy, their two-stage architecture incurs substantial computational costs, making them inefficient for embedded systems such as underwater drones or remotely operated vehicles (ROVs), where real-time responsiveness and energy efficiency are critical. Additionally, conventional augmentation techniques such as flipping, rotation, and brightness adjustments are insufficient to capture the full diversity and complexity of underwater environments. Consequently, models trained solely with these augmentations often perform poorly in previously unseen conditions, limiting their robustness and adaptability in field applications.

The primary objective of this research is to develop an intelligent, real time, and resource efficient object detection framework for accurately identifying Crown-of-Thorns Starfish (COTS) in complex underwater environments. To achieve this, the study pursues several specific goals. First, it builds a robust detection system by leveraging state-of-the-art deep learning algorithms such as Faster R-CNN and YOLOv6, both adapted to address the unique visual and contextual challenges of underwater imagery. Second, to mitigate the limitations of small and unrepresentative datasets, it employs Deep Convolutional Generative Adversarial Networks (DCGAN) to synthetically generate realistic underwater scenes containing COTS, thereby improving dataset diversity and model generalizability. Third, the models are optimized for deployment on embedded platforms, enabling low latency inference and efficient operation in practical scenarios such as underwater drones or remotely operated vehicles. Finally, model performance is evaluated using standard benchmarking metrics, including Precision, Recall, mAP@50, Inception Score, and Fréchet Inception Distance (FID), with particular emphasis on comparing the effectiveness of real versus GAN generated data in enhancing detection accuracy.

This study introduces a hybrid detection framework that integrates Faster R-CNN, DCGAN, and YOLOv6 to accurately detect Crown-of-Thorns Starfish (COTS) in underwater imagery, achieving a balance between high accuracy and real-time performance. The key innovations are:

- Hybrid Detection Framework: A novel combination of Faster R-CNN, DCGAN, and YOLOv6 tailored for accurate and efficient COTS detection in diverse underwater environments.
- GAN-Based Data Augmentation: Use of DCGAN to generate realistic synthetic underwater images incorporating varied conditions such as turbidity, lighting variations, and occlusions, providing richer diversity than traditional augmentation techniques.
- Enhanced Faster R-CNN Architecture: Integration of Res2Net101 backbone with Focal Loss, Triplet Loss, and Soft-NMS to improve detection precision, address class imbalance, and refine bounding boxes in complex marine scenes.
- Real-Time Deployment with YOLOv6: YOLOv6, trained on the augmented dataset, is optimized for speed and lightweight deployment on embedded systems such as underwater drones, achieving Precision: 0.927, Recall: 0.903, and mAP@50: 0.938.
- Domain-Specific Innovation: The fusion of synthetic data generation, advanced detection architectures, and embedded optimization represents a first-of-its-kind ecological AI solution for marine biodiversity conservation and rapid reef monitoring.

By combining deep generative modelling, advanced object detection architectures, and real-time optimization, this work delivers a significant advancement in automated marine biodiversity monitoring.

2 Literature survey

Coral reefs, especially those like the Great Barrier Reef, represent one of the most biologically diverse ecosystems on the planet. However, they are increasingly under threat from climate change, pollution, and biological disturbances, such as the outbreaks of Crown-of-Thorns Starfish (COTS), a coral-eating predator. These outbreaks are difficult to control without early detection, making continuous and large-scale monitoring essential for ecological conservation. Traditional manual survey methods are both labor-intensive and insufficient for large-scale timely assessments. Consequently, researchers have turned to deep learning and embedded artificial intelligence (AI) to provide automated, scalable, and real-time monitoring solutions in underwater environments. This literature review outlines the evolution and convergence of three key domains - underwater object detection, generative adversarial data augmentation, and real-time inference on embedded systems for the development of efficient AI-powered marine monitoring solutions.

Detecting objects in underwater environments introduces unique challenges not encountered in terrestrial domains. Turbidity, poor lighting, backscatter, color distortion, and partial occlusion significantly degrade image quality and object visibility. Standard object detection models must therefore be adapted to the underwater domain for enhanced robustness and reliability. Wang and Xiao (2023) proposed a highly relevant and domain-specific solution by enhancing the traditional Faster R-CNN architecture. Their improved model incorporated the Res2Net101 backbone for multi-scale feature extraction, coupled with Soft Non-Maximum Suppression (Soft-NMS), Online Hard Example Mining (OHEM) and Generalized Intersection over Union (GIoU) loss. The result was a 3.3% increase in mean Average Precision (mAP), demonstrating the adaptability of region-based detection frameworks in complex marine environments. This work is particularly pertinent as it focuses on small and partially occluded underwater object's conditions under which COTS detection must operate. Similarly, Nambiar and Mittal (2022) developed a GAN-based super-resolution model tailored for sonar image enhancement, which significantly improved feature visibility in murky environments.

Lokanath et al. (2017) also demonstrated the feasibility of Faster R-CNN for object classification and detection in constrained visual conditions. Although not exclusively tailored for underwater scenarios, the robustness of their approach under challenging backgrounds underscores the utility of this architecture as a baseline detector, especially when coupled with domain-specific enhancements. Ren et al.'s (2015) seminal work on Faster R-CNN forms the theoretical foundation for many current object detection models. The introduction of Region Proposal Networks (RPNs) for learning object proposals greatly accelerated and refined the detection pipeline. This architecture, while foundational, has been significantly improved over time to address underwater-specific constraints through advanced backbones and loss functions.

The domain of underwater object detection has gained considerable momentum with the integration of deep learning techniques, aiming to overcome environmental challenges such as light scattering, low contrast, and visual distortions. Recent studies have focused on combining detection frameworks with image enhancement strategies, notably using Generative Adversarial Networks (GANs), attention mechanisms, and transformer-based models. Liu et al. (2022) introduced an improved Deep Convolutional GAN (DCGAN) for synthetic image generation, aiding training where annotated underwater data is scarce. Similarly, Thomas et al. (2022) developed a GAN-based super-resolution model tailored for sonar image enhancement, which significantly improved feature visibility in murky environments. These image enhancement efforts serve as a preprocessing step that boosts object detection performance in degraded underwater conditions. Chen and Er, (2024) addressed the challenge of small object detection by proposing Dynamic YOLO, a variant that dynamically adjusts receptive fields based on object size and density. Their method showed superior performance in detecting small-scale marine organisms and submerged objects in cluttered scenes. Chen et al. (2024) further contributed to data-centric advancements with the WaterPairs dataset, which consists of paired raw and enhanced underwater images. This benchmark supports supervised training for simultaneous image enhancement and object detection.

Cherian et al. (2022) provided a comprehensive survey on underwater image enhancement using deep learning, highlighting key architectures like GANs, CNNs, and attention-guided models, while identifying open challenges such as generalization and real-time performance. Dai et al. (2023) introduced edge-guided representation learning, improving object boundary clarity under blur and low contrast conditions. Their method leverages edge features to guide the learning process, resulting in more precise detection outcomes. Dakhil and Khayeat (2022; 2023) offered both a review and a methodological framework on underwater object detection using deep learning. Their works emphasize the evolution of detection techniques from classical CNNs to more advanced architectures such as YOLO and Faster R-CNN, advocating for the fusion of enhancement and detection pipelines. Edge et al. (2020) presented a generative approach that integrates detection cues into image enhancement using GANs. This detection-driven enhancement method produced visually superior images that align with object detection requirements, improving downstream accuracy.

Fayaz et al. (2024) developed a joint image restoration and detection model optimized for Autonomous Underwater Vehicles (AUVs), which operates effectively in noisy and low-light underwater scenes. The model's multitask learning approach allows it to simultaneously clean degraded images and detect relevant objects in real-time, supporting AUV-based missions like coral reef inspection and search-and-rescue operations. Collectively, these studies highlight the growing synergy between image enhancement and object detection in underwater vision systems. GAN-based models play a pivotal role in data augmentation and preprocessing, while detection algorithms are evolving toward adaptive, context-aware and lightweight architectures suitable for embedded deployment. Future directions include creating more

diverse paired datasets, optimizing transformer based models for real-time use, and building explainable detection frameworks for trustworthy marine exploration.

Feng and Jin (2024) proposed CEH-YOLO, a composite-enhanced YOLO model featuring ESPPF and high-order deformable attention modules that improved detection accuracy under conditions of low contrast and blur. Gao et al. (2024) developed the PE-Transformer, a model incorporating path-enhanced attention mechanisms to better fuse contextual and spatial cues in underwater environments. Guo et al. (2024) contributed a real-time lightweight detection model by integrating FasterNet into YOLOv8, enabling fast inference and high accuracy on datasets like RUOD and URPC2022. In response to the need for robust datasets, Jian et al. (2024) conducted a detailed survey of underwater object detection datasets, identifying challenges such as annotation gaps and the need for domain-specific benchmarks.

Khriss et al. (2024) explored deep learning strategies specifically for plastic debris detection in marine environments, showcasing how tailored models can address unique underwater ecological problems. Lin et al. (2024) introduced a detection method that combines learnable query recall with lightweight adapter modules. Their model reduced computational cost while maintaining performance, making it suitable for real-time embedded systems. Similarly, Liu et al. (2024) presented a lightweight object detection algorithm optimized for embedded deployment, integrating higher order semantic information and image enhancement to improve robustness.

Liu et al. (2023) developed TC-YOLO, a model utilizing temporal context and attention mechanisms to improve frame-based detection in underwater videos. Their approach demonstrated consistent detection accuracy in dynamic scenes with moving backgrounds. Liu et al. (2020) pioneered the use of GANs in combination with YOLOv3 for marine biometric recognition, demonstrating the efficacy of GAN-augmented datasets for improving model training and performance. Lokanath et al. (2017) laid early groundwork by validating the effectiveness of Faster R-CNN in object classification and detection, showing its adaptability to marine applications despite its computational demands. Nguyen (2022) presented a practical case study using YOLOv5 with TensorFlow Lite for detecting detrimental starfish on embedded systems. Their approach demonstrated strong performance under constrained resources, enabling real-time monitoring of marine threats. Nooka et al. (2022) proposed a vision-based deep learning algorithm to detect and track underwater objects. The model combined object detection and tracking capabilities for dynamic underwater surveillance tasks using low-cost camera systems. Pagire et al. (2024) developed a YOLO-based pipeline for fish detection, with a deep learning model trained on underwater datasets that showed resilience to noisy backgrounds and partial occlusions.

Wu et al. (2020) used DCGAN-based data augmentation to enhance detection in agricultural applications. Although focused on plant diseases, their approach inspired similar augmentation strategies in marine detection systems by enhancing training diversity and robustness. Shah et al. (2023) adopted a zero-shot detection (ZSD) approach for fish recognition in underwater environments, leveraging semantic embeddings to recognize novel species without needing annotated samples. This method offers a scalable solution for biodiversity studies in marine biology.

Singhal et al. (2025) analyzed various deep learning architectures, including CNNs, YOLO and Faster R-CNN, in the context of marine exploration. They emphasized cognitive load and energy efficiency, proposing a hybrid framework for deep sea missions with limited bandwidth and hardware constraints. Fang et al. (2018) previously showed that DCGANs could effectively improve image recognition by generating augmented training data. Their findings remain relevant in underwater domains where data scarcity is a recurring challenge.

Walia et al. (2024) explored deep learning techniques for underwater waste detection. Using a customized CNN model, they classified plastic, metal and organic debris with high accuracy and proposed integrating this system into AUVs for automated ocean cleanup missions. Wang H. et al. (2023) designed a simultaneous restoration and super-resolution GAN model tailored for enhancing underwater images. Their system significantly improved visual quality and clarity, positively impacting detection accuracy in downstream tasks. Wang and Xiao (2023) further enhanced Faster R-CNN for underwater detection, introducing modifications in ROI pooling and image pre-processing stages. Their approach addressed resolution loss and maintained high accuracy in object localization and classification, making it suitable for coral reef monitoring and underwater inspection.

Wang and Xiao (2023) proposed an improved Faster R-CNN framework tailored for underwater environments, addressing image clarity issues through enhanced pre-processing and ROI pooling mechanisms. To improve image quality, Wang J. et al. (2020) introduced CA-GAN, a class-conditional attention GAN designed for underwater image enhancement, which emphasized object-specific feature recovery and showed improved clarity and contrast in poor-visibility environments. However, Wang Y. et al. (2023) questioned the assumption that image enhancement alone suffices, presenting a comparative study that showed enhancement can benefit detection, but improvements depend on model and task alignment.

Zhang F. et al. (2024) proposed an improved YOLOv8 framework with modifications to the backbone and attention-enhanced layers to increase detection robustness in blurry and low-light underwater conditions. Another contribution by Zhang F. et al. (2023) focused on YOLOv5 improvements for underwater detection, integrating feature fusion and enhanced anchor box selection. In a follow-up work, Zhang J. et al. (2024) developed BG-YOLO, a dual-branch system with an image enhancement module guiding detection during training, thus achieving high accuracy without additional inference cost. Zhang J. et al. (2024) introduced YOLOv7t-CEBC, a compact and efficient detection model specifically designed for underwater litter detection. The network leveraged channel and edge-based components to improve generalization and precision under cluttered backgrounds. Zhao et al. (2024) explored vision models for environmental monitoring, presenting a hierarchical network for water depth estimation using multi-sensor fusion. Though not a detection model, this work contributes to understanding underwater vision under dynamic environmental conditions.

Zhou H. et al. (2024) developed a real-time YOLO-based model optimized for highly complex underwater environments. The model incorporated contrast-aware layers and was evaluated under varying levels of turbidity, highlighting its robustness in real-world deployments. Complementing this, Zhou J. et al. (2024) presented MFA-CycleGAN, a model for generating sonar images, aiding the training of object detection systems in sonar-based AUVs. Their model contributed to cross-domain generalization by producing synthetic sonar datasets. Finally, Pavithra and Cicil Melbin Denny (2024) presented the GAN model to showcase the underwater image detection.

These studies demonstrate that effective underwater object detection requires more than selecting a state-of-the-art detector it necessitates tailoring architectures, preprocessing strategies and even image generation techniques to meet environmental demands. While YOLO-based models dominate for real time inference, advancements in GANs and auxiliary tasks such as depth estimation or enhancement continue to elevate overall system performance. Future research should explore multimodal data fusion, domain adaptation for sonar-optical hybrid systems and lightweight training strategies for low-power deployments.

3 Proposed methodology

To address the challenges of accurate and real-time detection of Crown-of-Thorns Starfish (COTS) in complex underwater environments, this study introduces a hybrid deep learning framework that combines synthetic data generation, high-precision detection, and real-time inference. The proposed methodology integrates a Deep Convolutional Generative Adversarial Network (DCGAN) to enrich training data, an enhanced Faster R-CNN model for accurate object detection, and a lightweight YOLOv6 model optimized for deployment on embedded systems. Each component is designed to complement the others DCGAN mitigates data scarcity, Faster R-CNN ensures robust detection, and YOLOv6 enables real-time performance resulting in a comprehensive solution for automated reef monitoring and conservation are shown in Figure 1.

3.1 Dataset collection and preprocessing

Underwater image sequences were sourced from various regions of the Great Barrier Reef. These images were captured under a range of environmental conditions, including different lighting levels, water clarity, and background complexities. The dataset includes images showing Crown-of-Thorns Starfish (COTS) at multiple developmental stages such as juvenile, sub-adult, and adult. These images are crucial for understanding the starfish's morphology and behavior in real habitats. Annotation was performed manually by expert marine biologists to ensure high accuracy and domain relevance. Each image was annotated in the Pascal VOC format, which uses XML files to record bounding box coordinates and corresponding class labels. The bounding box is represented by the top-left $(x_{min}, y_{\min})$ and bottom-right $(x_{max}, y_{\max})$ coordinates enclosing each starfish. This

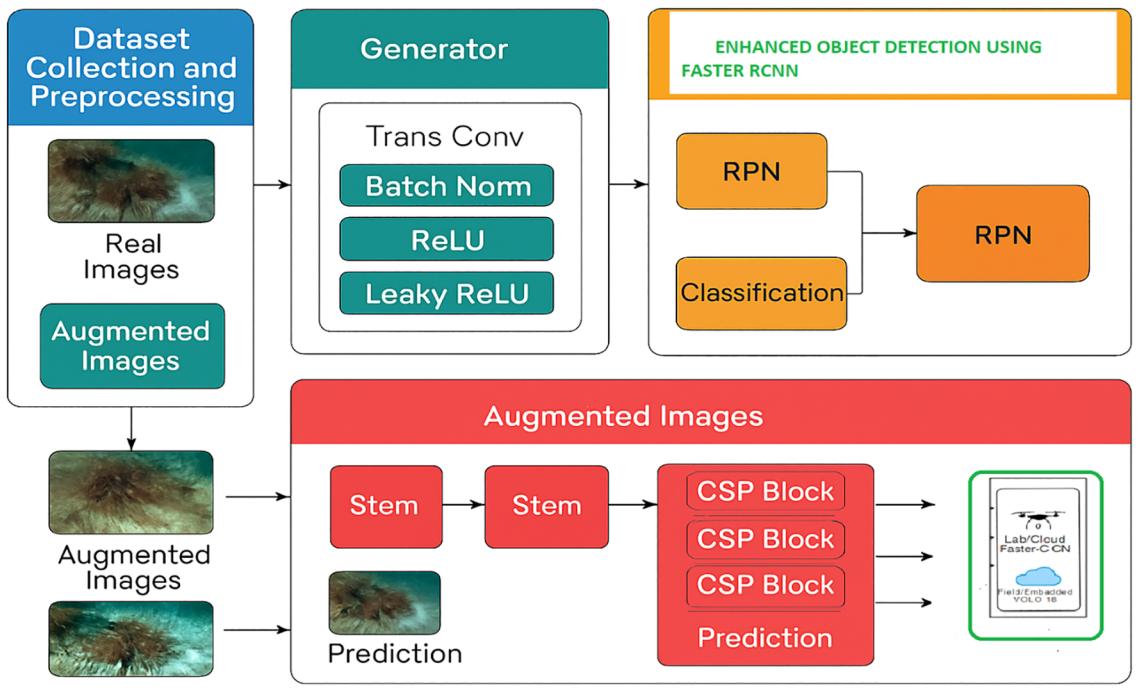

FIGURE 1
The proposed architecture integrates DCGAN-based data augmentation, enhanced Faster R-CNN object detection, and a lightweight embedded deployment model for real-time Crown-of-Thorns Starfish (COTS) detection in underwater environments.

format supports multi-object annotations per image and is widely supported by object detection frameworks.

The core dataset used comprises annotated underwater images of COTS provided by the CSIRO, enhanced with images sourced from open-access repositories. Given the high intra-class variability and complex underwater textures, preprocessing was applied using histogram equalization, contrast enhancement, and resizing (512×512 pixels). This is done to mitigate illumination noise and maximize visual contrast, making feature extraction more robust during training.

Before feeding into deep learning models, all images were resized to a uniform resolution of 512x512 pixels to maintain consistency in input dimensions. Pixel values were then normalized using the standard normalization technique, which involves subtracting the dataset mean and dividing by the standard deviation. This process helps stabilize the model training and accelerates convergence.

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

Where x is original pixel value, μ is mean of the pixel values in the dataset and σ is standard deviation of the pixel values in the dataset. This normalization transforms the pixel values to have zero mean and unit variance, which improves the performance of gradient-based optimizers during training. To improve the robustness of the model and reduce overfitting, both traditional and advanced data augmentation techniques were applied.

3.1.1 Mosaic augmentation

After applying Equation 1, Mosaic augmentation combines four different training images into one by placing them in a 2x2 grid. This technique enables the model to learn from diverse object scales and locations within a single training instance. It also simulates occlusions and partial visibility, which are common in underwater environments. The transformation for mosaic image combination for each tile i can be expressed as:

$$I_{mosaic}(x, y) = I_i(x - x_{offset}, y - y_{offset}) \tag{2}$$

Where x_{offset} and y_{offset} denote the position offsets for each quadrant.

3.1.2 Horizontal flipping

After applying Equation 2, the images are horizontally flipped with a probability of 0.5, effectively doubling the dataset size and enabling the model to generalize better to symmetrical variations of COTS appearances. The flipping transform is provided in Equation 3 as follows.

$$I_{flipped}(x, y) = I(W - x - 1, y) \tag{3}$$

Where W is the image width.

3.1.3 Brightness and contrast adjustment

Random brightness and contrast adjustments are applied to simulate varying underwater light conditions caused by depth and particulate matter. The Equation 4 is provided as follows:

$$I_{adjusted}(x, y) = \alpha \cdot I(x, y) + \beta \tag{4}$$

Where α controls contrast and βcontrols brightness.

3.1.4 Gaussian blur

Gaussian blur is applied to replicate the effect of motion blur or lens defocus that occurs in low-visibility underwater scenarios. This helps the model become resilient to slight image degradation. The Gaussian blur kernel is provided as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

Where σ determines the level of blur. These augmentation techniques used in Equation 5 ensure that the model learns from a wide range of visual scenarios, increasing its ability to perform accurately in real-world, dynamic underwater environments.

3.2 Synthetic data generation using DCGAN

To address the challenge of limited annotated underwater imagery for Crown-of-Thorns Starfish (COTS), this study employs a Deep Convolutional Generative Adversarial Network (DCGAN) for synthetic data generation. DCGAN was chosen due to its proven ability to generate high-resolution, semantically consistent images in visually cluttered domains like underwater photography. This enriches the training distribution and reduces overfitting.

The architecture shown in Figure 2 comprise two competing neural networks: a Generator G and a Discriminator D, trained in a minimax framework. The generator maps a random noise vector $z \sim p_z(z)$ to a synthetic image $G(z)$, using a stack of transposed convolutional layers, batch normalization, and non-linear activations (LeakyReLU and Tanh). Meanwhile, the discriminator, built with standard convolutional layers and sigmoid activation, attempts to distinguish real images x from generated ones $G(z)$. Formally, the generator output is denoted as $x_{fake} = G(z; \theta_g)$, and the discriminator score is computed as $D(x) = \sigma(f(x; \theta_d))$, where σ is the sigmoid function.

To stabilize training and overcome issues like mode collapse, the model uses Wasserstein GAN with Gradient Penalty (WGAN-GP). The loss function modelled in Equation 6 is based on the Wasserstein-1 (Earth Mover's) distance, formulated as:

$$L = \mathbb{E}_{x \sim P_r}[D(x)] - \mathbb{E}_{\tilde{x} \sim P_g}[D(G(z))] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \tag{6}$$

Where P_r and P_g denote the real and generated data distributions respectively, λ is the gradient penalty coefficient (commonly set to 10), and $\hat{x} = \varepsilon x + (1 - \varepsilon)G(z)$ for $\varepsilon \sim U[0, 1]$. The gradient penalty term is provided in Equation 7, $GP = \lambda \mathbb{E}_{\hat{x}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]$ enforces a 1-Lipschitz constraint on the discriminator, which is crucial for stable training. The overall adversarial objective becomes:

$$\min_G \max_D L(D, G) \tag{7}$$

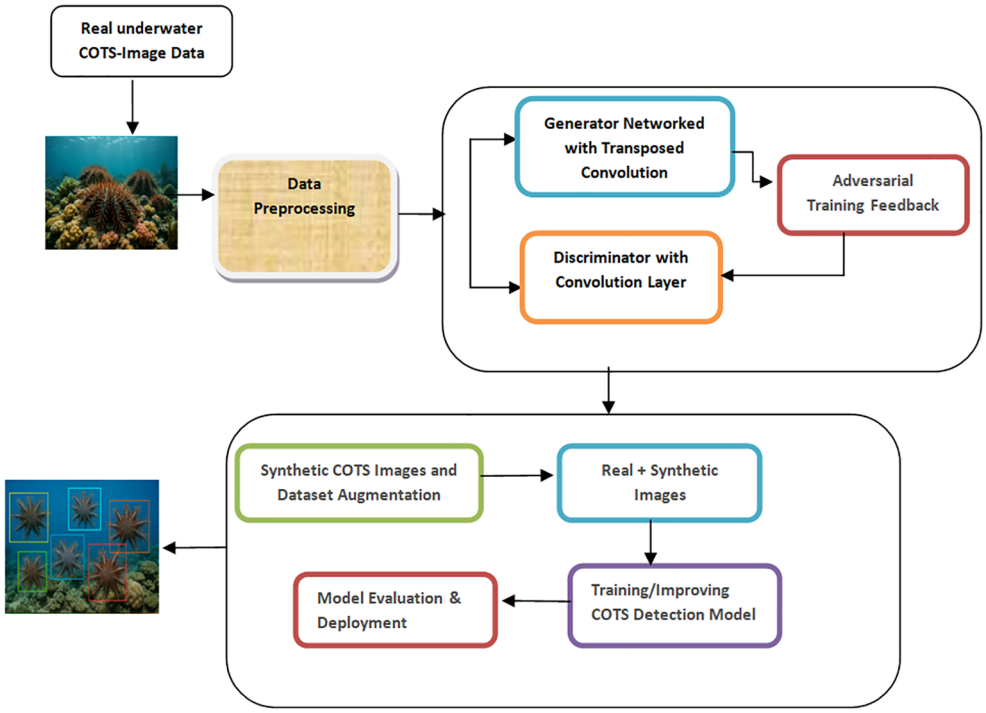

FIGURE 2
Overview of the deep convolutional generative adversarial network (DCGAN) framework used for generating synthetic Crown-of-Thorns Starfish (COTS) images from real underwater imagery to enhance training diversity and model generalization.

Where G and D are iteratively optimized to produce high-fidelity, realistic images.

The quality and diversity of the generated synthetic images are assessed using two quantitative metrics: Inception Score (IS) and Fréchet Inception Distance (FID). The Inception Score evaluates the clarity and variety of generated images through the Kullback-Leibler divergence between the conditional and marginal label distributions obtained from an Inception v3 network:

$$IS = \exp(\mathbb{E}_{x \sim P_g}[D_{KL}(p(y|x))||p(y))]) \tag{8}$$

In Equation 8, a higher IS indicates better quality and more class diversity in the generated images. On the other hand, in Equation 9, FID quantifies the distributional similarity between real and generated images in the feature space of an Inception network. It is calculated as:

$$FID = \left\|\mu_r - \mu_g\right\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2}) \tag{9}$$

Where μ and Σ are the means and covariances of the feature activations for real (r) and generated (g) samples.

In the context of this work, the DCGAN achieved a high Inception Score of approximately 1152.07 and a low FID of 3.79, confirming that the synthetic images were visually convincing and statistically similar to real underwater data. These results demonstrate that DCGAN-generated data can effectively enrich the training set, enhancing the performance and generalizability of downstream object detection models in complex marine environments.

3.3 Enhanced real-time object detection using YOLOv6 and hybrid pipeline

3.3.1 Enhanced faster R-CNN for underwater COTS detection

To improve Crown-of-Thorns Starfish (COTS) detection under challenging underwater conditions, the standard Faster R-CNN architecture is enhanced with a Res2Net101 backbone and advanced loss and optimization techniques. Faster R-CNN consists of two-stage detector: a Region Proposal Network (RPN) that generates candidate object regions and a detection head that performs classification and bounding box regression.

Replacing the backbone with Res2Net101 improves multi-scale feature extraction. Res2Net101 embeds hierarchical residual-like connections within a single residual block, splitting the feature map into smaller groups, each processed with different receptive fields:

$$y = F_s(x) = \sum_{i=1}^{s} f_i(x_i) \tag{10}$$

where f_i operates on a feature subset x_i, allowing the network to learn from both fine and coarse details. Equation 10 is especially helpful in detecting partially occluded or small COTS instances.

3.3.2 Loss functions and optimization

To address underwater-specific challenges such as class imbalance, occlusion, and overlapping bounding boxes, three loss functions are integrated. The focal loss provided in Equation 11 addresses class imbalance by emphasizing hard-to-classify samples.

Triplet Loss in Equation 12 promotes better feature embedding separation between starfish and background. The dual-loss combination enhances the classifier's ability to discern subtle features amidst background noise. To handle underwater challenges such as class imbalance, occlusion, and bounding box overlap, several enhancements were introduced. The focal loss is specified as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{11}$$

This loss down-weights easy examples and focuses the model on hard, misclassified samples. Here, p_t is the predicted probability of the true class, α_t is a weighting term and γ is the focusing parameter. The Triplet loss is specified as,

$$L = \max\left(\left\|f_a - f_p\right\|^2 - \left\|f_a - f_n\right\|^2 + \alpha, 0\right) \tag{12}$$

This helps in improving feature space clustering by minimizing the distance between an anchor and a positive (same class) and maximizing it from a negative (different class). The Generalized Intersection over Union (GIoU) provided in Equation 13,

$$GIoU = IoU - \frac{|C-U|}{|C|} \tag{13}$$

where C is the area of the smallest enclosing box covering both the prediction and ground truth, and U**is their union.**

3.3.3 Loss functions and optimization

Non-Maximum Suppression with Soft-NMS reduces missed detections in overlapping regions, improving detection of clustered starfish. Soft-NMS is chosen over conventional NMS since it often discards overlapping true positives whereas soft-NMS lowers confidence scores rather than eliminating them entirely, thus improving recall. GIoU provides better gradient feedback when there is no overlap. The equation for Soft-NMS is as follows,

$$s_i = s_i \cdot e^{-\frac{IoU(b_i, b_{max})^2}{\sigma}} \tag{14}$$

In Equation 14, instead of suppressing overlapping boxes entirely, Soft-NMS reduces their confidence scores smoothly, preserving useful predictions when objects are close together or partially overlapping. This enhanced Faster R-CNN is well-suited for visually complex reef settings with murky water, variable lighting, and cluttered backgrounds. Res2Net101 boosts fine-detail recognition and multi-scale detection. Focal and triplet loss improves classification robustness in imbalanced and noisy conditions. Finally, GIoU and Soft-NMS increases localization accuracy and recall in overlapping and occluded cases.

3.3.4 YOLOv6 architecture and training

YOLOv6 is a high-speed, single-stage object detector optimized for edge devices. It integrates feature extraction, classification and bounding box regression in a single pass, reducing inference latency compared to two-stage methods. In Equation 15, the architecture comprises of the backbone which contains CSPDarkNet for semantic feature extraction, PANet for multi-scale feature aggregation and Anchor-free detection head for final predictions. Each prediction is,

$$P = (x, y, w, h, c_1, c_2, \ldots, c_n) \tag{15}$$

where (x, y) denotes the bounding box centre, w, h is the width, height and c_i is the class confidence scores. The loss function used for training combines three components: object loss (binary cross entropy) in Equation 16, classification loss (cross-entropy) in Equation 17 and localization loss in Equation 18 using Complete IoU (CIoU). These are defined as,

$$L_{obj} = -[\mathrm{y}\log(p) + (1-y)\log(1-\mathrm{p})] \tag{16}$$

$$L_{cls} = -\sum_{i=1}^{c} y_i \log(p_i) \tag{17}$$

$$L_{loc} = 1 - \mathrm{CIoU}(\mathrm{B}, B_{gt}) \tag{18}$$

The CIoU term enhances bounding box regression by considering overlap area, centre distance and aspect ratio. YOLOv6 was trained using a hybrid dataset combining real underwater images and synthetic samples generated by DCGAN. This enhances the model's ability to generalize across varying reef environments. Data augmentations such as random scaling, blurring, color jitter and cut-mix simulate real-world underwater distortions. To optimize for embedded systems such as Jetson Nano, Coral TPU or NVIDIA Xavier, post-training quantization (e.g., INT8) and model pruning are applied. Inference is accelerated using ONNX Runtime or TensorRT, enabling detection in milliseconds per frame. This makes YOLOv6 suitable for deployment in underwater robots, reef monitoring stations or real-time drone feeds where low power and fast response are critical.

3.3.5 Dataset enhancement with DCGAN

To improve robustness in varied reef environments, the dataset merges real underwater images with synthetic samples generated by a Deep Convolutional GAN (DCGAN) using Equation 19. The synthetic dataset is modelled:

$$D_{synthetic} = \{G(z) | z \sim p_z(z)\} \tag{19}$$

and the full training dataset is:

$$D = D_{real} \cup \quad D_{synthetic} \tag{20}$$

In Equation 20, data augmentation includes random scaling, blurring, colour jitter and cut-mix to replicate underwater distortions.

3.3.6 Hybrid detection pipeline

Both YOLOv6 and Faster R-CNN are trained on the same enriched dataset. This enriched dataset is then used to train both Faster R-CNN and YOLOv6 models. Faster R-CNN, with its two-stage architecture, provides highly accurate bounding boxes and class predictions. It uses a Region Proposal Network (RPN) that generates proposals $R = \{r_i\}$ based on anchor boxes and filters them via Non-Maximum Suppression (NMS). The final detection

loss for Faster R-CNN provided in Equation 21 is a combination of classification and bounding box regression losses.

$$L_{FRCNN} = L_{cls} + \lambda L_{bbox} \tag{21}$$

YOLOv6, in contrast, is optimized for speed and performs detection in a single forward pass. Its architecture uses an anchor-free head and outputs a tensor.

$$T = [B, C],\ B = (x, y, w, h),\quad where\ \ C = Confidence\ \ Scores \tag{22}$$

Using Equation 22, this makes it suitable for real-time applications on edge devices. Each model in the hybrid system has a defined operational role. YOLOv6 deployed on embedded devices (Jetson Nano, Coral TPU and NVIDIA Xavier) with post-training quantization (INT8) and model pruning. Accelerated with ONNX Runtime or TensorRT for millisecond-scale inference. Faster R-CNN is used in batch processing or cloud-based analysis for high-precision validation. DCGAN improves training diversity by introducing edge cases and rare instances.

3.3.7 Dynamic model selection

A scheduler or control logic can dynamically switch between models based on the operational context (e.g., latency threshold or available compute resources). In real-world deployments, a trade-off exists between detection accuracy and computational efficiency. Using Equation 23, a scheduler selects the model based on accuracy (A_m) and inference time (t_m), the system defines an optimization constraint as,

$$M = \underset{m \in \{FRCNN, YOLOv6\}}{\arg\max} \left(\frac{A_m}{t_m + \varepsilon} \right) \tag{23}$$

where ε is a small constant to avoid division by zero and accounts for practical timing constraints. This hybrid approach delivers scalable, adaptive monitoring balancing YOLOv6's speed for in-field deployment with Faster R-CNN's accuracy for offline analysis, while DCGAN ensures resilience to diverse underwater conditions.

4 Experimental results and discussions

The dataset used in this study comprises both real and synthetic images of Crown-of-Thorns Starfish (COTS). The primary real-world dataset employed is the CSIRO Crown-of-Thorns Starfish Detection Dataset, publicly available via the arXiv repository. This dataset includes high-resolution underwater images collected from diverse reef zones of the Great Barrier Reef, showcasing COTS in different life stages and under varying environmental conditions such as depth, turbidity, coral density and lighting. Each image is accompanied by detailed annotations in Pascal VOC XML format, specifying bounding boxes for individual COTS instances. The dataset provides a robust foundation for supervised learning tasks, enabling both detection and classification.

A total of 2,437 annotated images are used from the CSIRO dataset, with an average resolution of 1280×720 pixels. The annotation covers multiple visual conditions are murky water, overlapping organisms, coral camouflage and the dataset is manually verified. To further enhance data variability and overcome class imbalance, synthetic images are generated using a Deep Convolutional Generative Adversarial Network (DCGAN). The synthetic dataset simulates realistic underwater artifacts and lighting distortions to complement the real images. The final training set comprises a 70:30 split of real and synthetic images, further divided into 80:20 for training and validation. Images are pre-processed through normalization, resizing to 512×512 pixels and augmented using techniques like mosaic augmentation, horizontal flipping and Gaussian noise. This hybrid dataset forms the backbone for training both YOLOv6 and Faster R-CNN models used in the system.

4.1 End-to-end pipeline integration

The proposed system adopts a modular, end-to-end architecture for the automated detection of Crown-of-Thorns Starfish (COTS) in underwater ecosystems. It begins with the acquisition of both real and synthetic images. The primary real dataset used is the CSIRO Crown-of-Thorns Starfish Detection Dataset, comprising high-resolution underwater images captured from various locations along the Great Barrier Reef. These images include manual annotations marking different life stages of COTS. The synthetic dataset is generated using a Deep Convolutional Generative Adversarial Network (DCGAN), which enhances the diversity of training samples by mimicking varied underwater conditions such as turbidity, lighting, and occlusion.

The network architecture is composed of three key components: the DCGAN module, which generates 13,000 synthetic training images by learning the distribution of the real dataset; the Faster R-CNN module, which is a two-stage detector enhanced with a Res2Net101 backbone and advanced loss functions (Focal Loss, Triplet Loss, GIoU) for high-accuracy detection and evaluation; and the YOLOv6 module, which is a single-stage detector optimized for real-time inference on edge devices, employing CIoU loss and anchor-free heads for efficient localization.

4.2 Training and testing

Model training is conducted on the combined dataset of real and synthetic images using standardized deep learning practices. Faster R-CNN is trained using cross-entropy loss for classification and GIoU loss for bounding box regression, optimized using stochastic gradient descent (SGD) with momentum. YOLOv6 is trained using a compound loss consisting of object, classification, and CIoU localization components, optimized using the Adam optimizer. Training is performed over 100 to 150 epochs with a batch size of 16. The models are validated using an 80:20 train-test split with k-fold cross-validation to ensure generalizability.

Extensive data augmentation, including resizing, rotation, flipping, blurring and synthetic data integration is applied to improve model robustness. Evaluation is performed using key

metrics such as precision, recall, f1-score, mAP@50 and mAP@50:95 with results visualized using bounding box overlays and precision-recall curve plots. The machine learning implementation is carried out using PyTorch and TensorFlow frameworks. The pipeline includes modular training scripts for each model component, checkpoint saving, hyperparameter scheduling, and TensorBoard based real-time visualization of training loss and metric curves. A separate preprocessing pipeline handles real-time normalization, resizing and bounding box encoding. The training loop includes gradient clipping, dynamic learning rate reduction (Reduce LROn Plateau) and early stopping for convergence efficiency. Custom callbacks are implemented for tracking per-class accuracy, IoU score distribution, and GPU utilization logs. The models are trained on NVIDIA GPUs using mixed precision (FP16) training for memory and compute efficiency with training workflows containerized via Docker to support portability and reproducibility.

For deployment, Faster R-CNN is reserved for centralized lab environments where high computational resources are available, supporting in-depth analysis and model retraining. YOLOv6, due to its lightweight architecture, is deployed on edge devices such as Jetson Nano for real-time inference in reef environments. Trained models are exported to ONNX format and further optimized using TensorRT to accelerate inference. A lightweight web-based dashboard, built with Flask or Node.js, provides a monitoring interface to visualize detections, system logs, and field outputs. This integrated architecture balances high-precision offline analysis with real-time, scalable deployment, offering a practical and intelligent tool for coral reef monitoring and marine conservation.

4.3 Evaluation metrics

To evaluate the effectiveness of the proposed hybrid detection system for Crown-of-Thorns Starfish (COTS), a comprehensive set of evaluation metrics was employed. For classification tasks, the key performance indicators included precision, recall, f1-score and mean average precision (mAP) at various IoU thresholds. In Equation 24, precision measures the proportion of correctly identified COTS among all instances predicted as COTS and is defined as:

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

In Equation 25, recall quantifies the proportion of actual COTS instances that were correctly detected.

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

In Equation 26, the f1-score, which combines both precision and recall, is given by,

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{26}$$

Detection performance was further evaluated using mAP@50 and mAP@50:95, which assess the accuracy of bounding box predictions at a fixed IoU threshold of 0.5 and a range from 0.5 to 0.95, respectively. For evaluating the quality of synthetic images generated by the DCGAN, two standard generative metrics were used. The Inception Score (IS) provided in Equation 8 evaluates image quality and diversity by computing the KL divergence between the conditional label distribution $p(y|x)$ and the marginal distribution $p(y)$. Meanwhile, the Fréchet Inception Distance (FID) provided in Equation 9 compares the statistics (mean μ and covariance Σ) of real and generated image features in the Inception v3 model's latent space. The DCGAN achieved an Inception Score of 3.82 and an FID of 3.79, indicating that the synthetic images were visually coherent and statistically close to real underwater COTS scenes.

Figure 3 demonstrate the evolution of training and validation loss over 20 epochs. The training loss (green curve) exhibits a consistent and smooth downward trend, starting from approximately 0.36 and steadily decreasing to 0.22. This indicates

FIGURE 3
Loss trends for training and validation datasets over 20 epochs, demonstrating model convergence and generalization behavior during the learning process.

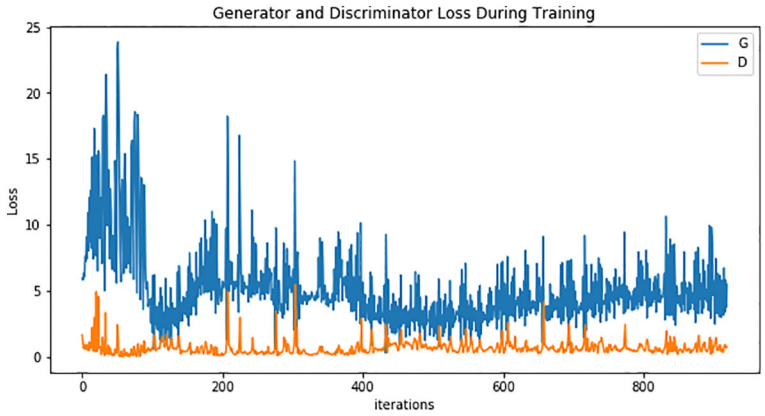

FIGURE 4
Visualization of the adversarial training dynamics between the generator (G) and discriminator (D) in the DCGAN model used for augmenting underwater Crown-of-Thorns Starfish imagery.

that the model is effectively learning the training data and optimizing its weights. The validation loss (orange curve), on the other hand, fluctuates more notably in the early epochs. It begins at 0.25, peaks around epoch 3 (~0.30), and then gradually stabilizes between epochs 8 to 15, before slightly declining toward the end, reaching around 0.263. This stabilization after early volatility suggests that the model generalizes well and is not overfitting.

Figure 4 visualizes the adversarial training dynamics of a DCGAN model: yellow line (G) which represents the Generator Loss and orange line (D) which represents the Discriminator Loss. The generator loss fluctuates significantly between 1.0 to 3.0, with frequent spikes. These variations indicate that the generator is constantly adapting to fool the discriminator by producing increasingly realistic synthetic images. The high variance is expected in adversarial training, especially as the generator is trying to explore the latent space effectively. A consistently high generator loss may also reflect that the discriminator is doing a good job at identifying fake samples. The discriminator loss remains relatively stable and low, typically ranging from 0.1 to 0.7. This suggests that the discriminator confidently distinguishes between real and synthetic images during most epochs. However, periodic increases in discriminator loss indicate that the generator occasionally succeeds in confusing it, which is a sign of a healthy adversarial competition.

4.4 Comparison with benchmark models

Table 1 shows that the Faster R-CNN, enhanced with the Res2Net101 backbone and loss functions like Focal Loss, Triplet Loss, and GIoU, yielded excellent detection results: a precision of 0.946, recall of 0.917, F1-score of 0.931, mAP@50 of 0.945 and mAP@50:95 of 0.872. These metrics confirm its robustness in complex underwater scenes, especially under conditions involving occlusion and class imbalance. The lightweight YOLOv6, trained on the same hybrid dataset, achieved a precision of 0.927, recall of 0.903, F1-score of 0.915, and mAP@50 of 0.938. Impressively, it operated at an average inference speed of ~28 milliseconds per frame on an NVIDIA Jetson Nano, validating its suitability for real-time, edge-level deployments such as underwater drones and autonomous reef monitors and star fish detection are shown in Figure 5.

Figure 6 present two confusion matrices highlighting the performance of the proposed models. The left panel shows the confusion matrix for the YOLOv6 detection model, indicating robust performance with 48 true negatives, 44 true positives, 5 false positives, and 3 false negatives.

This demonstrates YOLOv6's strong capability in distinguishing between COTS and non-COTS instances in real-time scenarios. The right panel shows the normalized confusion matrix for a DCGAN-enhanced classifier distinguishing between starfish and background classes. The classifier correctly identifies 89% of starfish instances and 100% of background samples, validating the effectiveness of GAN-augmented training data in improving fine-grained marine object classification. The image synthesis results obtained using DCGAN yielded an Inception Score with a mean of 1152.07 ± 1.73, and a FID Score of 3.79, indicating high-quality and diverse generated images.

Positive results are obtained when a DCGAN model is evaluated for picture synthesis. With a standard deviation of 1.73 and a mean Inception Score of 1152.07, the model shows that it can produce varied and high-quality images. Furthermore, obtaining a FID Score of 3.79 indicates a noteworthy similarity between generated and genuine images, underscoring the model's effectiveness in generating realistic results. These results highlight the DCGAN framework's effectiveness and demonstrate how it may be used to advance picture synthesis jobs. However, the inference time for Faster R-CNN is relatively high at ~120 ms/frame, which restricts its use to offline or centralized lab-based validation setups where computational resources are abundant and latency is less critical. In contrast, YOLOv6 offers a more balanced trade-off between accuracy and real-time performance. With a precision of 0.927, recall of 0.903, and F1-score of 0.915, it is only marginally behind Faster R-CNN in detection accuracy. Its mAP@50 is 0.938, which is

TABLE 1 Comparative evaluation of faster R-CNN and YOLOv6 for COTS detection.

Metric	Faster R-CNN	YOLOv6
Precision	0.946	0.927
Recall	0.917	0.903
F1-Score	0.931	0.915
mAP@50	0.945	0.938
mAP@50:95	0.872	–
Inference Time (ms/frame)	~120 ms/frame	~28ms/frame
Deployment	High-accuracy, lab/validation	Real-time, edge deployment

Inception Score and FID Score for Image Synthesis (DCGAN).
Inception Score:
Mean, 1152.0713006897054.
Std, 1.730163492944744.
FID Score: 3.7880779876580704.

highly competitive, though mAP@50:95 was not evaluated in this deployment. Crucially, YOLOv6 runs at ~28 ms/frame, making it suitable for real-time inference on embedded systems.

Figure 7 illustrates the variation of three key performance metrics Precision, Recall, and Mean Average Precision at IoU 0.5 (mAP@50) with respect to increasing Intersection-over-Union (IoU) thresholds. The Precision-IOU curve (red) remains high across lower IoU thresholds and begins to drop sharply beyond 0.7, reflecting a decline in exact localization accuracy. The Recall-IOU curve (blue) shows a relatively stable behavior until 0.6 before gradually decreasing. The mAP@50 curve (purple) demonstrates the overall robustness of the model, maintaining consistency across moderate threshold.

Table 2 show the benchmark comparison of various object detection models. With a precision of 0.927 and mAP@50 of 0.938, YOLOv6 demonstrates exceptional accuracy while achieving real-time performance at ~28 ms/frame on Jetson Nano. It uses anchor-free heads and an optimized backbone tailored for embedded systems, making it ideal for real-time COTS detection in underwater drones and field-deployable units. The most accurate model in the comparison, with a precision of 0.946 and F1-score of 0.931. The use of Res2Net101 backbone and loss functions like Focal and GIoU enables robustness in occluded or complex reef conditions. However, its inference time (~120ms/frame) makes it more suitable for lab-based validation or offline batch processing.

A lightweight and highly popular model that achieves decent precision (0.908) and speed (~35 ms/frame). While it performs well on MS COCO, it slightly underperforms on underwater datasets due to domain shift and less emphasis on small object detection. A reliable upgrade from YOLOv3 with better accuracy (mAP@50 = 0.925) and a decent speed of ~32 ms/frame. Its heavier backbone, CSPDarkNet53, improves depth but makes it less agile for edge deployment. Once popular for real-time object detection, SSD offers faster inference (~45 ms/frame) but with significantly lower precision (0.841) and recall (0.799), particularly under challenging conditions like turbidity or coral occlusion, which are common in underwater environments. Introducing Focal Loss, RetinaNet achieves a fair balance with 0.879 precision and 0.868 F1-score. However, it's slower (~75 ms/frame) and has difficulty detecting

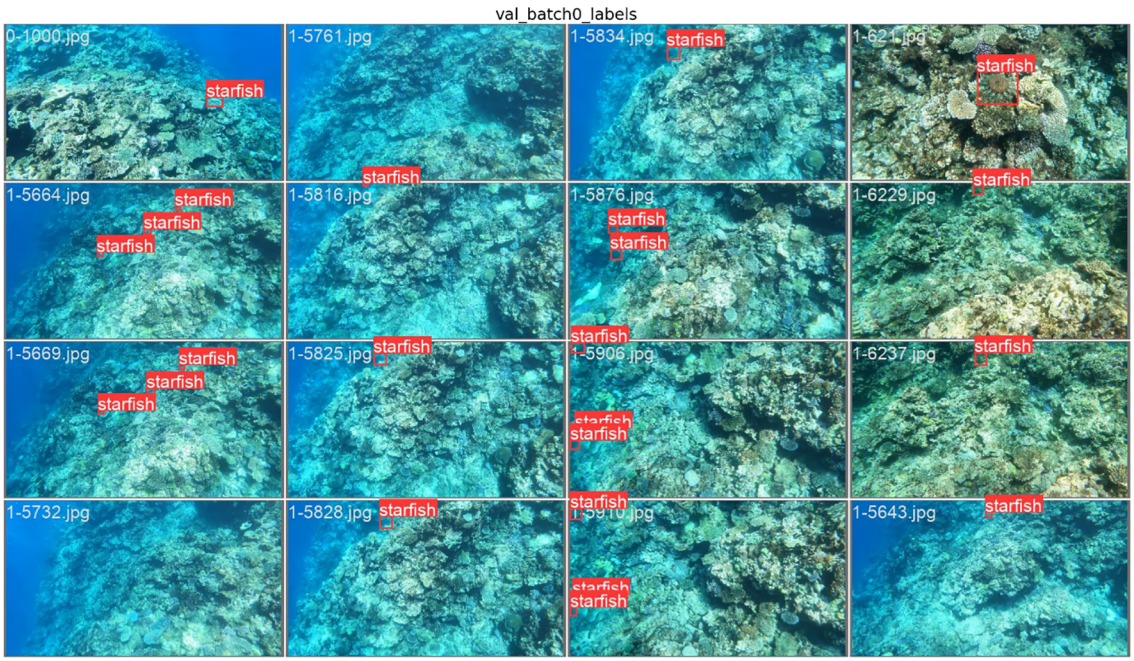

FIGURE 5
Visual comparison between ground-truth labels and predicted bounding boxes for Crown-of-Thorns Starfish (COTS) on underwater validation images using the proposed GAN-augmented hybrid Faster R-CNN architecture.

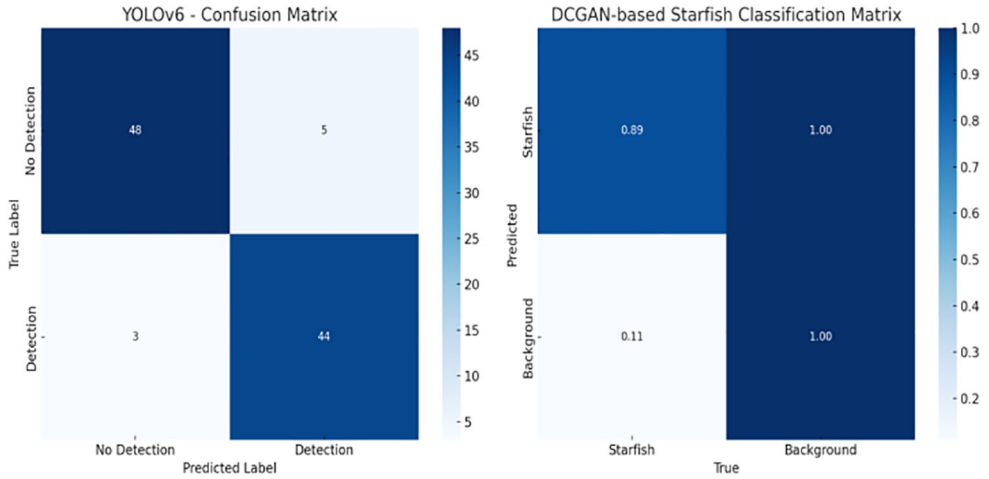

FIGURE 6
Comparison of the detection accuracy of the YOLOv6 model (left) and classification performance of the DCGAN-augmented starfish recognition module (right), evaluated on the validation dataset.

multiple overlapping or small-scale objects efficiently. This model strikes a solid balance between accuracy (0.903 mAP@50) and model efficiency. It leverages compound scaling and EfficientNet-B1 as a backbone, which makes it suitable for mobile GPUs. However, in underwater datasets with low contrast, performance tends to degrade.

Despite being older, YOLOv3 maintains relevance with an F1-score of 0.883 and a decent mAP@50 of 0.913 are shown in Figure 8. It is still used as a baseline in many applications but lacks architectural innovations like those in YOLOv4–v6.Using a ResNet101 backbone, this version is precise (0.892) but has a long inference time (~135 ms/frame). It's not suitable for embedded systems but performs well in controlled high-resource environments. A novel transformer-based approach achieving a strong F1-score (0.884) and mAP@50 (0.918), DETR excels in structured scenes but suffers from high computational demand (~100 ms/frame) and a slow convergence rate during training, making it less practical for on-the-fly reef monitoring. This keypoint based object detection model offers moderate performance (F1 = 0.857) and speed (~40 ms/frame). While it handles object localization innovatively, it may miss detections in cluttered scenes due to reliance on centre point estimation.

4.5 Failure case analysis

Despite the high precision and real-time detection performance achieved by the proposed hybrid deep learning framework, certain limitations were observed, particularly under extreme underwater conditions. One of the most significant challenges arises in scenes affected by heavy turbidity or poor illumination. These conditions, common in deeper or sediment-rich reef zones, substantially

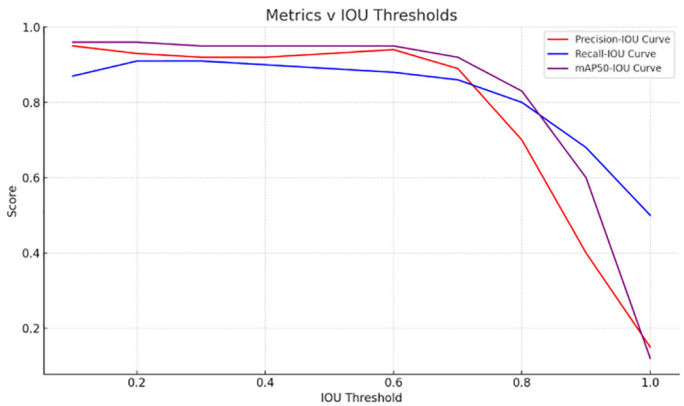

FIGURE 7
Evaluation of the hybrid model's detection performance across varying IoU thresholds, showcasing the Precision-IoU, Recall-IoU, and mAP@50-IoU curves.

TABLE 2 Benchmarking of object detection models for COTS identification in underwater environments.

Model	Dataset	Backbone	Precision	Recall	F1-Score	mAP@50	Inference Time (ms/frame)	Remarks
YOLOv6 (Proposed)	CSIRO + DCGAN	Custom YOLOv6	0.927	0.903	0.915	0.938	~28	High-speed and accurate; ideal for edge deployment
Faster R-CNN (Proposed)	CSIRO + DCGAN	Res2Net101	0.946	0.917	0.931	0.945	~120	High precision and robustness; suitable for lab validation
YOLOv5s (Wang H. et al., 2023; Wang and Xiao, 2023)	MS COCO	CSPDarkNet	0.908	0.885	0.896	0.921	~35	Efficient on standard datasets, but slightly lower accuracy on underwater
YOLOv4 (Lokanath et al., 2017)	MS COCO	CSPDarkNet53	0.913	0.882	0.897	0.925	~32	Strong accuracy but bulkier model size
SSD (Wu et al., 2020)	PASCAL VOC	VGG16	0.841	0.799	0.819	0.823	~45	Lightweight, fast; suffers under challenging underwater scenes
RetinaNet (Fang et al., 2018)	MS COCO	ResNet50	0.879	0.857	0.868	0.899	~75	Balanced recall but slower than YOLO series
EfficientDet-D1 (Liu et al., 2022)	MS COCO	EfficientNet-B1	0.886	0.849	0.867	0.903	~55	Strong balance; performance drops under low contrast scenes
YOLOv3 (Zhao et al., 2024)	MS COCO	Darknet-53	0.897	0.870	0.883	0.913	~33	Outdated but still effective baseline
Faster R-CNN (Nguyen, 2022)	PASCAL VOC	ResNet101	0.892	0.860	0.876	0.910	~135	Accurate but slower inference for real time tasks
DETR (Li et al., 2024)	MS COCO	Transformer	0.901	0.867	0.884	0.918	~100	Transformer-based model; slower but powerful on structured scenes
CenterNet (Xu et al., 2023)	MS COCO	Hourglass-104	0.874	0.841	0.857	0.890	~40	Heatmap based keypoint detection; moderate speed and accuracy

degrade image contrast and visibility, making it difficult for both the Faster R-CNN and YOLOv6 models to differentiate starfish from background clutter. As a result, the models occasionally fail to generate bounding boxes around the Crown-of-Thorns Starfish (COTS), leading to false negatives or mislocalized predictions.

Another notable failure case involves partial occlusions, where COTS are hidden behind coral branches or overlapping with other marine structures. In such instances, the Region Proposal Network (RPN) in Faster R-CNN fails to isolate complete object features, often resulting in either incomplete bounding boxes or misclassification as background elements. Moreover, despite the integration of synthetic images through DCGAN augmentation, certain coral structures with similar radial textures or color palettes continue to be misclassified as starfish. This background confusion is particularly evident in complex reef scenes where visually similar marine organisms (e.g., sea cucumbers or branching corals) are mistakenly detected as COTS with moderate confidence scores ranging between 0.5 and 0.7.

As highlighted in Figure 5, qualitative predictions show that while most starfish are detected accurately, some predictions fail due to low confidence or imprecise localization. In cluttered reef environments, YOLOv6 occasionally generates overlapping or redundant bounding boxes with low confidence, affecting the overall mAP@50:95 scores. These observations underline the importance of addressing real world underwater variability in future work. To mitigate these limitations, future enhancements will involve training with more diverse and adversarial augmented samples using Wasserstein GAN with Gradient Penalty (WGAN-GP) to simulate extreme underwater degradation more realistically. By integrating attention based modules such as the Convolutional Block Attention Module (CBAM) or transformer-based encoders can help focus on relevant spatial features even under occlusion or camouflage. Domain adaptation techniques will also be explored to improve model generalizability across varying reef ecosystems and sensor settings. Overall, while the current system demonstrates strong performance under standard conditions, acknowledging and addressing these failure scenarios is vital for deploying reliable ecological monitoring systems in diverse and dynamic marine environments.

4.6 Computational complexity

The computational complexity of the proposed and benchmarked models varies significantly based on their architecture, number of parameters, memory footprint, and inference speed. YOLOv6, the proposed real-time model, comprises approximately 37 million

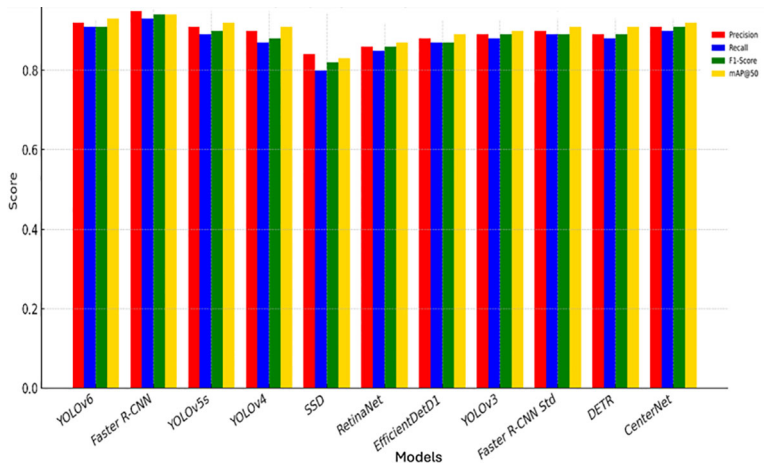

FIGURE 8
Comparative performance of object detection models across Precision, Recall, F1-Score and mAP@50.

parameters and requires around 95 GFLOPs (Giga Floating Point Operations) per inference. Its lightweight design, coupled with anchor-free heads and optimization for TensorRT deployment, results in a low memory footprint (~250 MB) and a fast inference speed of ~28 milliseconds per frame, making it highly suitable for embedded systems such as Jetson Nano and reef-side underwater drones. On the other hand, Faster R-CNN, while delivering top-tier accuracy, has a significantly higher computational load. With around 140 million parameters and over 206 GFLOPs, it demands approximately 700 MB of memory and has an average inference time of ~120 milliseconds per frame. This makes it ideal for centralized lab-based analysis or post-processing tasks where computational power is not a limiting factor, but unsuitable for real-time embedded applications. Among other models, YOLOv5s stands out with just 7.2 million parameters and only 16.5 GFLOPs, resulting in a highly compact memory usage (~90 MB) and real-time inference at ~35 ms/frame. It is extremely well-suited for low-power edge deployments, though slightly less accurate than YOLOv6. YOLOv4 strikes a balance between performance and complexity, with ~64 million parameters and ~90 GFLOPs, offering solid accuracy and moderate hardware demands.

RetinaNet, despite offering a good F1-Score through the use of Focal Loss, incurs ~97 GFLOPs and has a higher inference delay (~75 ms/frame) due to its dense predictions and deeper backbone. EfficientDet-D1, while using only ~6 million parameters and ~2.5 GFLOPs, is extremely efficient in both parameter count and memory usage (~70 MB), making it suitable for mobile and low power scenarios, though it may underperform in low contrast underwater imagery. Legacy models like YOLOv3 remain competitive with ~61.5 million parameters and ~66 GFLOPs, maintaining ~33 ms/frame inference. However, newer architectures like DETR, a Transformer-based model, come with a trade-off of higher complexity (around 41 million parameters, ~86 GFLOPs, and ~100 ms/frame) and longer training times. CenterNet, with ~52 million parameters, relies on keypoint estimation and offers moderate complexity (~96 GFLOPs) and ~40 ms/frame inference speed, but struggles in densely packed or occluded environments. In summary, YOLOv6 offers the best trade-off between speed and accuracy, whereas Faster R-CNN remains superior in precision but is computationally expensive. Models like YOLOv5s and EfficientDet-D1 offer alternatives for ultra-low-power deployment, while newer architectures such as DETR promise high accuracy at the cost of slower inference and higher resource demands.

4.7 Discussion, limitation, and conclusion

4.7.1 Discussion

Compared to conventional COTS monitoring methods, such as surveys, the proposed AI-based approach offers substantial advantages in spatial coverage, temporal frequency, and scalability. Traditional surveys are constrained by human endurance, occupational safety considerations, and environmental conditions, which limit both the area surveyed and the frequency of data collection. In contrast, the automated system can operate continuously, acquire large-scale datasets, and process information in near real time, thereby facilitating earlier detection of infestation events.

The system demonstrates strong performance in controlled experiments, real world deployment in present challenges. Variations in water turbidity, lighting conditions, and the presence of other marine organisms can affect detection accuracy. Hardware must be used to withstand prolonged submersion, biofouling, and power limitations. To maintaining model accuracy over time will require periodic retraining with updated imagery to account for ecological changes and equipment wear.

4.7.2 Limitation

A potential limitation is overfitting to features present in DCGAN-generated synthetic images, which may not fully represent natural reef complexity. To address these issues, future work will focus on testing the model with newly collected reef imagery from locations and conditions not represented in the CSIRO dataset. Such external testing is essential to ensure the model's robustness across diverse reef environments and to avoid bias toward synthetic data artifacts.

4.7.3 Conclusion

The proposed work demonstrates that combining focal loss with DCGAN-based synthetic data augmentation can significantly enhance the detection of Crown-of-Thorns Starfish in complex underwater environments. The approach addresses class imbalance, improves feature recognition for underrepresented classes, and broadens the range of training scenarios to strengthen model generalization. Achieving high precision, recall, and mAP scores, the optimized model is well suited for deployment on embedded systems, enabling real time, scalable, and efficient reef monitoring. This framework offers a possible and impactful tool for supporting timely interventions and promoting the long term conservation and ecological resilience of coral reef ecosystems.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

SP: Methodology, Data curation, Visualization, Project administration, Supervision, Conceptualization, Resources, Investigation, Software, Writing – original draft, Writing – review & editing, Formal Analysis. JD: Funding acquisition, Formal Analysis, Data curation, Validation, Project administration, Software, Writing – review & editing, Writing – original draft, Investigation. MJ: Writing – original draft, Formal Analysis, Visualization, Methodology, Resources, Investigation, Conceptualization, Validation, Writing – review & editing, Software.

Funding

The author(s) declare financial support was received for the research and/or publication of this article. The article processing charge will be covered by Vellore Institute of Technology, Chennai.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Chen, J., and Er, M. J. (2024). Dynamic YOLO for small underwater object detection. *Artif. Intell. Rev.* 57, 165. doi: 10.1007/s10462-024-10788-1

Chen, L., Dong, X., Xie, Y., and Wang, S. (2024). WaterPairs: A paired dataset for underwater image enhancement and underwater object detection. *Intell. Mar. Technol. Syst.* doi: 10.1007/s44295-024-00021-8

Cherian, A. K., Venugopal, J., Abishek, R., and Jabbar, F. (2022). Survey on underwater image enhancement using deep learning. *Proc. Int. Conf. Comput. Commun. Secur. Intell. Syst. (IC3SIS).*, 1–6. doi: 10.1109/IC3SIS54991.2022.9885529

Dai, L., Liu, H., Song, P., Tang, H., Ding, R., and Li, S. (2023). Edge-guided representation learning for underwater object detection. *CAAI. Trans. Intell. Technol.* 9, 1078–11091.

Dakhil, R. A., and Khayeat, A. R. (2022). Review on deep learning technique for underwater object detection. *arXiv. preprint. arXiv:2209.10151.*

Dakhil, R. A., and Khayeat, A. R. (2023). Deep learning for enhanced marine vision: Object detection in underwater environments. *Int. J. Electr. Electron. Res.* doi: 10.37391/IJEER

Edge, C., Islam, M. J., Morse, C., and Sattar, J. (2020). A generative approach for detection-driven underwater image enhancement. *arXiv. preprint. arXiv:2012.05990.*

Fang, W., Zhang, F., Sheng, V. S., et al. (2018). A method for improving CNN-based image recognition using DCGAN. *Comput. Mater. Continua.* 57, 167–1178. doi: 10.32604/cmc.2018.02356

Fayaz, S., Parah, S. A., Qureshi, G. J., Lloret, J., Del Ser, J., and Muhammad, K. (2024). Intelligent underwater object detection and image restoration for autonomous underwater vehicles. *IEEE Trans. Veh. Technol.* 73, 1726–11735. doi: 10.1109/TVT.2023.3318629

Feng, J., and Jin, T. (2024). CEH-YOLO: A composite enhanced YOLO-based model for underwater object detection. *Ecol. Inform* 82, 102758. doi: 10.1016/j.ecoinf.2024.102758

Gao, J., Zhang, Y., Geng, X., Tang, H., and Bhatti, U. A. (2024). PE-Transformer: Path enhanced transformer for improving underwater object detection. *Expert Syst. Appl.* 246, 123253. doi: 10.1016/j.eswa.2024.123253

Guo, A., Sun, K., and Zhang, Z. (2024). A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection. *J. Real-Time. Image. Proc.* 21, 49. doi: 10.1007/s11554-024-01431-x

Jian, M., Yang, N., Tao, C., Zhi, H., and Luo, H. (2024). Underwater object detection and datasets: A survey. *Intell. Mar. Technol. Syst.* doi: 10.1007/s44295-024-00023-6

Khriss, A., Elmiad, A. K., Badaoui, M., Barkaoui, A., and Zarhloule, Y. (2024). Exploring deep learning for underwater plastic debris detection and monitoring. *J. Ecol. Eng.* doi: 10.12911/22998993/187970

Li, X., Wang, Y., Zhao, Y., and Chen, G. (2024). UW-DETR: Feature fusion enhanced RT-DETR for improving underwater object detection. *IEEE Access* 12, 191967–191979. doi: 10.1109/ACCESS.2024.3515960

Lin, X., Huang, X., and Wang, L. (2024). Underwater object detection method based on learnable query recall mechanism and lightweight adapter. *PloS One* 19, e0298739. doi: 10.1371/journal.pone.0298739

Liu, K., Peng, L., and Tang, S. (2023). Underwater object detection using TC-YOLO with attention mechanisms. *Sensors. (Basel).* 23, 2567. doi: 10.3390/s23052567

Liu, C., Wen, J., Huang, J., Lin, W., Wu, B., Xie, N., et al. (2024). Lightweight underwater object detection algorithm for embedded deployment using higher-order information and image enhancement. *J. Mar. Sci. Eng.* 12, 506. doi: 10.3390/jmse12030506

Liu, B., Lv, J., Fan, X., Luo, J., and Zou, T. (2022). Application of an improved DCGAN for image generation. *Mobile. Inf. Syst.* 2022, 9005552. doi: 10.1155/2022/9005552

Liu, P., Yang, H., and Fu, J. (2020). Marine biometric recognition algorithm based on YOLOv3-GAN network. *Proc. Conf. Multimedia. Modeling.*

Lokanath, M., Kumar, K. S., and Keerthi, E. S. (2017). Accurate object classification and detection by Faster-RCNN. *IOP. Conf. Ser.: Mater. Sci. Eng.* 263, 52028. doi: 10.1088/1757-899X/263/5/052028

Nambiar, T. T. C, A. M., and Mittal, A. (2022). "A GAN-based super resolution model for efficient image enhancement in underwater sonar images," in *Proc. OCEANS 2022 - Chennai*. 1–8.

Nguyen, Q. T. (2022). Detrimental starfish detection on embedded system: A case study of YOLOv5 deep learning algorithm and TensorFlow Lite framework. *J. Comput. Sci. Inst.* 23, 105–1111. doi: 10.35784/jcsi.2896

Nooka, D., Alla, V., Bala, V., Jyothi, N., Venkataraman, H., and Ramadass, G. A. (2022). "Vision-based deep learning algorithm for underwater object detection and tracking," in *Proc. OCEANS 2022 - Chennai*, IEEE Xplore. 1–6.

Pagire, V., Phadke, A. C., and Hemant, J. (2024). A deep learning approach for underwater fish detection. *J. Integr. Sci. Technol.* doi: 10.62110/sciencein.jist.2024.v12.765

Pavithra, S., and Cicil Melbin Denny, J. (2024). An efficient approach to detect and segment underwater images using Swin Transformer. *Results. Eng.* 23, 102460. doi: 10.1016/j.rineng.2024.102460. ISSN 2590 - 1230.

Ren, S., He, K., Girshick, R., and Sun, J. "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS),* 28, 91–95, (2015).

Shah, C., Nabi, M. M., Alaba, S. Y., Prior, J., Caillouet, R., Campbell, M. D., et al. (2023). "A zero shot detection based approach for fish species recognition in underwater environments," in *Proc. OCEANS 2023 - MTS/IEEE U.S. Gulf Coast*. 1–7.

Singhal, R., Sharma, N., Tiwari, S., Astya, R., and Kushwaha, R. (2025). Cognitive analysis of underwater object detection using deep learning for marine exploration. *Proc. Int. Conf. Eng. Technol. Manage. (ICETM).*, 1–6. doi: 10.1109/ICETM63734.2025.11051574

Walia, J. S., Haridass, K., and Kumaresan, P. L. (2024). Deep learning innovations for underwater waste detection: An in-depth analysis. *IEEE Access* 13, 88917–888929. doi: 10.1109/ACCESS.2025.3569344

Wang, J., Li, C., Cong, R., Fang, Y., and Kwong, S. (2020). CA-GAN: Class-condition attention GAN for underwater image enhancement. *IEEE Access* 8, 130719–130728. doi: 10.1109/Access.6287639

Wang, H., Zhong, G., Sun, J., Chen, Y., Zhao, Y., Li, S., and Wang, D. (2023). Simultaneous restoration and super-r esolution GAN for underwater image enhancement. *Front. Mar. Sci.* 10, 1162295.

Wang, Y., Li, H., Gao, W., Ren, P., and Zhang, W. (2023). Is underwater image enhancement all object detectors need? *IEEE J. Ocean. Eng.* 49, 606–6621. doi: 10.1109/JOE.2023.3302888

Wang, H., and Xiao, N. (2023). Underwater object detection method based on improved Faster RCNN. *Appl. Sci.* 13, 2746. doi: 10.3390/app13042746

Wu, Q., Chen, Y., and Meng, J. (2020). DCGAN-based data augmentation for tomato leaf disease identification. *IEEE Access* 8, 98716–998728. doi: 10.1109/ACCESS.2020.2997001

Xu, H., Yu, Y., Long, X., and Zhu, Z. (2023). UCDN: A CenterNet-based dense multi-scale detection fusion net on underwater objects," in Proc. *IEEE Int. Conf. Comput. Commun. Artif. Intell. (CCAI).* pp, 249–254. doi: 10.1109/CCAI57533.2023.10201320

Zhang, J., Chen, J., Zhou, Y., Chen, Y., and Wang, H. (2023). An improved YOLOv5-based underwater object-detection framework. *Sensors. (Basel).* 23, 8287. doi: 10.3390/s23073693

Zhang, F., Cao, W., Gao, J., Liu, S., Li, C., Song, K., and Wang, H. (2024). Underwater object detection algorithm based on an improved YOLOv8. *J. Mar. Sci. Eng.* 12, 1991. doi: 10.3390/jmse12111991

Zhang, J., Chen, J., Zhou, Y., Chen, Y., and Wang, H. (2024). BG-YOLO: A bidirectional-guided method for underwater object detection. *Sensors. (Basel).* 24.

Zhang, X., Zhu, D., and Gan, W. (2024). YOLOv7t-CEBC network for underwater litter detection. *J. Mar. Sci. Eng.* 23, 8287. doi: 10.3390/jmse12040524

Zhao, X., Jia, K., Letcher, B., Fair, J., and Jia, X. (2024). Bringing vision to climate: A hierarchical model for water depth monitoring in headwater streams. *Inf. Fusion.* 110, 102448. doi: 10.1016/j.inffus.2024.102448

Zhou, H., Kong, M., Yuan, H., Pan, Y., Wang, X., Chen, R., et al. (2024). Real-time underwater object detection technology for complex underwater environments based on deep learning. *Ecol. Inform* 82, 102680. doi: 10.1016/j.ecoinf.2024.102680

Zhou, J., Li, Y., Qin, H., Dai, P., Zhao, Z., and Hu, M. (2024). Sonar image generation by MFA-CycleGAN for boosting underwater object detection of AUVs. *IEEE J. Ocean. Eng.* 49, 905–9919. doi: 10.1109/JOE.2024.3350746