Proceedings of 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS 2024)
IEEE Xplore Part Number: CFP24DJ3-ART; ISBN: 979-8-3503-7999-0

Advancing Underwater Trash Detection: Harnessing Mask R-CNN, YOLOv8, EfficientDet-D0 and YOLACT

Ritik Jain
Dept. Of Computer Engineering
AISSMS IOIT
Pune, India
ritikjain7350@gmail.com

Dr.Sarika Zaware
Dept. Of Computer Engineering
AISSMS IOIT
Pune, India
sarika.zaware@aissmsioit.org

Niraj Kacholia
Dept. Of Computer Engineering
AISSMS IOIT
Pune, India
niraj42002@gmail.com

Het Bhalala
Dept. Of Computer Engineering
AISSMS IOIT
Pune, India
hetbhalala81@gmail.com

Oam Jagtap
Dept. Of Computer Engineering
AISSMS IOIT
Pune, India
oam.jagtap@gmail.com

2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS) | 979-8-3503-7999-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICSCSS60660.2024.10625362

***Abstract*—Every year, millions of tons of trash and other pollutants are introduced into the ocean. Marine animals and coral reefs face significant threats to their lives due to human negligence in properly disposing of trash. Millions of marine life perish every single year due to the contamination of water bodies. Before embarking on this research, several challenges were faced, including the scarcity of high-quality underwater trash datasets, the complexity of underwater environments, and the limitations of existing detection models in terms of accuracy and speed. Additionally, selecting suitable deep learning models presented significant hurdles due to the varying availability of pretrained models, differing capabilities in handling underwater imagery, and the trade-offs between detection accuracy and computational efficiency. Amid these increasing environmental concerns, this research explores novel deep learning models for detecting underwater trash namely Mask R-CNN, YOLOv8, EfficientDet-D0, and YOLACT, aiming to contribute to cleaner water bodies. These models were trained on the benchmark TrashCan 1.0 dataset. The pretrained weights of the COCO 2017 dataset were used for transfer learning. The study found that YOLOv8 has outperformed all the remaining models in terms of mAP, Recall, and F1 Score. YOLOv8 achieved the highest mAP (0.714), outperforming Mask R-CNN (0.627), EfficientDet-D0 (0.560), and YOLACT (0.542). Notably, YOLOv8 trained 3 to 10 times faster than other models. This accelerated training time positions YOLOv8 as the preferred choice for real-world deployments of Autonomous Underwater Vehicles(AUVs) where rapid adaptation to new types of trash is essential. On the other hand, Mask R-CNN performs well at detecting small trash in complex underwater scenes. This detailed analysis can be used to improve the training data for YOLOv8, potentially boosting its accuracy in detecting smaller objects and complex underwater environments while maintaining its real-time processing advantage. This comprehensive analysis paves way for further advancements in trash detection algorithms, fostering a more effective mitigation strategy against ocean pollution.**

***Index Terms*—Underwater Trash, Trash Detection,Deep Learning Models, Mask R-CNN, YOLOv8, EfficientDet-D0, YOLACT, Ocean Pollution, TrashCan 1.0.**

I. Introduction

The escalating concern surrounding underwater trash poses a substantial environmental threat, putting ecosystems and marine life at risk. Anthropogenic activities, such as irresponsible waste disposal and littering, contribute significantly to the accumulation of debris in aquatic environments. Traditional methods of waste detection and removal in underwater settings, including human divers and manual sampling, prove to be labor-intensive, expensive, and potentially hazardous. Consequently, there is a pressing need for efficient, faster, cost-effective, and automated approaches to identify and monitor underwater trash.

The alarming pervasiveness of plastic waste is a growing environmental threat. Plastic debris litter even the most far-flung corners of our planet, from the peak of Mount Everest to the deepest trenches of the Indian Ocean. A staggering estimate suggests 5.25 trillion plastic items are polluting our oceans, and this concerning number continues to rise each day threatening the lives of millions of marine animals. Wetlands, vital ecosystems, are disappearing at an alarming rate, with losses equivalent to the ecological value of vast areas. This environmental degradation disproportionately affects hundreds of millions of people who rely on fish for protein. A staggering 3.1 billion people could face limitations in obtaining sufficient animal protein due to polluted waters. Recent research by Oceans Asia, a marine conservation organisation [7] highlights a new concern, disposable masks. With an estimated 1.56 billion masks discarded into oceans in 2020 alone, these plastic pollutants take centuries to degrade and potentially endanger marine life. The urgency to protect our water resources is undeniable.

Underwater environments present a formidable challenge

for object detection models. Fluctuating light availability, light scattering that induces image distortion, and refraction that bends light rays as they traverse water all contribute significantly to degraded image quality. Further complicating this task is the presence of suspended particles within the water column. These particles not only reduce visibility but also introduce noise into the captured imagery. The constant introduction of new debris necessitates models capable of adapting to an ever-expanding array of waste items. Furthermore, the limited availability of annotated underwater trash datasets hinders the development of robust models. Finally, the ability of certain types of trash to mimic marine life introduces an additional classification hurdle for detection models, making it difficult to distinguish between the two.

II. Literature Survey

Deep learning, particularly through convolutional neural networks (CNNs), has revolutionised the field of computer vision. These networks have shown remarkable capabilities, especially in healthcare, where they enable high-accuracy predictions of conditions like heart disease by analysing medical images. The strength of CNNs lies in their ability to automatically learn and classify features from images, making them indispensable for image classification tasks.

Region-based convolutional neural networks (R-CNNs) represent a significant advancement in object detection. Despite their effectiveness, R-CNNs are hampered by slow processing times due to the reliance on selective search, a technique that can take 40-50 seconds per image. To overcome this limitation, the Faster R-CNN was developed. This improved model integrates a Region Proposal Network (RPN) directly into the object detection pipeline, eliminating the need for external region proposal methods. This integration not only speeds up the processing time but also maintains high detection accuracy, making Faster R-CNN a more efficient tool for object detection tasks. [8].

In the realm of environmental monitoring, deep learning has led to significant advancements, particularly in detecting underwater objects, fish, oil spills, and marine pollution. Researchers have introduced various innovative solutions to enhance the precision and effectiveness of detection systems. For instance, modifications to the Faster R-CNN [23], incorporating Res2Net101, [1]have been introduced to improve underwater object detection. In fish detection, a hybrid approach combining Gaussian Mixture Models and optical flow has been suggested to effectively recognize fish in unconstrained underwater videos [2]. Additionally, hybrid models that integrate CNNs for feature extraction with k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) for classification have shown promise in accurately identifying underwater fish species [3].

Researchers have also tackled specific environmental challenges with tailored solutions. In fish farming, for example, an enhanced YOLO-V4 network has been used to identify leftover food scraps, significantly improving detection accuracy from 65.4% to 92.6%, [4]. For oil pollution, a comprehensive approach combining image processing, SVM, and optical flow has been proposed to accurately identify and monitor oil pollution areas in the ocean [5]. In marine plastic detection, deep learning models like YOLOv4 and YOLOv5 have been explored, achieving real-time accuracy ranging from 40% to 80% [6]. For waste management, a modified YOLOv4 has been developed for high-speed and high-precision garbage detection, which has been implemented in an autonomous garbage-collecting robot [7]. Turning attention to specific ecological concerns, [8] seagrass detection has been addressed through the introduction of an Inception V2-based Faster R-CNN network, achieving high precision in detecting Halophila ovalis.

Comparative analyses of CNN models reveal that Mask R-CNN is among the most accurate due to its ability to combine object detection with pixel-level segmentation, leading to detailed object boundaries and segmentation. However, this accuracy comes at a higher computational cost. On the other hand, earlier versions of the YOLO model, such as YOLOv2, YOLOv3, and YOLOv5, have proven effective in balancing computational efficiency and performance. While these models deliver accurate results, their accuracy falls short compared to Mask R-CNN, especially in the context of underwater trash detection.

While Faster R-CNN offers significant speed improvements over R-CNN, it still requires substantial computational resources. Similarly, YOLO models are praised for their real-time performance and efficiency, yet their accuracy, particularly in complex environments like underwater settings, may not be as high as more computationally intensive models like Mask R-CNN. These discussions help identify areas where enhancements can be made, such as improving accuracy without sacrificing speed or reducing computational requirements without significantly impacting performance.

III. Technique description

This research aimed to identify potential candidate models capable of surpassing or matching the performance of Mask R-CNN models for underwater object detection. These models, previously unused in underwater object detection, have demonstrated notably strong results in other object detection tasks. The objective was to determine which of these models is most suitable for this specific task. The methodology involves the training of a Convolutional Neural Network (CNN) on the TrashCan 1.0 dataset. By delving into the comparative analysis of four prominent deep learning models—Mask R-CNN, YOLOv8, EfficientDet-D0, and YOLACT—this research aims to offer helpful discoveries into their potential applicability and capability in resolving the crucial challenge of underwater trash detection.

In an effort to tackle the complexities of underwater object detection, a diverse range of models was opted for, each offering unique potential benefits. Mask R-CNN, renowned for its ability to handle intricate object shapes, emerges as a promising candidate for identifying irregular trash items. YOLOv8's real-time processing capabilities are particularly

valuable for rapid trash detection within dynamic underwater environments. EfficientDet models are known for achieving high performance with fewer computational resources compared to other architectures. It utilises Bidirectional Feature Pyramid Networks (BiFPN) to efficiently aggregate multi-scale features, enhancing detection accuracy. YOLACT is designed for real-time instance segmentation, offering fast inference speeds suitable for applications requiring rapid analysis.

Moreover, the choice to employ these particular models is not only rooted in their popularity but also in their minimal exploration in underwater settings. Masked-RCNN has been used as a reference model due to its established accuracy in similar research, while the other models were chosen because of the limited work done in an underwater setting.

IV. Methodology

A. Dataset

In the exploration of underwater trash detection, a critical component is the availability of high-quality annotated datasets. This research leverages the TrashCan 1.0 dataset, a valuable resource containing 7,212 annotated images epicting observations of marine debris, remotely operated vehicles (ROVs), and diverse underwater flora and fauna. The TrashCan 1.0 dataset utilises instance segmentation annotations, a detailed approach where bitmaps with pixel-level masks mark objects within each image. This rich annotation format provides precise information on object location and boundaries, aiding the development of accurate detection models. The imagery within TrashCan 1.0 originates from the J-EDI dataset. Since 1982, ROVs have been capturing videos primarily within the Sea of Japan, all of which are housed within this extensive library. TrashCan 1.0 offers two versions, TrashCan-Material and TrashCan-Instance, each configured for distinct object class representation. This research focuses on the TrashCan-Instance version, capitalising on its detailed instance segmentation annotations for effective trash detection [10].

The classes in the TrashCan 1.0 dataset are divided into three main groups. The first group is the ROV, which contains a single class, rov, representing the remotely operated vehicles used to capture the underwater imagery. The second group is Aquatic Life, which includes various classes representing different types of underwater flora and fauna such as animal_fish, plant, animal_starfish, animal_eel, animal_crab, and animal_shells. The third group is Trash, comprising various classes representing different types of marine debris including trash_unknown_instance, trash_bag, trash_container, trash_can, trash_branch, trash_wreckage, trash_pipe, trash_net, trash_bottle, trash_tarp, trash_rope, trash_snack_wrapper, trash_clothing, and trash_cup.

The dataset employs instance segmentation annotations, a sophisticated approach that uses pixel-level masks to mark objects within each image. These annotations are provided in JSON format, which includes detailed information about the location and boundaries of objects. This granular annotation format is instrumental in the development of accurate detection models. However, it's important to note that some classes in the dataset are underrepresented, such as the trash_cup class which has only 59 instances while other classes like rov have a substantial number of instances (3317). This underrepresentation could impact the performance of models trained on this data, leading to biased predictions.

TABLE I
Original Class Instances

No.	Class	Total Instance
1	rov	3317
2	trash_unknown_instance	2756
3	trash_bag	908
4	animal_fish	764
5	trash_container	510
6	plant	507
7	trash_can	459
8	animal_starfish	398
9	animal_eel	343
10	trash_branch	336
11	animal_crab	309
12	animal_shells	249
13	animal_etc	235
14	trash_wreckage	165
15	trash_pipe	156
16	trash_net	127
17	trash_bottle	126
18	trash_tarp	121
19	trash_rope	117
20	trash_snack_wrapper	84
21	trash_clothing	82
22	trash_cup	59

B. Dataset Challenges

Nonetheless, it's essential to recognise the inherent difficulties linked with underwater image datasets.

1) Scarcity of Annotated datasets.: Gathering annotated datasets for underwater imagery poses a significant challenge for underwater object and trash detection. The scarcity of annotated datasets in this domain arises from the difficulties associated with capturing images in underwater environments. Exploring the deeper parts of the ocean is both expensive and technologically demanding. The aquatic setting introduces complexities such as varying light conditions, turbidity, and the need for specialised equipment, hence acquiring high-quality annotated data is a daunting task. Addressing these annotation challenges through strategies like active learning or transfer learning from well-annotated datasets can be crucial for overcoming these limitations.

2) Generalizability.: Underwater waste detection models face a hidden challenge which is the bias introduced by the origin of their training data. Imagine a model trained primarily on footage from harbors and coastal areas – it would likely excel at recognising plastic bottles and food wrappers, common debris in such environments. However, this model would struggle in deeper, low-light conditions where fishing gear or lost scientific instruments are more prevalent. The dataset might primarily represent a specific geographic region or depth range, potentially limiting the model's capability to generalise and perform effectively in unforeseen scenarios. The consequence of this bias can be inaccurate detection when deployed in new environments. Diverse and geographically distributed images become necessary to tackle the problem.

3) Low-light environment.: The prevalence of colourless images captured in low-light conditions is a major obstacle for object detection models. Many datasets, including potentially a portion of TrashCan 1.0, might contain primarily greyscale imagery. Light scattering and refraction contribute to image distortion. They significantly degrade image quality and also with increasing depth, the images become progressively blurred and eventually lose colour information altogether, transforming into a greyscale world. This poses a very complex environment for a model to be trained on or a model that was trained on clear, high-quality images for object detection.

4) Emergence of novel waste.: Static datasets like TrashCan 1.0 falter in the face of the ever-evolving nature of aquatic debris. New materials constantly find their way into our waterways, with packaging innovations, industrial discards, and even unexpected items adding to the ever-growing variety. Finally, the rapid pace of industrial change throws another wrench into the works. The "throw-away" culture ensures a constant stream of new waste types, rendering static models quickly outdated. Imagine a model trained on plastic bags struggling to identify the recent influx of single-use containers. This necessitates models that can adapt to an ever-growing variety of waste items.

5) Resemblance with marine life.: Underwater object detection systems face a tricky challenge in the deep sea which is the uncanny resemblance between plastic and marine life. A plastic bag might appear indistinguishable from a jellyfish for the model, leading to a critical misidentification. This unexpected resemblance between plastic and marine life throws a significant wrench into the efforts to automate underwater waste management by reducing the accuracy of the model significantly. It highlights the limitations of current object detection models and the need for more sophisticated approaches that can differentiate between genuine marine life and its plastic doppelgangers.

C. Custom dataset

The TrashCan 1.0 dataset, utilized for model comparison, comprises 7,212 pre-annotated images across 22 classes. Due to restricted computational capabilities and non-uniformity in the number of class instances, the number of classes was reduced to three by merging similar categories. For instance, classes like "animal_crab", "animal_fish", "animal_starfish", "animal_eel" and "animal_shells" were combined into a single class called "fish" while various types of debris such as "trash_can", "trash_branch", "trash_bag", "trash_container", "trash_wreckage", "trash_pipe", "trash_net", "trash_bottle", "trash_tarp", "trash_rope", "trash_cup", "trash_clothing" and "trash_snack_wrapper" were merged into the "trash" class. The "rov" class remained unchanged due to its significant representation with 3,317 instances. [22].

The restructuring resulted in three main groups—trash, rov, and fish—to address the challenge of class imbalance and facilitate more efficient computation while still providing a comprehensive dataset for model training and comparison. According to the dataset statistics, the "trash" class had the highest number of instances at 6,006, followed by the "rov" class with 3,317 instances, and the "fish" class with 2,805 instances.

To ensure uniformity across different models, all images were resized to a standard size of 512 x 512 pixels. The dataset was divided into a training set, validation set, and testing set in a 7:2:1 proportion. Specifically, the training dataset consisted of 5,046 images, the validation set comprised 1,443 images, and the testing set included 723 images. This setup facilitated the effective training and evaluation of models on a well-distributed and balanced dataset.

TABLE II
CUSTOM CLASS INSTANCES

No.	Class	Total Instance
1	trash	6006
2	rov	3317
3	fish	2805

D. Why these models?

The selection of deep learning models for this research on underwater trash detection is based on a clear and comprehensive rationale. The models chosen—Mask R-CNN, YOLOv8, EfficientDet D-0, and YOLACT—were selected due to their proven effectiveness in object detection and relevance to challenging underwater environments.

Mask R-CNN was chosen as the reference model because the majority of research papers reviewed had used it and reported good results, making it a robust benchmark for this study. Its widespread use and established performance in instance segmentation further justify its inclusion. Mask R-CNN's architecture includes a Region Proposal Network (RPN) for generating candidate object proposals and an ROI (Region of Interest) Align layer for precise spatial alignment, leading to highly accurate object detection and segmentation. The model's ability to handle multiple objects in complex scenes with high precision makes it particularly suitable for underwater trash detection, where the background can be cluttered and objects of interest can be overlapping or partially occluded.

YOLO models, such as YOLOv4 and YOLOv5, have been extensively utilized in older research papers focused on underwater object detection. To evaluate the impact of the latest advancements in the YOLO family, YOLOv8 was included in this study. YOLOv8 represents state-of-the-art real-time object detection capabilities, making it a suitable candidate for comparison. YOLOv8's improved architecture enhances its detection accuracy and speed, making it well-suited for applications requiring rapid processing and immediate results.

EfficientDet D-0 was selected for its balance between accuracy and computational efficiency. This model, part of the EfficientDet family, is designed to achieve high performance with optimized computational resources, making it ideal for scenarios where both accuracy and speed are critical. EfficientDet employs a compound scaling method that uniformly scales the resolution, depth, and width of the network, ensuring efficient use of computational resources while maintaining high accuracy. Despite its common usage in general object detection tasks, EfficientDet has not been extensively explored in underwater environments, which adds a novel aspect to this research. The architecture of EfficientDet includes a BiFPN (Bi-directional Feature Pyramid Network), which allows for efficient and scalable feature fusion, enhancing the model's ability to detect objects at different scales.

YOLACT, while also commonly used in general object detection, has not been widely studied in underwater scenarios. YOLACT was chosen for its ability to perform fast instance segmentation. It is designed to offer a good balance between speed and detailed detection, making it suitable for real-time applications. YOLACT's architecture integrates a real-time instance segmentation approach, where the model predicts masks for each object in parallel with the bounding box detection, significantly speeding up the process without sacrificing much accuracy. This model's ability to generate instance masks rapidly makes it an interesting candidate for tasks requiring real-time performance in dynamic underwater environments.

E. Models Insights

1) Mask R-CNN: Mask R-CNN, specialises in providing image segmentation at the pixel level, delineating every object instance in the image. [21] Mask R-CNN simplifies the object detection process by dividing the image into regions of interest and simultaneously analysing each section. It utilises a two-phase structure, initially creating region proposals through a Region Proposal Network (RPN) and then refining these proposals to produce precise object masks and bounding boxes. [21] Mask R-CNN utilises a ResNet-101 deep convolutional neural network as its backbone, extracting features from the input image. These features are subsequently inputted into two parallel branches, one for bounding box regression and object classification, and the other for generating instance masks. Mask R-CNN introduces a third branch dedicated to generating instance masks. This branch employs a pixel-wise segmentation network to generate masks corresponding to each detected object, providing precise segmentation of object instances within the image. Mask R-CNN employs a multi-task loss mechanism during training, integrating losses for bounding box regression, object categorisation, and mask anticipation.

Smooth L1 Loss:

$$L_{\text{smooth}}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1, \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{1}$$

Cross-Entropy Loss :

$$L_{\text{CE}}(p, q) = -\sum_i p_i \log(q_i) \tag{2}$$

Binary Cross-Entropy Loss :

$$L_{\text{BCE}}(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{3}$$

The total loss during training is the sum of these individual losses, each weighted by specific parameters to ensure comprehensive learning.

2) YOLOv8: YOLOv8, standing for "You Only Look Once version 8". YOLOv8 simplifies object detection by dividing the image and analysing each section simultaneously. It starts by dividing the image into a grid. Then it analyses each grid section looking for objects. If an object is found it assigns a class to it and estimates the size and location of the object by drawing a bounding box around it. Then combine the information to understand the entire scene. During the whole process, it only needs a single pass through the image hence the name "You Only Look Once". YOLOv8 utilises the CspDarkNet network as its backbone. The Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) are employed in the neck section. FPN+PAN better aggregates features from different levels of the backbone. This is crucial for detecting objects of different sizes within an image.
The Classification Head predicts the probability of each detected object belonging to different classes using a mathematical function called a sigmoid activation. This function assigns a score between 0 and 1 to each class prediction.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

The Box Prediction Head determines the precise location of objects in the image. It achieves this by adjusting predefined boxes, called anchor boxes, to fit around the detected objects accurately. In YOLOv8, the model directly focuses on whether an object exists within a box rather than predicting its likelihood. Regarding error correction, YOLOv8 employs distinct loss functions. For class predictions, it uses Binary Cross Entropy Loss (BCELoss), which measures the discrepancy between predicted and actual class probabilities. For localisation, it utilises two loss functions, Distribution Focal Loss(DFL), which helps address the class imbalance by focusing more on difficult examples, and Complete Intersection over Union Loss(CIoU Loss), which enhances accuracy in box predictions. Total Loss(T) is given by :

$$\text{T} = \lambda_{\text{cls}} * \text{BCELoss} + \lambda_{\text{box}} * \text{DFL} + \lambda_{\text{iou}} * \text{CIoU Loss} \tag{5}$$

During model training, YOLOv8 combines these loss functions, weighted by specific parameters, to ensure comprehensive learning. By balancing the model's capacity to categorise objects and precisely determine their locations, YOLOv8 aims to improve its object detection capabilities.

3) EfficientDet-D0: Developed by Google in 2020, EfficientDet is an object detection model designed to achieve high accuracy while maintaining efficiency. It accomplishes this through a compound scaling method that optimises three aspects of the model simultaneously: depth, width, and resolution. This allows EfficientDet to find a sweet spot between precise object detection and the computational resources required to run the model.

a) Efficient Feature Extraction with EfficientNet Backbone.: EfficientDet builds upon the foundation of EfficientNet, a family of convolutional neural networks known for their efficiency in feature extraction. This combination enables EfficientDet to extract informative features from images effectively while remaining computationally lightweight.

b) BiFPN a Novel Approach to Multi-Scale Feature Fusion.: One of the key differentiators of EfficientDet is its use of BiFPN (Bi-directional Feature Pyramid Network) for combining features extracted at different scales within the network. Traditionally, object detection models like YOLO and SSD rely on FPNs (Feature Pyramid Networks) for this task. EfficientDet takes a more sophisticated approach with BiFPN. This network introduces a more efficient and lightweight method for multi-scale feature fusion.

c) Bi-directional Information Flow.: Unlike FPNs with their primarily top-down pathway, BiFPN incorporates both top-down and bottom-up pathways. This allows information to flow in both directions, enabling features from different levels to interact more effectively. This interaction potentially leads to richer and more informative representations for object detection. [19]

d) Adaptive Feature Weights.: Prior to EfficientDet, most multi-scale feature fusion approaches treated all input features equally, regardless of their resolution. However, features at different resolutions contribute unequally to the final output. To address this, EfficientDet introduces adaptive weights for each input feature. The network learns the importance of each feature during training, allowing it to focus on the most informative ones. [19]

e) Fast Normalised Fusion and Compound Scaling.: The final BiFPN in EfficientDet integrates both the bi-directional flow of information and a fast normalised fusion technique. This combination allows for efficient and effective feature representation. [19]
Here is an example of level 6 of the BiFPN with fused features.

$$P_6^{td} = \text{Conv}\left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot \text{Resize}\left(P_7^{in}\right)}{w_1 + w_2 + \epsilon}\right) \tag{6}$$

$$P_6^{out} = \text{Conv}\left(\frac{w_1' \cdot P_6^{in} + w_2' \cdot P_6^{td} + w_3' \cdot \text{Resize}\left(P_5^{\text{out}}\right)}{w_1' + w_2' + w_3' + \epsilon}\right) \tag{7}$$

Another key aspect of EfficientDet is its use of compound scaling. Unlike other models that simply scale up the entire baseline detector, EfficientDet utilises a compound scaling approach. This approach focuses on scaling up all dimensions of the network simultaneously, including the backbone network, the BiFPN network, and the input image resolution. This ensures that all parts of the model benefit from the scaling process, leading to improved accuracy without sacrificing efficiency disproportionately. Finally, EfficientDet leverages the same width/depth scaling coefficients used in EfficientNet models (B0-B6) [20] for its backbone, ranging from D0 to D7. This established approach provides a solid foundation for scaling the model effectively.

4) YOLACT: YOLACT, which stands for 'You Only Learn to Aggregate and Convolutions Together', is selected for its unique approach to instance segmentation. Unlike traditional methods, YOLACT employs a fully convolutional model, providing efficient and accurate instance segmentation predictions. This model excels in capturing fine-grained details, making it well-suited for identifying diverse underwater pollution instances. The innovative approach of YOLACT in aggregating and convolving features make it a compelling choice for evaluating instance segmentation performance in the context of underwater trash detection. YOLACT is a fully convolutional model which is developed for real-time instance segmentation. Instance segmentation is an approach in object detection where the object is detected using a mask. YOLACT adds very little overhead over the one-stage detector providing a fast approach for instance segmentation [9]. YOLACT produces the mask for the object in two stages which work in parallel in the first stage it generates prototype masks that are not particular to any of the classes and is equal to the size of the image using an FCN [protonet FPN(Feature Pyramid Network) or ReLU for more interpretable prototypes] [9].In the second stage in addition to the two branches one for predicting class and another for prediction bounding box, it adds the third layer for predicting the mask coefficients. [9] To combine the output of these two stages they used sigmoid nonlinearity to generate the final mask.

$$M = \sigma\left(PC^T\right) \tag{8}$$

Where P is the prototype mask h * w * k matrix where h is height, w is width, k is the number of mask coefficients of mask prototype and C is n * k matrix where n is the number of instances that survived after a different threshold and k is the number of mask coefficient [9]. With this methodology, YOLACT has achieved superior speed performance compared to the leading instance segmentation models. It managed to deliver results that were 3.9 times quicker than the fastest previous method while maintaining a competitive mask mAP.

V. Training

To ensure a thorough and impartial comparison of these models, all training parameters that could potentially impact the results were meticulously controlled. This comprehensive

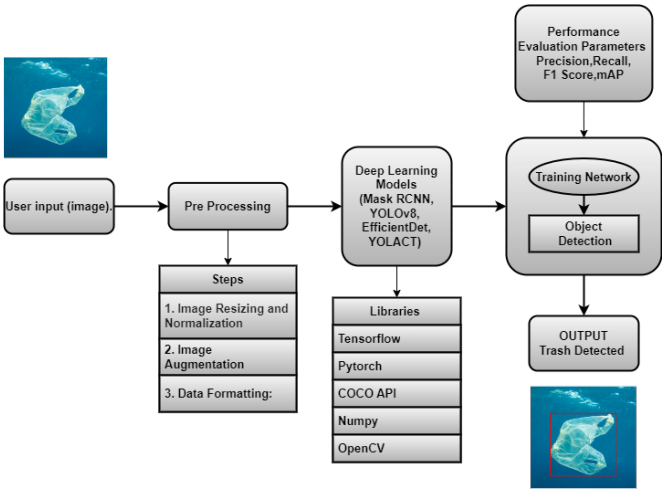

Fig. 1. System Architecture

standardisation was implemented to guarantee that any observed performance differences were solely attributed to the inherent characteristics of the models, rather than variations in the training environment or configuration. The training process for all models followed the same split of training, testing, and validation datasets as outlined in the dataset section. Each model underwent 80 epochs of training with a batch size of 8, totaling 50,000 iterations, on a system with identical configuration. Training took place in a conda environment within a Kaggle notebook, utilising the Tesla P100 GPU as an accelerator. It utilised pre-trained weights on a model previously trained on the COCO dataset 2017. Given the unavailability of a substantial amount of data for training from scratch, all models adopted a transfer learning approach, leveraging different pre-trained weights. This approach ensured the efficient training of the models and facilitated a fair comparison of their performances. The system architecture is explained in Fig. 1.

1) Mask R-CNN.: This Mask R-CNN model leverages a pre-trained ResNet-101 backbone, which provides a strong foundation for object recognition. The model utilises data stored in the COCO JSON format for the training process [21]. This format offers an efficient way to manage and access labelled data for object detection and segmentation tasks. The extended training time of Mask R-CNN as discussed later reflects the model's complexity and the intricate task of not only detecting objects but also generating precise masks to delineate their boundaries.

2) YOLOv8.: This model was trained using the CSP-DarkNet53 backbone. The model uses data in the UltraLytics JSON format for training.

3) EfficientDet D-0.: This model was developed using the EfficientNet-B0 as the underlying architecture. It leveraged the pre-existing weights of EfficientNet. The model was trained using data in the TfRecords format.

4) YOLACT.: This model was trained using the resnet-50 as the underlying architecture. The model required a dataset in COCO JSON format for its training.

VI. Result

A. Evaluation Metrics

1) Precision(P): Precision is a measure used to assess the exactness of the detection made by an object detection model. It evaluates the ratio of predicted objects that correspond to real objects in the image. A high precision value indicates that the model is producing a high number of correct detection with minimal false positives. Conversely, a low precision suggests the model makes many mistakes by identifying non-existent objects.

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \tag{9}$$

2) Recall: It quantifies the fraction of actual objects that are accurately identified by the object detection model. High recall indicates that the model recognizes most of the relevant objects, while low recall indicates that the model misses most of the relevant items.

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \tag{10}$$

3) F1 Score: The F1 score is a measure that assesses the comprehensive performance of an object detection model by harmonising precision and recall. It considers both the model's ability to detect all actual objects (comprehensiveness) and the precision of its detection. A high F1 score indicates a good balance between these two aspects, while a lower score suggests the model might be struggling in one or both areas.

$$F1Score = \frac{2 * precision * recall}{precision + recall} \tag{11}$$

4) Intersection over Union (IoU): IoU is a metric used to measure the overlap between a predicted bounding box and the actual bounding box (or mask) for an object. It essentially quantifies the accuracy of the model's localisation for a detected object. A higher IoU value indicates a greater degree of overlap and, consequently, better localisation accuracy.

5) mAP: Mean Average Precision (mAP) is an important metric used to evaluate the overall performance of the object detection model. It takes into account the performance of the model at different thresholds of the Intersection Over Union(IoU). IoU measures the overlap between an object's estimated bounding box and its ground truth value. Calculating mAP provides insight into the consistency of the model. A higher mAP value indicates a model that excels across varying levels of strictness in bounding box overlap, while a lower mAP suggests the model's performance might be more sensitive to the chosen IoU threshold. mAP is an important metric for comparing the performance of different detection systems.

VII. Qualitative Analysis.

Qualitative Analysis in the case of model evaluation involves examining the outputs and behaviour of machine learning models beyond just numerical metrics. It provides deeper insights into the inner workings of the model and its implications for real-world applications. To do qualitative analysis the models were tested on the TrashCan 1.0 dataset itself as well as on unseen images that were scrapped from the internet. By analysing the below images, one can observe the model's proficiency in accurately detecting trash.For producing comparative results, the threshold value of the confidence score was set to 0.5 for each model.

In Fig. 2 for the first test image all models successfully detected the ROV, likely due to its large size and clear visibility. However, their performance differed when dealing with smaller, less visible objects. The image shows various types of trash, including plastic bags, food containers, and a metal can. These items are partially submerged, shadowed, or appear in murky water, making them difficult to detect. While YOLOv8 and YOLACT misses most of this trash, other models manage to pick up at least one, demonstrating better sensitivity towards smaller or less obscured items. For the second test image, there is a cluster of plastic bottles. These are transparent or partially submerged, adding to the detection challenge. Here, Mask-RCNN stands out by successfully identifying the plastic bottles. This suggests Mask-RCNN might have an advantage in detecting objects with specific shapes or textures. Moving to the last image, YOLOv8 identified most of the visible trash. However, its performance suffered when dealing with overlapping objects, like a pile of plastic bags. Here, Mask R-CNN accurately detected the overlapping trash. Interestingly, YOLOv8 and EfficientDet misclassified certain trash items, particularly those resembling fish. This highlights the ongoing challenge of differentiating trash from aquatic life.

A. Quantitative Analysis.

To objectively assess the performance of the trained Mask R-CNN, YOLOv8, EfficientDet-D0, and YOLACT a quantitative analysis was conducted using established metrics commonly employed in object detection tasks.

B. Loss Graph Analysis.

Fig. 3 depicts the total loss curves of the trained models during the training phase. The total loss for each model is a combination of various individual loss components, with class loss and bounding box loss (bbox loss) being common to all the models. Mask R-CNN and YOLACT, both models exhibit a sharp decrease in total loss during the initial epochs, indicating rapid learning of basic patterns in the training data. Subsequently, the reduction in loss becomes more gradual, reaching a stable state as training progresses. In contrast to Mask R-CNN and YOLACT, YOLOv8's loss curve shows a gradual decrease in total loss over the first 10 epochs. This is followed by a sustained but steady reduction in loss, indicating continued refinement throughout training. Similar to YOLOv8, EfficientDet experiences a substantial drop in the total loss during the initial epochs, suggesting efficient learning in the early stages. However, its loss reduction follows a trend more akin to Mask R-CNN and YOLACT, stabilising after 40-50 epochs. After approximately 40-50 epochs, Mask R-CNN, EfficientDet, and YOLACT demonstrate similar trends in their loss curves. However, YOLOv8 maintains a gradual decrease in loss throughout its training period. In summary, each model displays distinct learning patterns. Mask R-CNN and YOLACT show a more pronounced initial learning phase, while YOLOv8 exhibits a gradual and consistent refinement. EfficientDet initially learns efficiently but eventually stabilises, similar to Mask R-CNN and YOLACT.

1) Training Time Analysis.: Real-world deployments of underwater trash detection systems often require adaptability to changing environments or object types. In such scenarios, the ability to quickly re-train the model becomes crucial. Table III highlights a significant advantage for YOLOv8 in terms of training time. YOLOv8 requires only 2.79 hours to train, while Mask R-CNN takes a considerably longer 26.80 hours. This translates to a nearly tenfold difference in training speed. EfficientDet-D0 and YOLACT fall between these two extremes, requiring 6.33 hours and 6.50 hours for training, respectively.

This substantial difference in training times has practical implications. Faster training times, as exhibited by YOLOv8, allow for quicker model updates and adaptation to changing underwater environments. This is particularly beneficial for real-world deployments where environmental conditions or the types of trash objects might evolve over time. Additionally, faster training times can potentially lead to lower computational costs for training.

2) Comparison of Performance Metrices.: The performance comparison between the models Mask R-CNN, YOLOv8, EfficientDet, and YOLACT used for underwater object detection is summarised in Table III.To comprehensively evaluate the object detection models' performance for underwater trash detection, established metrics (mAP, Recall, and F1 Score) were employed across a confidence threshold (0.5: 0.95). These values enable analysis of the models' ability to detect trash objects within the threshold confidence levels. A higher confidence threshold indicates a stronger belief in the model's accuracy. By analysing these metrics at this threshold, a nuanced understanding of the model's strengths and weaknesses is gained.

TABLE III
Performance Comparison.

Model	mAP (0.5-0.95)	Recall (0.5-0.95)	F1 Score	Training Time (Hrs)
Mask R-CNN	0.627	0.703	0.663	**26.80**
YOLOv8	**0.714**	**0.868**	**0.783**	**2.79**
EfficientDet-D0	0.560	0.677	0.613	6.33
YOLACT	0.542	0.615	0.576	6.50

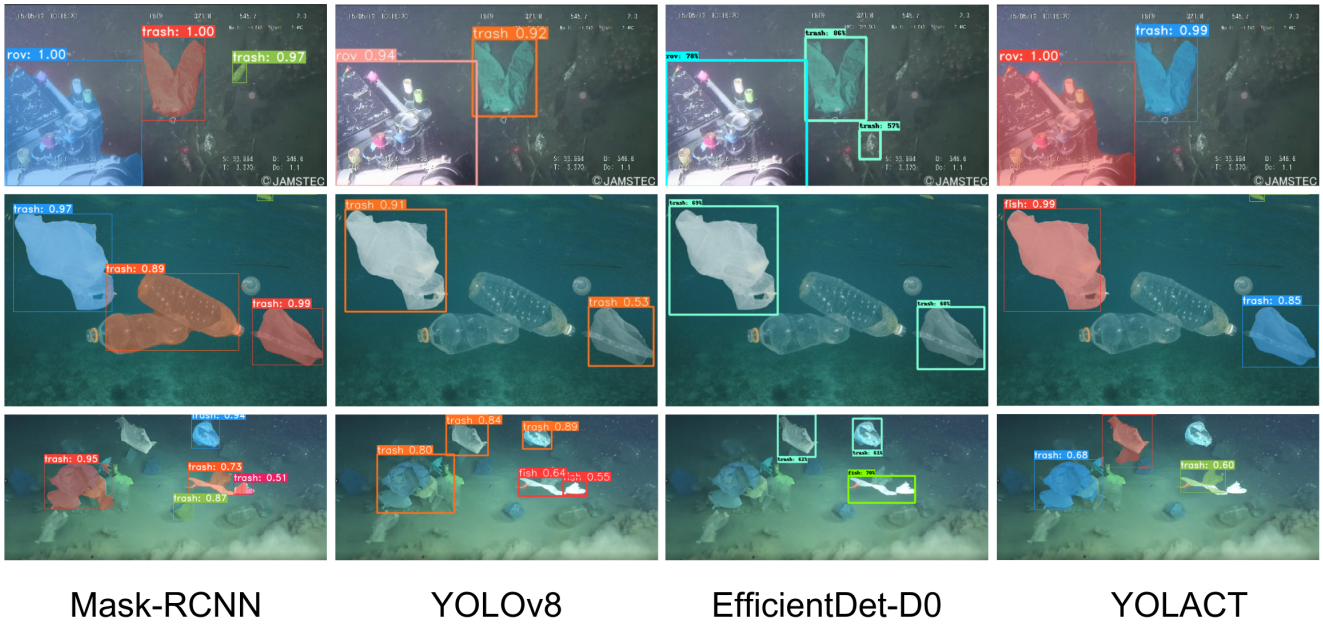

Fig. 2. Prediction of the models on unseen data.

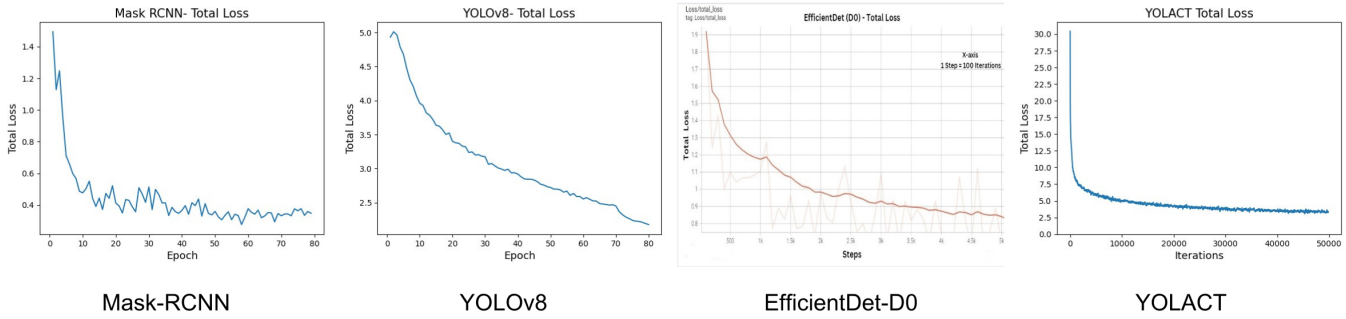

Fig. 3. Total loss curves of the trained models.

YOLOv8 achieved the highest mAP value of 0.714, followed by Mask R-CNN with 0.627, EfficientDet-D0 with 0.560, and YOLACT with 0.542. This suggests that YOLOv8 generally outperformed the other models in terms of overall detection accuracy. However, focusing solely on mAP might not provide a complete picture. YOLOv8 maintains the lead in Recall (0.868), indicating its effectiveness in finding a high proportion of trash objects. YOLOv8 also achieves a higher F1 Score (0.783) compared to other models. F1 Score balances precision and recall, suggesting that YOLOv8 strikes a good balance between finding most of the trash objects (high Recall) and ensuring those detections are accurate (high precision).For underwater trash detection, both minimising missed trash and ensuring accurate detections are crucial. YOLOv8's performance in both metrics suggests its potential as a strong candidate for this application. YOLOv8 provide a more balanced performance while also maintaining a relatively faster training time (2.79 hours). This makes YOLOv8 a compelling choice for real-world deployments where accuracy and efficiency are equally important.

Here it can be observed that while the performance of Mask R-CNN, EfficientDet, and YOLACT were close to each other, YOLOv8 exceptionally outperformed all these models in terms of prediction, accuracy and speed of training.

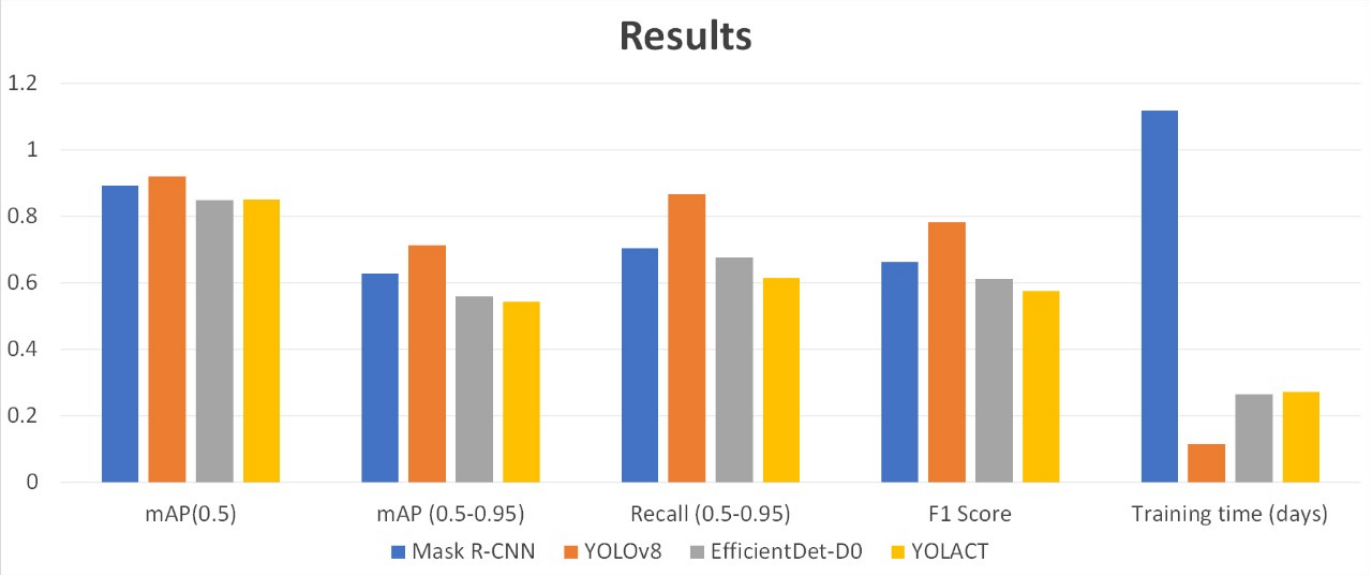

Fig. 4. Result graph of the trained models.

VIII. Conclusion

This research paper has explored the capabilities of various deep learning models for underwater object and trash detection, aiming to address the critical environmental challenge of underwater pollution. The models which are utilised for this study are Mask R-CNN(reference model), YOLOv8, EfficientDet-D0, and YOLACT. This research shows that YOLOv8 emerges as the top performer in the evaluation, achieving the highest mAP(0.5:0.95) value of 71.4%, while also maintaining a higher F1 Score of 78.3% compared to other models. This balanced performance, coupled with a faster training time of 2.79 hours, positions YOLOv8 as a promising choice for real-world underwater trash detection applications. In specific applications, such as litter collection using Autonomous Underwater Vehicles (AUVs), YOLOv8's real-time capabilities make it a practical choice. Its superior performance metrics and faster training times contribute to efficient litter identification and collection in dynamic underwater environments. Conversely, Mask R-CNN proves to be a superior option for monitoring litter distribution due to its ability to detect smaller-sized trash and identify waste precisely in complex underwater scenes. However, Mask R-CNN requires a considerable time to process images which makes it unsuitable for real-time trash collection. The detailed analysis provided by Mask R-CNN could be utilised to train YOLOv8, thereby enhancing the performance of the existing model. As the research utilised the base versions of YOLACT and EfficientDet, there exists potential for further improvement in their performance by opting for their higher versions. For example, upgrading to YOLACT++ or using a higher version of EfficientDet, like EfficientDet-D2 or higher, may yield better performance. However, it's essential to take into account the balance between precision and computational resources, as higher versions tend to be more computationally heavy. In conclusion, this research advances the discipline of underwater object and trash detection by evaluating and comparing various deep-learning models. By understanding the advantages and constraints of each model, it can tailor their usage to specific applications, ultimately aiding in the sustainable management of aquatic environments and the preservation of marine ecosystems.

IX. Scope for further Research

Based on the results of our object detection project, we have identified several promising areas for future work aimed at enhancing the efficiency, accuracy, and reliability of our models. These improvements are essential for advancing the applicability and performance of our models in various real-world scenarios.

To increase efficiency, we plan to implement advanced model optimization techniques, such as model pruning and quantization. Pruning will involve reducing the number of less important weights or entire filters in our neural networks, thereby decreasing the size of the models without significantly affecting their performance. Quantization will convert model weights from 32-bit floating-point to 8-bit integers, which will drastically reduce memory usage and increase inference speed. Another strategy will involve experimenting with different backbones for our models. By selecting more efficient backbone architectures, we can reduce the computational complexity and improve the inference speed without sacrificing too much accuracy. Furthermore, we plan to explore newer versions of the YOLO family of models, which promise improved efficiency and accuracy over previous versions. These models are designed to balance speed and performance more effectively, making them suitable candidates for enhancing our current setup.

In order to boost the accuracy of our models, we will employ sophisticated data augmentation techniques, such as CutMix, MixUp, and mosaic augmentation. These methods create more diverse training samples, enhancing the generalisation capabilities of our models. Additionally, we will explore ensemble learning by combining predictions from EfficientDet, YOLOv8, and YOLACT. This approach leverages the strengths of each model, potentially leading to improved overall performance by averaging their predictions or using voting mechanisms to determine the final output. We will also undertake hyperparameter optimization using Bayesian techniques to find the optimal set of parameters for our models. This process will involve fine-tuning key hyperparameters, including learning rates, batch sizes, augmentation parameters, and network architecture settings. Furthermore, we aim to incorporate newer versions of the YOLO models, which have shown significant improvements in accuracy in recent benchmarks, into our workflow. These models, with their state-of-the-art performance, will be crucial in pushing the accuracy of our object detection systems to higher levels.

Ensuring the reliability of our models is crucial for their deployment in real-world applications. We will focus on training our models on diverse datasets to improve their ability to generalize across different domains and conditions. This approach will enhance the models' robustness to variations in input data. To make our models resilient against adversarial attacks, we will employ adversarial training techniques, which involve training the models with adversarial examples designed to fool them. This will improve the models' security and reliability in scenarios where they might encounter intentional perturbations. Additionally, we will integrate methods for uncertainty estimation, such as Monte Carlo Dropout or Bayesian Neural Networks, which allow our models to quantify their confidence in predictions. This can help in identifying uncertain predictions that may require further verification, thus improving the overall reliability of the system. Consistent application of regularization techniques like dropout, weight decay, and batch normalization will also help prevent overfitting, ensuring that the models maintain good performance on unseen data and remain stable across different datasets. This will involve regularly retraining the models with updated data and validating their performance, which will help maintain their accuracy and reliability over time. By addressing these strategies in detail, we aim to provide a comprehensive plan for future improvements, thereby enhancing the scope and impact of our object detection models in practical applications.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to Dr. Sarika Zaware, our esteemed project mentor and the Head of the Department of Computer Engineering at AISSMS Institute of Information Technology. Her unwavering support, guidance, and mentorship have been vital in shaping the direction of this research project. Dr. Zaware's wealth of knowledge, commitment to excellence, and dedication to fostering a conducive research environment have significantly contributed to the successful completion of this endeavor. We would also like to extend our gratitude to the faculty and staff of the Computer Engineering Department at AISSMS Institute of Information Technology for their valuable insights, encouragement, and collaborative spirit. The conducive academic atmosphere provided by the department has played a crucial role in nurturing our research aspirations. Furthermore, our sincere thanks go to the administration and technical support teams at AISSMS Institute of Information Technology for their seamless assistance, ensuring that we had the necessary resources and infrastructure for the smooth progression of our research. Finally, we acknowledge the continuous encouragement and understanding from our friends and family, whose support has been a source of inspiration throughout the research journey. Their unwavering belief in our capabilities has been a driving force, and we are deeply grateful for their enduring support.

REFERENCES

[1] Wang, H., Xiao, N.: Underwater Object Detection Method Based on Improved Faster RCNN. Appl. Sci. **13**, 2746 (2023). https://doi.org/10.3390/app13042746

[2] Ahmad Salman, Shoaib Ahmad Siddiqui, Faisal Shafait, Ajmal Mian, Mark R. Shortis, Khawar Khurshid, Adrian Ulges, Ulrich Schwanecke: Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES Journal of Marine Science **77**(4), 1295–1307 (2020). http://dx.doi.org/10.1093/icesjms/fsz025

[3] B Vikram Deep, Ratnakar Dash: Underwater Fish Species Recognition using Deep Learning Techniques. In: 6th International Conference on Signal Processing and Integrated Networks (SPIN) (2019).

[4] Hu, X., Liu, Y., Zhao, Z., Liu, J., Yang, X., Sun, C., Chen, S., Li, B., Zhou, C.: Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network. Computers and Electronics in Agriculture **185**, 106135 (2021). https://doi.org/10.1016/j.compag.2021.106135

[5] Wu, C.-H., Hsieh, J.-W., Wang, C.-Y., Ho, C.-H.: Marine Pollution Detection based on Deep Learning and Optical Flow. In: 2020 International Computer Symposium (ICS) (2020). https://doi.org/10.1109/ics51289.2020.00081

[6] Bhanumathi M, DhanyaS, Gugan R, KirthikaKG: Marine Plastic Detection Using Deep Learning. In: Advances in Parallel Computing Algorithms, Tools and Paradigms. IOS Press (2022). https://doi.org/10.3233/apc220057

[7] Tian, M., Li, X., Kong, S. et al.: A modified YOLOv4 detection method for a vision-based underwater garbage cleaning robot. Front Inform Technol Electron Eng **23**, 1217–1228 (2022). https://doi.org/10.1631/FITEE.2100473

[8] Moniruzzaman, M., Islam, S. M. S., Lavery, P., Bennamoun, M.: Faster RCNN Based Deep Learning for Seagrass Detection from Underwater Digital Images. In: 2019 Digital Image Computing: Techniques and Applications (DICTA) (2019). https://doi.org/10.1109/dicta47822.2019.8946048

[9] Daniel Bolya, Chong Zhou, Fanyi Xiao, Yong Jae Lee :YOLACT: Real-time Instance Segmentation. https://doi.org/10.48550/arXiv.1904.02689

[10] University of Minnesota Libraries. (2022). Open Access Fundamentals [Data set]. University of Minnesota Conservancy. Trashcan 1.0 official site. https://conservancy.umn.edu/handle/11299/214865

[11] Matterport: Mask R-CNN. GitHub Repository. https://github.com/matterport/Mask_RCNN

[12] TensorFlow: Object Detection with TensorFlow Lite. Official Documentation. https://www.tensorflow.org/lite/examples/object_detection/overview

[13] PyTorch: Torchvision Tutorial. Official Documentation. https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

[14] Ultralytics Documentation. Official Documentation. https://docs.ultralytics.com/

[15] Ultralytics GitHub Repository. Official GitHub Repository. https://github.com/ultralytics/ultralytics

[16] YOLACT GitHub Repository. Official GitHub Repository. https://github.com/dbolya/yolact
[17] EfficientDet Github Repository. Official Github Repository. https://github.com/xuannianz/EfficientDet
[18] TensorFlow TPU: Official Models - EfficientNet. GitHub Repository. https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet
[19] Mingxing Tan, Ruoming Pang, Quoc V. Le. *EfficientDet: Scalable and Efficient Object Detection*. arXiv preprint arXiv:1911.09070 (2020). https://arxiv.org/abs/1911.09070
[20] Mingxing Tan, Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. https://arxiv.org/abs/1905.11946
[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. *Mask R-CNN*. arXiv preprint arXiv:1703.06870 (2017). https://arxiv.org/abs/1703.06870
[22] Roboflow. https://roboflow.com/
[23] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun https://doi.org/10.48550/arXiv.1506.01497