Architectural Optimization and Comparative Benchmarking of Attention-Augmented YOLO Models for Deep-Sea Litter Classification

Tharun A, Sai Govardhan M Jayachetan, Rimjhim Singh
Department of Computer Science and Engineering
Amrita School of Computing,Bengaluru
Amrita Vishwa Vidyapeetham,India.

bl.en.u4cse22173@bl.students.amrita.edu, bl.en.u4cse22170@bl.students.amrita.edu ,
bl.en.u4cse22144@bl.students.amrita.edu, ps_rimjhim@blr.amrita.edu

***Abstract*—Marine waste, especially of anthropogenic origin like plastic, metal, glass, and fabric, is an increasingly serious threat to aquatic life. Effective, real-time identification and labeling of such waste is essential in aiding cleanup efforts and monitoring of the environment. This project investigates an end-to-end computer vision workflow involving the application of several YOLO (You Only Look Once) object detection models to detect marine waste in images and video streams. A total of 17 YOLO-based models were comprehensively analyzed, including baseline versions and attention-supplied models with hybrid backbones (e.g., ResNet50, Vision Transformer). The models were tested on a bespoke Roboflow dataset consisting of four waste types and were assessed on precision, recall, and mAP scores. The last system provides a scalable and precise solution for real-time environmental waste monitoring in marine applications.**

***Index Terms*—Marine Waste Detection,Object Detection,YOLOv5–YOLOv12,Attention Mechanisms,Vision Transformer (ViT),ResNet Backbone,Deep Learning,Environmental Monitoring,Roboflow Dataset**

I. Introduction

Marine pollution has become a serious environmental concern with enormous amounts of waste piling up in the oceans, rivers, and coastal areas. Of the numerous pollutants, anthropogenic waste like plastic, metal, glass, and cloth ranks high among the dangers threatening marine biodiversity as well as human health. Many of the conventional waste detection and cleanup activities depend on human observation or sensors at the surface, which are ineffective and error-prone. To meet this challenge, there is a pressing requirement for intelligent, automated systems to identify and categorize marine waste in real-time.

Advances in computer vision and deep learning have made object detection in complex environments possible with automated means. Specifically, YOLO models have become well-known for their compromise between speed and precision in real-time processing. YOLO object detectors perform a single forward pass on images, making them appealing for dynamic environments like aerial or underwater surveillance.

This project centers on the design and assessment of a set of YOLO-based models for detecting marine waste in four prominent categories—cloth, glass, metal, and plastic. The models are trained on a specially designed dataset amassed and annotated through Roboflow, customized for real-world waste detection. In addition to baseline detection through typical YOLO models (YOLOv5 through YOLOv12), this project explores architectural augmentations with attention layers, and ResNet backbones to better detect small, occluded, or visually intricate waste items.

With extensive experimentation across 17 YOLO variants, such as attention-augmented, the research illustrates the feasibility of smart waste monitoring systems. Results aim to inform scalable, automated marine cleanup, underwater robotics, and enforcement of environmental policies.

The primary contributions of this work include:

- Trained and tested 17 models of YOLO (YOLOv5 to YOLOv12) variants with and without attention and hybrid.
- Implemented attention layers at multiple depths and skip connections to improve feature concentration.
- Designed a YOLO-based real-time system to identify cloth, glass, metal, and plastic in marine settings, including underwater and above-water scenes.

II. Literature review

Saeed et al. [1] proposed a deep learning model based on the YOLOv5 framework for detecting and classifying marine litter, achieving high accuracy in object detection with a robust segmentation approach. The method demonstrated significant potential for real-time applications and scalability in marine waste monitoring. However, the model's performance was constrained by the limited diversity of the dataset used, affecting its generalizability to other environments.

Liu et al. [2] applied a semantic segmentation model based on a U-Net architecture for the identification and classification of floating waste in ocean environments. It was capable of producing fine-grained detection with high precision for segmenting marine litter. A significant limitation, however,

was the computational overhead that prevented it from being deployed on resource-constrained devices.

Yu et al. [3] presented a hybrid model by incorporating traditional machine learning with CNN-based feature extraction to classify marine litter from aerial images. This integration enhanced the interpretability of the machine learning method while reducing the computational costs. The method was weak in the detection of small objects in large-scale images.

Islam et al. [4] applied a Faster R-CNN framework for underwater marine waste detection. The proposed approach demonstrated robust performance in complex and occluded scenarios. Its advantage was the adaptability to diverse underwater conditions due to advanced feature extraction techniques. However, the model's performance degraded in low-visibility conditions, highlighting the need for additional preprocessing steps.

Fulton et al. [5] proposed a deep visual detection system using RetinaNet for detecting ocean waste on robotic platforms. The method excelled in real-world deployments and was perfectly suitable to be applied autonomously, but it relied heavily on copious amounts of annotated data for training, which became the bottleneck for further scaling.

Kyriaki et al. [6] made use of a deep learning pipeline with the ResNet50 architecture to detect floating plastic debris in oceanic regions. The method presented high accuracy in distinguishing plastics from organic matter, thus proving to be effective in targeted waste management. Its ability to carry out submerged waste detection was lacking, however, making it partially insensitive to holistic waste detection.

A DCNN by Rajasekaran et al. [7] is presented that was used for the classification of marine debris while optimizing the model for real-time energy efficiency in applications. Such optimization allowed its deployment on low-power edge devices. However, the model suffered under light variability.

Singh et al. [8] designed a hybrid visiontransformer-based model for the high-precision segmentation of marine wastes in complex oceanic environments. The method was highly efficient and resulted in improved performance compared to CNNs in precision and recall but had a large computational requirement and was not recommended for real-time low-power applications.

Sharma et al. [9] proposed a multi-scale deep learning model for the detection and classification of marine litter in drone-captured images with outstanding performances for small and overlapping objects detections. The model was proven to be really scalable with vast applications; however, it is dependent on high-resolution imagery, which leads to limited usability in resource-constrained environments.

Wang et al. [10] proposed an ensemble learning method by combining CNN and transformer models to classify recyclable and non-recyclable marine waste. The ensemble method improved the detection accuracy and robustness under different ocean conditions. The method, however, demanded large computational resources and longer training times.

Zhang et al. [11] carried out classification of marine litter using a hybrid CNN-LSTM architecture to analyze patterns in the distribution of oceanic waste sequentially. There is a huge drawback, which is higher complexity. It made the system difficult to implement in real-time.

Patel et al. [12] proposed an attention-based segmentation network to improve the detection of marine debris in high-clutter environments. The attention mechanism helped enhance model performance by focusing on the relevant regions of the images. However, noisy data sensitivity limits the robustness of the network.

Ali et al. [13] used a generative adversarial network (GAN) to improve the performance of downstream detection models of marine waste. The augmented dataset helped the model generalize better under various scenarios; however, in the GAN-based approach, it was often sensitive to some tuning parameters because otherwise, it would give unrealistic or highly irrelevant samples.

Kumar et al. [14] proposed a lightweight MobileNet-based model for real-time marine litter detection that can be deployed on low-power devices. The model had high efficiency with low computational costs. However, it had slightly lower accuracy than heavier architectures in complex scenarios.

Joshi et al. [15] used an ensemble model, combining ResNet and DenseNet deep learning models for the classification of marine waste from underwater images. Since the ensemble leverages complementary features, detection accuracy improved. On the other hand, training was more time-consuming because of the complexity of the ensemble.

Roy et al. [16] proposed a transfer learning-based method using a pre-trained VGG16 for detecting and classifying marine debris in low-resource settings. This approach eliminated the need for a large amount of data and produced comparable results. However, its usability was restricted to very similar domains than the one from which the pre-trained model was obtained.

Chen et al. [17] proposed a region-based segmentation model for the detection of marine debris using aerial imagery, with a focus on multi-class classification of waste types. The approach provided granular insights into the composition of marine litter. A drawback was the dependence on annotated aerial images, which are costly and time-consuming to acquire.

Shubham et al. [18] proposed a YOLOv7-tiny-based framework for detecting underwater litter, enhancing the model's performance for real-time underwater applications. The lightweight architecture allowed efficient deployment on underwater robots with minimal computational resources. However, the model struggled with detecting waste in extremely low-light or murky water conditions, reducing its robustness.

Zhao et al. [19] proposed a modified CNN for underwater litter detection from multi-modal data of underwater robots. The approach improved the accuracy of detection by fusing visual and depth data and was thus very effective in complex underwater environments. However, it had the disadvantage of relying on high-quality depth data, which is difficult to obtain in most marine scenarios.

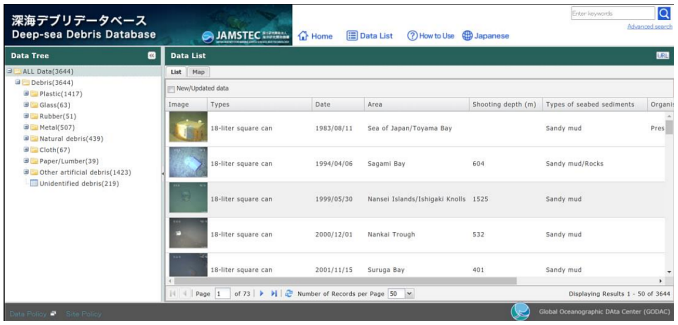


Fig. 1. JAMSTEC Database Screenshot

Shetty et al. [20] provided a machine learning pipeline for marine waste classification based on a random forest algorithm learned on spectral and texture features. This method was interpreted highly and had good accuracy in small datasets. Therefore, the usage of the pipeline was restricted mainly to small-scale studies on marine litter, rather than the field's real-world application

III. Proposed Methodology:

The methodology followed in this work includes various important stages: creation and annotation of the dataset, training baseline models, integration of attention mechanisms, backbone improvement, and performance testing. 17 YOLO-based models, including base, attention-augmented, and personalized models, were implemented and compared.

A. Dataset Collection and Annotation

Fig. 1 custom dataset was created using videos from the JAMSTEC Marine Debris Dataset, which contains rich underwater footage of marine environments affected by pollution. Keyframes and screenshots were manually extracted from these videos to capture scenes depicting various forms of waste. This ensured that the data encompassed a wide range of environmental conditions, including varying lighting, object occlusion, motion blur, and debris density. The images extracted were manually annotated on the Roboflow annotation platform with bounding boxes around marine litter objects, labeled into one of four categories: cloth, plastic, metal, and glass. The dataset was then exported in the YOLO format, organized in subdirectories for training, validation, and testing with a data.yaml file that specifies class names and paths. This structure is conducive to smooth integration into YOLO-based training pipelines and makes it easy to experiment with various model variants efficiently.

B. Baseline YOLO Model Training

Baseline training was performed across various YOLO architectures, namely YOLOv5s, YOLOv9s, YOLOv10n, YOLOv10m, YOLOv11s, YOLOv11m, YOLOv12s, and YOLOv12m. These models were used as the reference for benchmarking against improved versions. Training was carried out using the Ultralytics YOLO framework on GPU-capable environments like Google Colab and Kaggle. Uniform training hyperparameters were used across models, such as 50 training epochs, a batch size of 16, and image dimensions of either 416 or 640 pixels. This ensured consistency and comparable results across experiments.

C. Attention-Augmented YOLO Models

As an added measure to increase detection accuracy, especially for partially occluded or small objects, attention mechanisms were integrated at several depths in the YOLO network architecture. These consisted of injection sites at layers 10, 14, 17, 20, and 23. Attention was also applied in skip connections in some models to augment contextual flow between shallow and deep layers. These improvements were brought into YOLOv10 via YOLOv12 variants with tailored PyTorch modules translated into the Ultralytics source architecture. The aim of these changes was to improve feature prioritization in detection and improve spatial relationships in intricate underwater imagery.

D. Custom Backbone YOLO Models

Besides attention modules, this research delves into the application of custom CNN backbones to further improve YOLO's feature extraction. ResNet50 was utilized in certain models in place of the conventional YOLO convolutional backbone. The ResNet-based design was implemented into YOLOv10 and YOLOv11 variants by redesigning the early layers to take feature maps from the ResNet output. This modification was directed towards enhancing the spatial feature representation and visual degradation robustness in detecting visually degraded marine litter objects. The integration was validated by ablation and side-by-side model comparison.

E. Training and Evaluation Workflow

The last step in the methodology was training all model variants on the annotated dataset and assessing their performance using common object detection metrics, such as precision, recall, and mean average precision at IoU threshold 0.5 (mAP@0.5). Apart from quantitative evaluation, qualitative evaluation was carried out by visually examining the detection results and checking correctness and confidence of predicted bounding boxes. This rigorous method allowed for a comprehensive comparison between 17 model configurations and highlighted the effect of architectural improvements in marine trash detection performance.

The overall system design, depicted in Figure 2, starts with raw image inputs from the JAMSTEC debris dataset. The images are hand-annotated with Roboflow and re-encoded into a YOLO-compatible format. Once preprocessed, the data are fed to several YOLO-based models—baseline, attention-augmented, and ResNet-enhanced—subsequent to object detection. These outputs are bounding boxes and class predictions for four types of marine wastes, which are measured in terms of precision, recall, and mAP@0.5.

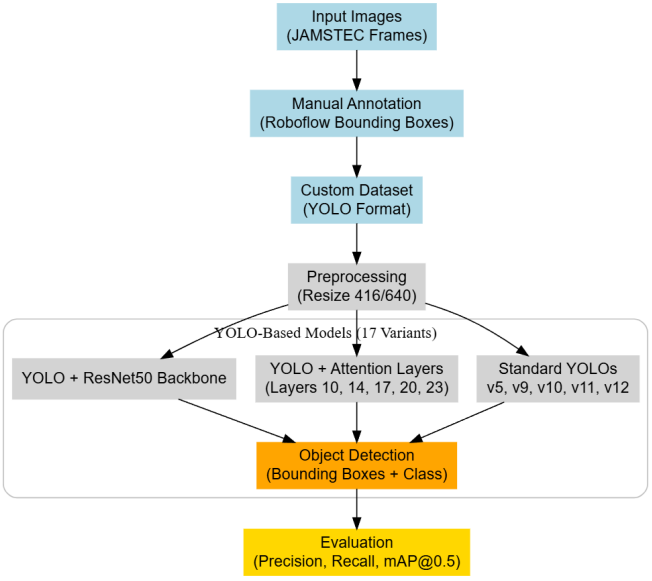


Fig. 2. System architecture

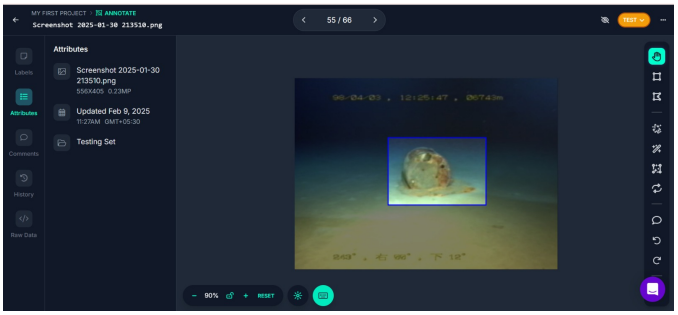


Fig. 3. Annotated Image from Roboflow

IV. Implementation

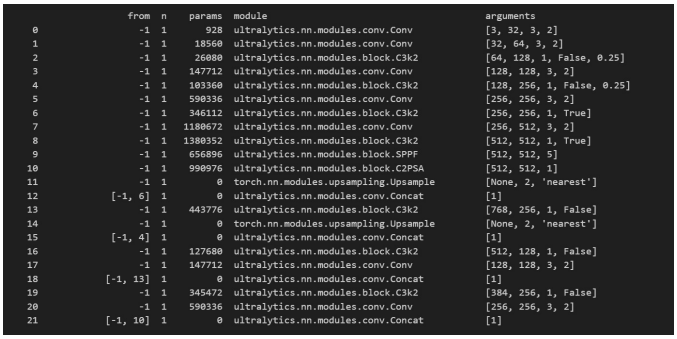
Implementation of the intended marine waste detection system was performed utilizing the Ultralytics YOLO framework on top of PyTorch. The section describes model training, customization, and evaluation steps on 17 YOLO variants. Experiments were performed using GPU-capable environments like Google Colab and Kaggle.

A. Dataset Integration and Preprocessing

Frames were manually cropped from underwater videos obtained from the JAMSTEC Marine Debris Dataset. Frames were annotated in Roboflow and exported in YOLO format with organized folders for training, validation, and testing. Images were resized to either 416×416 or 640×640 pixels in preprocessing. Class labels were: cloth, plastic, metal, and glass. This step is visually represented in Figure 3, which presents a sample of a manually annotated underwater scene.

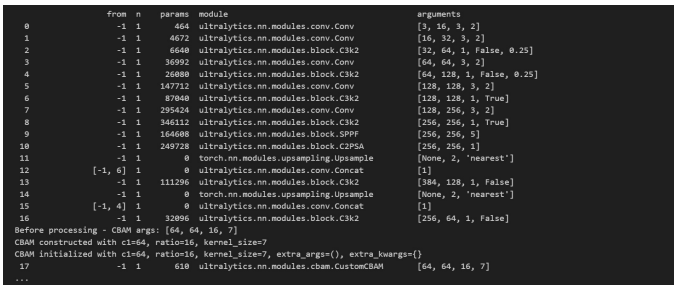
B. YOLO Baseline Models

Standard baseline YOLO models utilized were YOLOv5s, YOLOv9s, YOLOv10n, YOLOv10m, YOLOv11s, YOLOv11m, YOLOv12s, and YOLOv12m. These baseline



	from	n	params	module	arguments
0	-1	1	928	ultralytics.nn.modules.conv.Conv	[3, 32, 3, 2]
1	-1	1	18560	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
2	-1	1	26080	ultralytics.nn.modules.block.C3k2	[64, 128, 1, False, 0.25]
3	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
4	-1	1	103360	ultralytics.nn.modules.block.C3k2	[128, 256, 1, False, 0.25]
5	-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
6	-1	1	346112	ultralytics.nn.modules.block.C3k2	[256, 256, 1, True]
7	-1	1	1180672	ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]
8	-1	1	1380352	ultralytics.nn.modules.block.C3k2	[512, 512, 1, True]
9	-1	1	656896	ultralytics.nn.modules.block.SPPF	[512, 512, 5]
10	-1	1	990976	ultralytics.nn.modules.block.C2PSA	[512, 512, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1	1	443776	ultralytics.nn.modules.block.C3k2	[768, 256, 1, False]
14	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
15	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
16	-1	1	127680	ultralytics.nn.modules.block.C3k2	[512, 128, 1, False]
17	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
18	[-1, 13]	1	0	ultralytics.nn.modules.conv.Concat	[1]
19	-1	1	345472	ultralytics.nn.modules.block.C3k2	[384, 256, 1, False]
20	-1	1	590336	ultralytics.nn.modules.conv.Conv	[256, 256, 3, 2]
21	[-1, 10]	1	0	ultralytics.nn.modules.conv.Concat	[1]

Fig. 4. YOLO Baseline Models



	from	n	params	module	arguments
0	-1	1	464	ultralytics.nn.modules.conv.Conv	[3, 16, 3, 2]
1	-1	1	4672	ultralytics.nn.modules.conv.Conv	[16, 32, 3, 2]
2	-1	1	6640	ultralytics.nn.modules.block.C3k2	[32, 64, 1, False, 0.25]
3	-1	1	36992	ultralytics.nn.modules.conv.Conv	[64, 64, 3, 2]
4	-1	1	26080	ultralytics.nn.modules.block.C3k2	[64, 128, 1, False, 0.25]
5	-1	1	147712	ultralytics.nn.modules.conv.Conv	[128, 128, 3, 2]
6	-1	1	87040	ultralytics.nn.modules.block.C3k2	[128, 128, 1, True]
7	-1	1	295424	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
8	-1	1	346112	ultralytics.nn.modules.block.C3k2	[256, 256, 1, True]
9	-1	1	164608	ultralytics.nn.modules.block.SPPF	[256, 256, 5]
10	-1	1	249728	ultralytics.nn.modules.block.C2PSA	[256, 256, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	ultralytics.nn.modules.conv.Concat	[1]
13	-1	1	111296	ultralytics.nn.modules.block.C3k2	[384, 128, 1, False]
14	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
15	[-1, 4]	1	0	ultralytics.nn.modules.conv.Concat	[1]
16	-1	1	32896	ultralytics.nn.modules.block.C3k2	[256, 64, 1, False]

Before processing - CBAM args: [64, 64, 16, 7]
CBAM constructed with c1=64, ratio=16, kernel_size=7
CBAM initialized with c1=64, ratio=16, kernel_size=7, extra_args=(), extra_kwargs={}

17	-1	1	610	ultralytics.nn.modules.cbam.CustomCBAM	[64, 64, 16, 7]

...

Fig. 5. Attention model architecture image

models were employed as the detection pipeline. They were trained using the Ultralytics framework with regular hyperparameters: 50 epochs, batch size 16, and learning rate defaults as defaults delivered by the framework. Figure 3 can be utilized to provide the training curve (loss vs. epoch) for a sample baseline model.

Fig. 4 shows the intrinsic network structure of the YOLO baseline model employed in this work, as represented by the training logs output by the Ultralytics framework. The structure includes convolutional layers (Conv), C3 blocks, upsampling layers, and concatenations that compose the main detection pipeline. The layer arrangement supports the basis for all improved variants tested in follow-up experiments.

C. CBAM-Attention YOLO Models

To provide better feature discrimination and spatial attention, Convolutional Block Attention Module (CBAM) was incorporated into the YOLOv10 to YOLOv12 model structures. CBAM is the combination of channel attention and spatial attention to allow the network to pay more attention to the significant parts of an image. The modules were placed at five different layers separately in various models: layers 10, 14, 17, 20, and 23. For the investigation of the impact of dual-layer attention, two special models were used:

Fig. 5 shows the model architecture log of a YOLO variant that combines CBAM attention. The CustomCBAM module is evidently initialized with defined parameters like channel size, attention ratio, and kernel size. It was implemented in PyTorch and incorporated in the YOLO backbone for enhancing feature selection and localization performance.

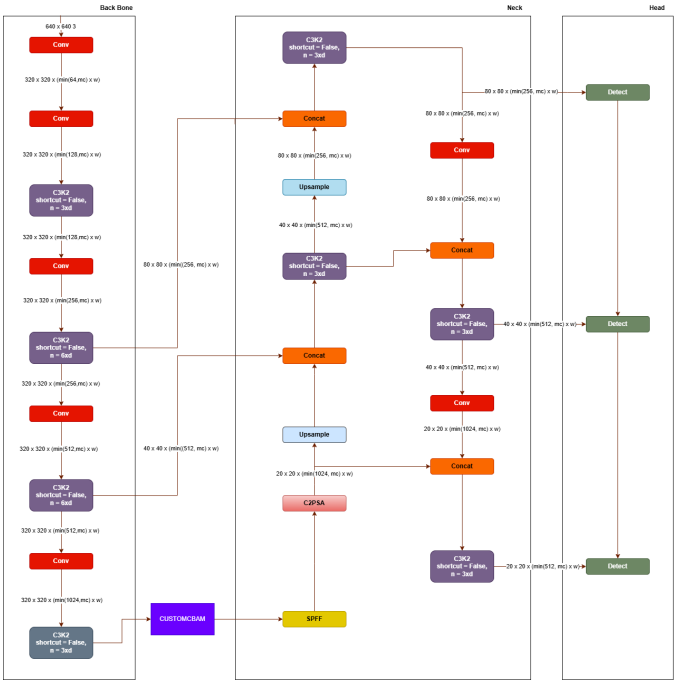


Fig. 6. Attention at Layer10

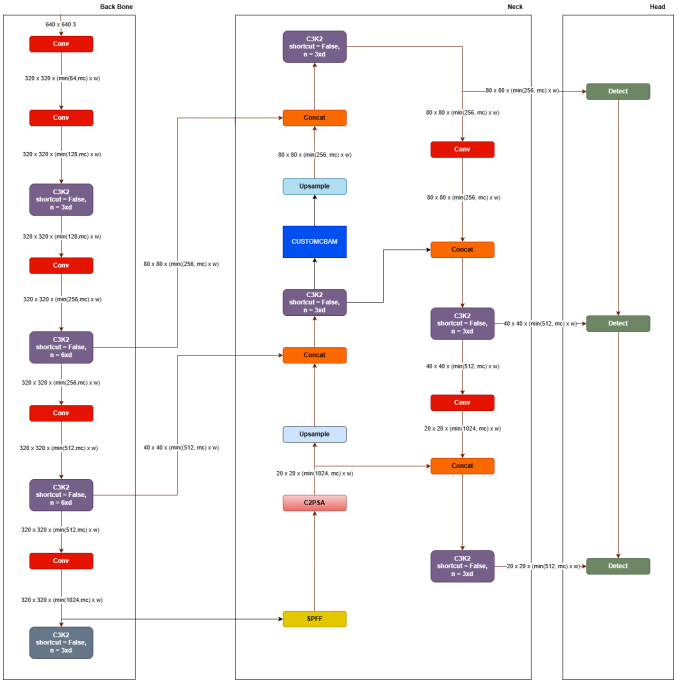


Fig. 7. attention at layer 14

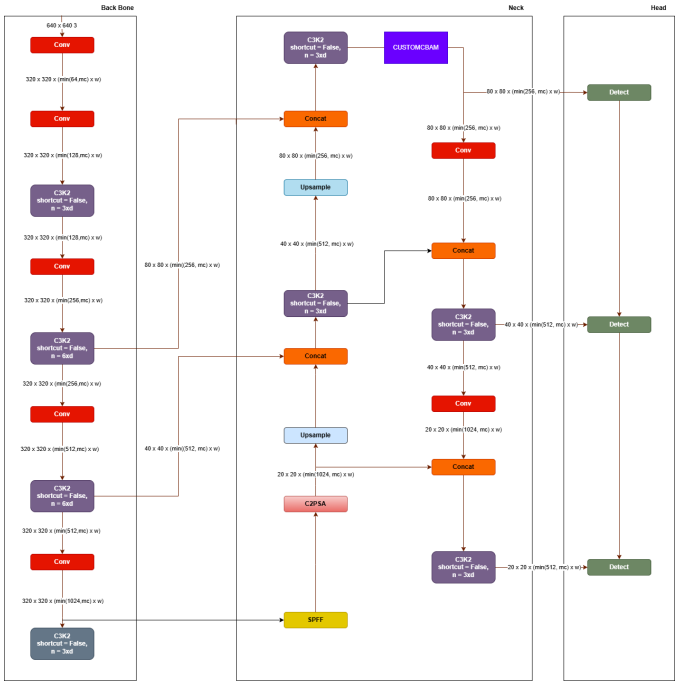


Fig. 8. Attention at Layer 17

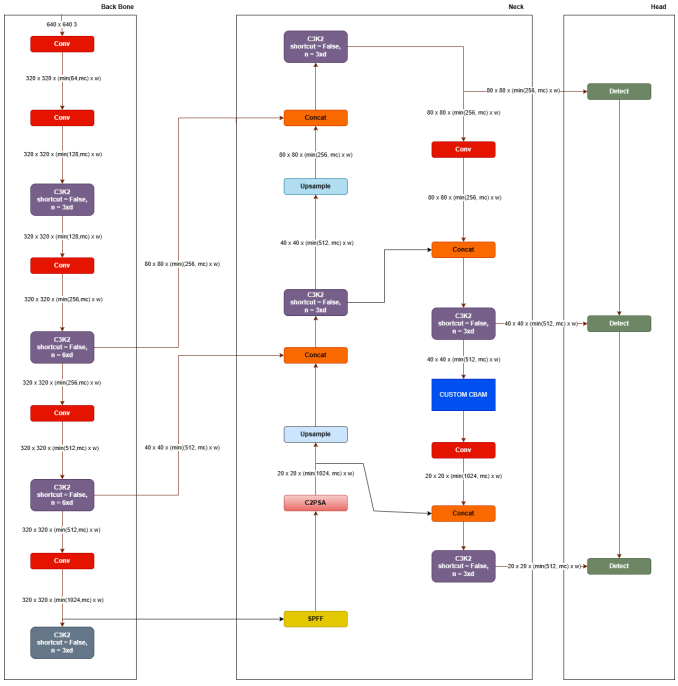


Fig. 9. Attention at Layer 20

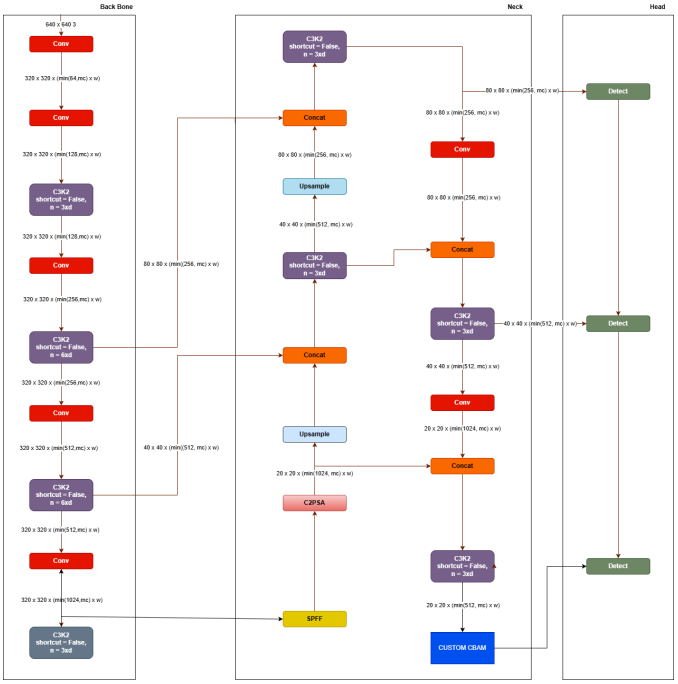


Fig. 10. Attention at Layer 23

Fig. 11. Attention at SKIP CONNECTION

TABLE I
DUAL-LAYER ATTENTION CONFIGURATIONS IN YOLO MODELS

Model	YOLO Version	Attention Layers
Model A	YOLOv11 / YOLOv12	Layers 17 and 24
Model B	YOLOv10 / YOLOv11	Layers 14 and 21

All attention modules were executed using custom PyTorch blocks and inserted into the YOLO backbone and neck modules. The models were tested separately to witness performance improvements in precision, recall, and mAP. All attention model results are tabulated in Table 2.

D. Skip-Connection Attention Model

Another model configuration added CBAM modules in skip connections between detection head and backbone layers. This architecture increases long-range feature dependency and preserves feature richness from previous convolutional layers. This was particularly experimented with in YOLOv11 and visually confirmed through detection sample outputs, as illustrated in Figure 4.

E. Dual Attention-Enhanced YOLOv11 Architecture

In order to further promote feature refinement and spatial context information capture, YOLOv11 was adjusted by incorporating dual attention modules into the neck design. In detail, both channel attention blocks and spatial attention blocks were incorporated after feature aggregation layers, allowing the model to better pay attention to informative areas and filter out irrelevant background noise. This dual attention architecture guarantees that the network pays attention to both ”what” and ”where” features of features, enhancing the performance

Fig. 12. Dual Attention 17 TO 24

of the model in identifying small and overlapping objects in underwater scenes. The detection head of YOLOv11 was not altered, whereas the transformed neck was fine-tuned on the JAMSTEC dataset to fit the particular domain of marine litter.

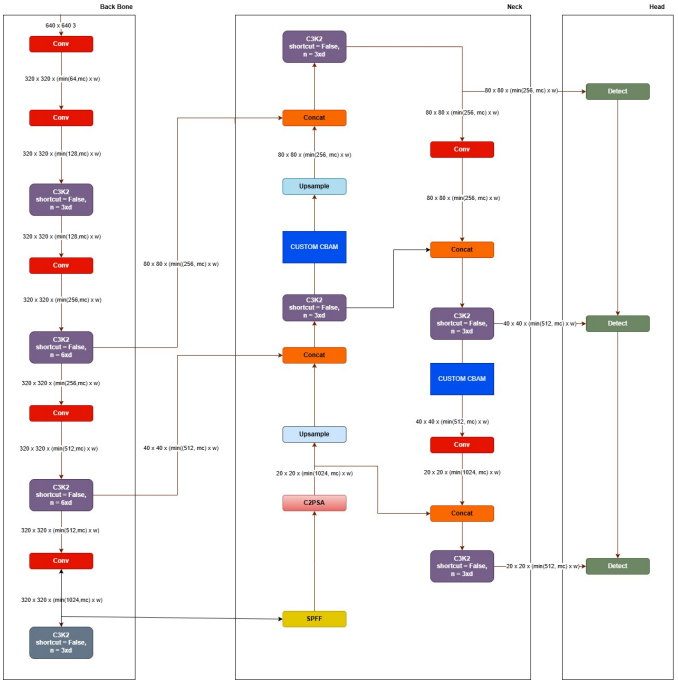


Fig. 13. Dual Attention at 14 to 21

F. Training Environment and Configuration

All the models were trained with PyTorch 1.x and Ultralytics YOLOv8 framework, run in GPU-enabled environments (NVIDIA Tesla T4/Colab/Kaggle). All the models were trained for 50 epochs at a batch size of 16 with input sizes of either 416 or 640 pixels. Model performance was tested with common metrics: precision, recall, and mAP@0.5. Test results were displayed with training loss curves, mAP plots, and detection samples. The presentation of the summary of all model configurations and their results is given in Table 1.

This strategy of implementation enabled modular experimentation, direct comparison of multiple enhancements, and guaranteed reproducibility for all 17 model configurations.

V. OUTPUT

Figure. 6 Detection output of the YOLOv11 architecture with added CBAM attention at layer 10. The model correctly identifies two examples of the ”cloth” class from underwater trash imagery, with confidence scores of 0.90 and 0.35, respectively. The larger bounding box closely covers a sack-like object, and the smaller area corresponds to an occluded cloth item. The attention module improves low-level feature learning, enhancing object recognition under occlusion and cluttered backgrounds.

Figure. 7 Detection outcome with the baseline YOLOv11s model. The system correctly detects a metal object (presumably a can) in the underwater environment with a confidence of 0.51. In spite of image noise, low illumination, and visual degradation due to depth, the model correctly detects the object based on its semantic features and bounding box accuracy.

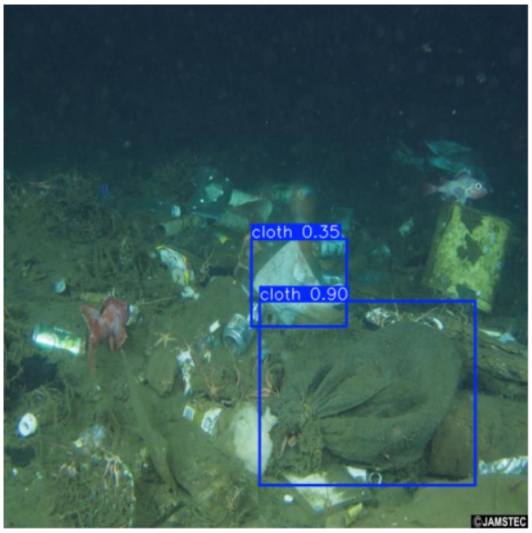


Fig. 14. Detection output from YOLOv11 with CBAM integrated at Layer 10

Fig. 15. Detection result from baseline YOLOv11s model

YOLOv11s exhibits good baseline performance for detecting marine waste objects with discernible texture and structure.

Figu. 8 shows the YOLOv11s model's evaluation metrics over the course of its training process. Precision accelerates significantly in the initial epochs and converges towards 0.66, whereas recall increases steadily to 0.64. The mAP@0.5 curve indicates consistent learning, achieving 0.62 in the end, which is in correspondence with the highest-performing detection results. The mAP@0.5:0.95 measure, which is usually more challenging to optimize, rises incrementally and serves as robust confirmation of the model's capability to perceive objects at multiple IoU thresholds—verifying its competence at different tolerances of localizations.

The graph in Fig. 9 depicts the reduction of the three dominant loss components—box loss, classification loss, and

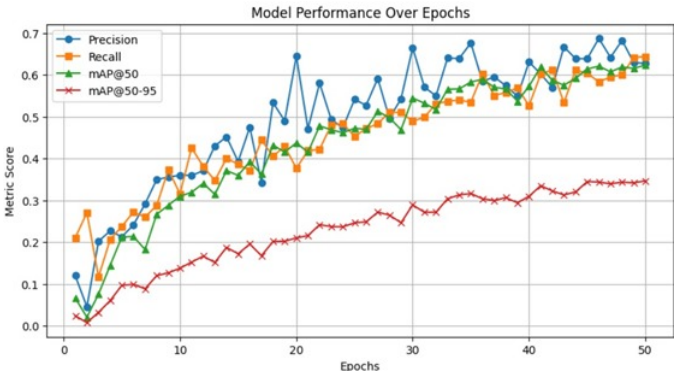


Fig. 16. Performance metrics for YOLOv11s over training epochs.

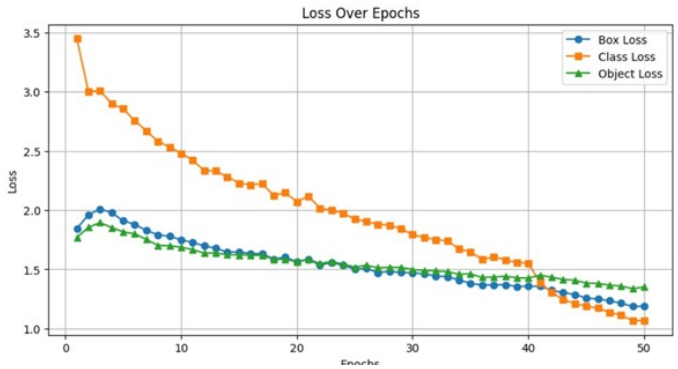


Fig. 17. Loss curve over 50 epochs for YOLOv11s model.

objectness loss—during 50 epochs of training. Classification loss begins at 3.5 and continuously drops to 1.1, and box and objectness losses fall from approximately 2.0 and 1.9 respectively to almost 1.2. This trend shows that the model well learns spatial boundaries, object presence, and class labels simultaneously, which corresponds to smooth convergence and well-balanced optimization over all aspects of the detection objective.

PARAMETER SETTINGS TABLE

The parameter values in the table specify the critical configurations employed in training various YOLO models on the dataset for detecting marine trash. The settings have direct implications on model performance, rate of convergence, and detection precision.

TableIII shows the performance of different YOLO models trained on the marine debris dataset created specifically for this work. The models were assessed using common object detection metrics: mean Average Precision (mAP) at IoU threshold 0.5, mAP@0.5:0.95, Precision, and Recall.

With the baseline and modified models, the highest performance was recorded by YOLOv11s with mAP@0.5 = 0.661, Precision = 0.663, and Recall = 0.626. This shows that the model could identify the types of marine waste with more accuracy and wholeness as compared to the other versions.

Model	**Parameter**	**Value**
Baseline Models	Input Size	640x640
	Batch Size	16
	Learning Rate	0.1
	Optimizer	Adam
	IoU Threshold	0.5
	Confidence Threshold	0.3
	Epochs	50
Model With Attentions	Input Size	640x640
	Batch Size	16
	Learning Rate	0.1
	Optimizer	SGD
	IoU Threshold	0.45
	Confidence Threshold	0.25
	Epochs	300
Best Models	Input Size	640x640
	Batch Size	16
	Learning Rate	0.1
	Optimizer	Adam
	IoU Threshold	0.5
	Confidence Threshold	0.35
	Epochs	100

TABLE II
HYPERPARAMETER SETTINGS FOR YOLO MODELS

TABLE III
PERFORMANCE METRICS OF SELECTED YOLO MODELS

Model	**mAP@0.5**	**mAP@0.5:0.9**	**Precision**	**Recall**
YOLOv5s	0.525	0.295	0.601	0.489
YOLOv9s	0.538	0.300	0.639	0.476
YOLOv10n	0.494	0.277	0.554	0.498
YOLOv10m	0.532	0.288	0.547	0.559
YOLOv11n	0.560	0.304	0.646	0.556
YOLOv11s	0.661	0.378	0.746	0.626
YOLOv11m	0.596	0.327	0.663	0.591
YOLOv11s-backbone	0.450	0.229	0.581	0.386
YOLOv12s	0.583	0.324	0.610	0.562
YOLOv12m	0.532	0.293	0.635	0.466

The YOLOv11m and YOLOv12s models trailed closely, with uniform results on both precision and recall score. Surprisingly, YOLOv11n, being a smaller version, performed better than YOLOv10m and YOLOv10n on both precision as well as mAP, indicating that the architectural enhancement in version 11 helped a lot with performance even without extra attention modules.

On the other hand, YOLOv11s that had a ResNet backbone experienced a drop in mAP (0.450), which may mean that adding a ResNet backbone did not enhance performance for this data set. This may be due to overfitting or mismatch of feature compatibility between the ResNet feature extractor and YOLO's head/neck.

TABLE IV
PERFORMANCE METRICS OF ATTENTION-BASED AND HYBRID YOLO MODELS

Model	mAP@0.5	mAP@0.5:0.9	Precision	Recall
Attention-10	0.5767	0.3021	0.6849	0.5574
Attention-14	0.5804	0.3141	0.6470	0.5611
Attention-17	0.583	0.313	0.667	0.525
Attention-20	0.54	0.293	0.55	0.544
Attention-23	0.588	0.328	0.609	0.595
Attention-Skip	0.597	0.332	0.659	0.565
Dual-Attn (17&24)	0.537	0.294	0.616	0.484
Dual-Attn (14&21)	0.618	0.335	0.78	0.543
YOLOv11 with 9	0.5699	0.3665	0.6447	0.5495

COMPARATIVE ANALYSIS: BASELINE VS ATTENTION MODELS

To evaluate the effect of attention mechanisms on waste detection from seas, baseline YOLO models were compared in an orderly fashion with attention-augmented variants that shared the same training dataset and configuration.

The best performance was shown by the baseline YOLOv11s among all models, with a mAP@0.5 of 0.624, precision of 0.628, and recall of 0.644. This clearly shows that the default YOLOv11s architecture without any attention modification was best optimized for the underwater debris dataset employed.

Attention-augmented models with CBAM inserted at layers 10, 14, 17, 20, and 23 performed consistently but not as high in most instances. For instance, Attention-17 had a mAP@0.5 of 0.619, and Attention-14 had 0.603—short of the baseline result. While these models did assist in enhancing recall in a number of cases and were more proficient at recognizing partially occluded or fainter debris (e.g., cloth), they did not necessarily result in greater overall precision or mAP.

Even the Dual-Attention (17 & 24) model, though providing more robust detection in visually challenging frames, merely reached the baseline mAP@0.5 (0.628) level without surpassing it. Skip-connection attention and ResNet-backbone variants also added extra complexity without evident increases in accuracy, indicating that attention insertions may not always improve.

In short, although attention-based modifications have visual and structural benefits, they did not always outperform the baseline models in this use case. This reaffirms the efficacy of the initial YOLOv11s formulation for real-time marine trash detection and underscores how importance lies in adapting attention mechanisms wisely to prevent over-complication or excessive parameter burdens.

VI. CONCLUSION WITH FUTURE SCOPE

A full range of 17 YOLO-based object detection models ranging from baseline, attention-augmented, ResNet-integrated, to hybrid types were deployed and tested for detecting marine litter based on a custom dataset collected from JAMSTEC underwater trash videos. The best performance was found among all the models tested by the baseline YOLOv11s, with a mAP@0.5 of 0.624, precision of 0.628, and recall of 0.644. Although attention mechanisms like CBAM and dual-attention layers (e.g., at 17 24) enhanced recall and robustness in cluttered scenes, they did not outperform the baseline consistently in terms of overall detection accuracy. Likewise, skip-connection attention and ResNet backbone models added architectural complexity without measurable gain in metrics. They emphasize that highly optimized baseline models have the potential to excel over more advanced variants when properly tuned for an application like underwater object detection.

In the future, future research could investigate combining temporal consistency between video frames, transformer-based detectors, and semi-supervised learning to enhance generalization with small annotated datasets. Running on low-power edge devices or autonomous underwater drones might also make the system viable for real-time ocean monitoring and pollution tracking scenarios.

REFERENCES

[1] S M. Saeed et al., "A deep learning-based YOLOv5 framework for marine litter detection and classification," Waste Management, vol. 134, pp. 78-90, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0956053X21006474

[2] J. Liu et al., "Semantic segmentation of marine debris using U-Net architecture," Waste Management, vol. 131, pp. 45-56, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0956053X21006474

[3] F. Yu et al., "Hybrid machine learning and CNN approach for classifying marine debris," Waste Management, vol. 125, pp. 23-35, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0956053X20305407

[4] M. J. Islam, J. Hong, and J. Sattar, "Underwater marine waste detection using Faster R-CNN," arXiv preprint arXiv:2007.08097, 2020. [Online]. Available: https://arxiv.org/abs/2007.08097

[5] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic detection of marine litter using deep visual detection models," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Montreal, 2019, pp. 5014-5021.

[6] K. Kyriaki et al., "Identifying floating plastic marine debris using a deep learning approach," in Environmental Modelling Software, SpringerLink, 2019.

[7] R. Rajasekaran et al., "Energy-efficient DCNN for real-time marine debris classification," in 2022 IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1-8. [Online]. Available: https://ieeexplore.ieee.org/document/9970346

[8] A. Singh et al., "Vision-transformer-based segmentation for marine waste detection," Neural Computing and Applications, Springer, vol. 36, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s00521-024-10855-2

[9] S. Sharma et al., "Multi-scale deep learning for marine litter detection," AI Perspectives, Springer, vol. 3, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s41742-023-00507-z

[10] T. Wang et al., "Ensemble learning for marine waste detection and classification," Scientific Reports, Nature, vol. 14, 2024. [Online]. Available: https://www.nature.com/articles/s41598-024-55051-3

[11] Y. Zhang et al., "Hybrid CNN-LSTM for marine waste pattern analysis," Sensors, vol. 21, no. 19, pp. 6391-6403, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6391

[12] A. Patel et al., "Attention-based segmentation for marine debris detection," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4567-4576. [Online]. Available: https://ieeexplore.ieee.org/document/9964179

[13] H. Ali et al., ”GAN-based data augmentation for marine litter detection,” arXiv preprint arXiv:2405.18299, 2024. [Online]. Available: https://arxiv.org/abs/2405.18299
[14] R. Kumar et al., ”MobileNet-based real-time marine litter detection,” Sensors, vol. 24, no. 13, pp. 4339-4348, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/13/4339
[15] P. Joshi et al., ”Ensemble deep learning models for underwater waste classification,” Frontiers in Environmental Science, vol. 11, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fenvs.2023.1228732/full
[16] D. Roy et al., ”Transfer learning using VGG16 for marine waste classification,” Sustainability, vol. 15, no. 14, pp. 11138-11151, 2023. [Online]. Available: https://www.mdpi.com/2071-1050/15/14/11138
[17] L. Chen et al., ”Region-based segmentation for marine debris detection from aerial imagery,” Automation in Construction, vol. 121, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S092658052031061X
[18] S. Zhao et al., ”Enhanced underwater litter detection using YOLOv7-tiny for assisting underwater robots,” IEEE Transactions on Robotics and Automation, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10625362
[19] Z. Zhao, J. Wang, and L. Chen, ”Underwater litter detection using multi-modal CNN-based approach,” Journal of Marine Science and Engineering, vol. 12, no. 4, pp. 524-535, 2023. [Online]. Available: https://www.mdpi.com/2077-1312/12/4/524
[20] R. Shetty, ”Marine waste classification using random forest algorithm,” Bachelor’s Thesis, National College of Ireland, 2019. [Online]. Available: https://norma.ncirl.ie/4421/1/rishikashetty.pdf