Comparative Analysis of Different Underwater Trash Detection Models

47999 [1] **42502** [1] **43263** [1] **51348** [1]

Abstract

Marine plastic pollution poses a critical threat to ocean ecosystems, biodiversity, and human health. Traditional monitoring methods are often limited by scalability and inefficiency due to harsh underwater conditions. This project explores some of the state-of-the-art deep learning-based object detection frameworks to test their performance in real-time underwater trash detection and segmentation. Models such as YOLOv8, EfficientDet, and U-Net with various backbones, including VGG, ResNet, and ConvNeXt, were trained and evaluated on the Ocean Waste dataset for their ability to detect and segment diverse underwater waste objects in visually challenging environments. Our system leverages targeted data augmentation and class balancing to mitigate dataset imbalances and improve the detection of underrepresented debris types through extensive experimentation and visual inspection. This study contributes to the growing field of oceanic AI research by offering analysis to the scalable and adaptable approaches that support marine cleanup efforts and sustainable ocean monitoring.

1. Introduction

Oceans, covering over 70% of the Earth's surface, are facing escalating threats from human induced pollution foremost among them, is marine debris. Plastic waste, in particular, has emerged as one of the most pressing global environmental challenges. According to the United Nations Environment Programme (UNEP), more than 8 million tons of plastic enter the oceans annually, with devastating ecological, economic, and health-related consequences. Marine organisms frequently ingest or become entangled in plastic debris, leading to injury or death. Meanwhile, microplastics have been detected in the food chain, posing long-term risks to human health. Beyond harming biodiversity, underwater trash degrades marine habitats, disrupts fishing and shipping industries, and negatively impacts coastal tourism.

Traditional monitoring methods such as manual scuba surveys, sonar imaging, and remotely operated vehicles (ROVs) are often expensive, labor-intensive, and limited in coverage. Their effectiveness is further compromised by underwater environmental challenges like low visibility, variable lighting conditions, and turbidity. These constraints underscore the urgent need for scalable, automated systems capable of detecting and classifying underwater waste in real-time.

In recent years, deep learning has transformed image analysis and object detection across numerous domains, yet its application to underwater environments remains relatively nascent. Motivated by prior work such as "A Robotic Approach Towards Quantifying Epipelagic Bound Plastic Using Deep Visual Models", this project seeks to leverage state-of-the-art deep learning techniques to develop an AI-powered system for accurate detection, classification, and segmentation of underwater trash.

To address the inherent complexities of underwater scenes, our system incorporates advanced object detection architectures - namely YOLOv8, EfficientDet, and U-Net augmented through transfer learning and tailored data augmentation strategies. These models are trained on the Ocean Waste dataset, which contains annotated images of marine debris spanning across various underwater conditions, waste types, and object scales.

By focusing on both performance and practicality, the project aims to create a robust solution suitable for deployment in real-world scenarios, including integration with autonomous underwater vehicles (AUVs) or support for marine cleanup initiatives. In doing so, this work contributes not only to technological advancement in underwater computer vision but also to broader global efforts in marine conservation and environmental sustainability.

2. Problem Description

Marine pollution particularly the proliferation of plastic waste has emerged as a critical threat to aquatic ecosystems, biodiversity, and public health. Accurate monitoring and detection of marine debris are vital to inform cleanup operations, quantify ecological damage, and support sustainable marine conservation efforts. However, traditional detection methods such as diver-led inspections, sonar-based mapping, and manual image annotation are often inefficient, resource-intensive, and unsuitable for large-scale deployment.

The core technical challenge lies in accurately identifying

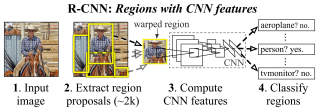

Figure 1. R-CNN: Regions with CNN Features

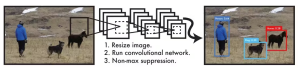

Figure 2. The YOLO Detection System

and localizing debris within underwater imagery, which is often degraded by conditions such as low visibility, turbidity, inconsistent lighting, and color distortion. In addition, marine debris exhibits wide variability in size, shape, material composition, and visual appearance, making it difficult for conventional object detection systems to generalize effectively, especially when trained on terrestrial datasets.

From a computer vision perspective, the task is inherently multifaceted, encompassing:

1. Object detection – identifying the presence and location of debris within an image,

2. Classification – assigning the correct label to each detected item (e.g., bottle, glove, net),

3. Segmentation – Outlining the precise spatial boundaries of the objects for further analysis or manipulation.

To meet these challenges, this project investigates the application of deep learning-based models specifically tailored for underwater environments. Our focus is on evaluating and comparing object detection architectures such as YOLOv8, EfficientDet, and U-Net, each chosen for their potential to balance speed, accuracy, and robustness under harsh underwater conditions. These models are trained and validated on the Ocean Waste dataset, which contains various annotated images of marine debris in various underwater scenes.

By incorporating domain-specific data enhancement strategies and addressing class imbalance issues, the proposed system aims to deliver high-performance, real-time detection capabilities. Ultimately, our goal is to develop a scalable and adaptable solution that can integrate with autonomous underwater vehicles (AUVs) or monitoring platforms to enhance the efficiency and effectiveness of marine debris tracking in real-world oceanic settings.

2.1. Related Work

Traditional image detection algorithms and despite the high accuracies researchers managed to get pose limitations in complex environments like underwater, where the complexity stems from the high level of noise (blurring, low contrast and color deviation) present in underwater imagery. Deep learning has been in the spotlight for underwater detection due to its capabilities in feature extraction and representation.

In 2014, (Girshick et al., 2014) introduced R-CNNs or Region-based Convolutional Neural Networks as a breakthrough in computer vision and in object detection specifically, as it addressed the location of objects within images in bounding boxes. Many models built on the idea of R-CNNs such as Fast R-CNN (Girshick, 2015), Faster R-CNNS (Ren et al., 2016), and Mask R-CNNs (He et al., 2017), all of which attempted to improve the performance of the baseline R-CNN. The concept these models introduced, and as seen in Figure 1 is multi-stage detection where in the first stage regions of interest are identified (up to 2000 regions), then these regions are parsed and passed to pre-trained CNNs to generate feature maps. The feature mapping result is then passed to a classifier (original R-CNN used Support Vector Machines, but recent architectures started using softmax for classification). Then, another post-processing regression step is used to refine the bounding boxes. While these methods are high in accuracy, they are slow, computationally expensive, and sometimes they might process redundant areas due to the high number of regions per image. The baseline R-CNN originally proposed achieved more the 30% mAP compared to the previous benchmark at that time.

YOLO, or Unified Real-Time Object Detection (Redmon et al., 2016), approached object detection as a regression problem to spatially separated bounding boxes and associated class probabilities instead of a classification problem as the method above, thus a one stage approach. The name comes from the main concept this model is built on, Figure 2, where each image is processed only once and straight from the pixels, the objects are identified. This method is extremely fast as it doesn't rely on patching the image or sliding a window, and also it makes fewer errors on background noise compaared to Fast and Faster R-CNNs. The drawback of YOLO presents itself in localisation, and while the model can identify objects pretty fast, it struggles in precise localisation. Thus, despite being fast, this model still trailed the accuracy of the multi-stage detectors.

In 2017, some researches (Lin et al., 2017) attempted to improve the accuracy of one-stage detectors by addressing the main problem that was impeding them from matching the accuracies of multi-stage detectors, but retaining the speed. They attempted to identify class imbalance during training and introduced a new loss function, dynamically scaled cross-entropy loss, and a focal loss. This loss forces the model to focus on hard objects in the image during the training. Their model, RetinaNet surpassed both the speed of one-stage detectors and the accuracy of multi-stage detectors.

3. Models

This section presents the deep learning models used in the development of the underwater trash detection system. Each model was selected based on its ability to perform accurate object detection and segmentation under challenging visual conditions such as low contrast, turbidity, and lighting inconsistencies - common in underwater environments. The architectures covered include YOLOv8, EfficientDet, and U-Net with backbone variants such as VGG, ResNet, and ConvNeXt. These models differ in design principles, detection speed, and feature extraction capabilities, allowing us to explore trade-offs between performance and computational efficiency. For each model, we describe the architecture, training methodology, numerical results, and interpretation of performance in the context of marine debris detection.

3.1. YOLOv8

3.1.1. Architecture

YOLOv8 is a single-stage object detector designed for high speed and accuracy. It introduces an anchor-free detection head, allowing the model to directly predict object centers and bounding boxes without relying on predefined anchor boxes. This simplifies the architecture and improves generalization.

The model features a decoupled head, separating classification and regression tasks to enhance learning efficiency. It also uses a lightweight CSP-based backbone for effective feature extraction with reduced computational cost.

In our project, we used the YOLOv8s variant, which offers a strong balance between detection performance and real-time processing, making it well-suited for underwater trash detection where fast and reliable inference is essential.

3.1.2. Training Methods

To train the YOLOv8 model effectively for underwater trash detection, we first prepared our dataset by mounting Google Drive in Google Colab and extracting the required data from a ZIP archive. We then defined a custom YAML file outlining the dataset structure, specifying paths for training, validation, and test images, along with 15 distinct class labels representing various trash types (e.g., plastic bag, bottle, rope, mask, etc.). Recognizing the class imbalance in our dataset, we applied targeted data augmentation for underrepresented classes using horizontal flips and brightness adjustments. The original and augmented images were then consolidated into a combined training directory. Training was initiated using a pre-trained YOLOv8n model with transfer learning. Hyperparameters included a batch size of 16, input image size of 640×640, learning rate of 0.01, and dropout of 0.15. Additional augmentation parameters such as hue/saturation variation, mosaic augmentation, and image flipping were enabled to improve model robustness. The training process was conducted over 50 epochs with early stopping patience set to 10, and results were logged to a dedicated project folder for evaluation. This systematic approach helped enhance the model's ability to generalize across diverse underwater scenes while preserving computational efficiency.

3.1.3. Numerical Results

To evaluate the performance of our underwater trash detection model, we trained the YOLOv8n architecture for 50 epochs using a combined dataset that included both original and synthetically augmented images.

Throughout the training process, key loss metrics box loss, classification loss, and objectness loss were continuously monitored. A consistent decline in these values across epochs indicated stable convergence. We incorporated early stopping with a patience value of 10 to prevent overtraining while ensuring that the model had sufficient opportunity to generalize well on unseen data.

After training, we evaluated the model on a held-out test set of underwater images. The inference results showed accurate detection of multiple classes, including frequently occurring objects like plastic bags, bottles, and cups, as well as less represented categories such as ropes, straws, and wrappers. Detection confidence scores were generally high, often exceeding 0.85 for common objects. The model achieved an average inference time of 7.5 milliseconds per image, making it suitable for real-time applications such as underwater drone deployments.

Prediction outputs were saved and visually inspected to validate model performance. The model was able to correctly localize and classify overlapping objects in cluttered underwater environments. We observed that after implementing class-specific augmentation (targeting minority classes like gloves, tubes, and fragments), the model's sensitivity to these rare classes improved notably, reducing misclassification errors that were previously more common. These enhancements confirmed the importance of balancing the dataset during training, particularly in tasks with skewed class distributions.

3.1.4. Interpretation

The YOLOv8 model exhibits a strong ability to detect and localize underwater trash objects across a wide range of visual conditions. It performs well even in complex scenarios involving occlusions, overlapping objects, and visually noisy backgrounds like coral beds or sediments. The model accurately identifies both large and small debris items, such as bottles, cups, wrappers, and even thin objects such as

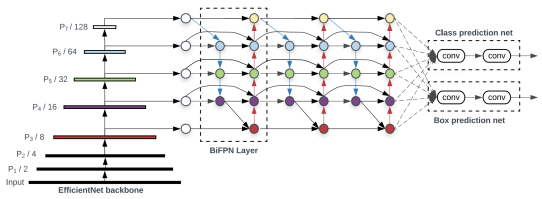

Figure 3. EfficientDet architecture

straws and gloves, highlighting its sensitivity to fine-grained spatial details.

While it handles most classes with high confidence, occasional confusion is observed between visually similar categories such as masks and plastic bags. These overlaps likely stem from shared textures or shapes and suggest that clearer class separation or additional training examples could further improve precision. Despite this, the model maintains overall consistency, particularly for objects with distinctive shapes or color contrast.

Moreover, the model's ability to generalize across varied lighting and turbidity conditions without explicit tuning suggests robustness is suited for real-world underwater deployments. It's consistent output across diverse environments demonstrates its potential for practical applications in marine waste monitoring and autonomous underwater navigation systems.

3.2. EfficientDet

3.2.1. Architecture

EfficientDet was proposed as an attempt to scale one-stage object detection algorithms in such a way that increases the performance without complicating the model and the training time. It builds on the scalability concept of EfficientNets for image classification but for detection tasks, thus they proposed a family of EfficientDet models (D0, D1, D2, D3, D4, D5, D6, D70) each having different number of parameters and requiring different computational resources. Rather than a grid search, they used a compound-scaling method, heuristic-based scaling approach to jointly scale all the dimensions. EfficientDet, developed by Google, proposed a breakthrough in the performance of one-stage detectors over two-stage detectors that tend to be more complex and computationally expensive.

The optimized architecture of this model, as shown in Figure 3 compromises pretrained **EfficientNets** as backbone network. Which is trained on ImageNet and responsible for extracting hierarchical features from the input images at different scales.

These features are then passed to a **BiFPN**, bi-directional feature pyramid network. This idea they proposed that leverages both top-down and bottom-up feature fusion allowed for multi-scale feature fusion, introducing learnable weights that allowed the model to learn the importance of different input features.

A shared **Box/class prediction network** is the last network of the model that has a fixed width equal to the BiFPN layer before it.

3.2.2. Training Methods

This study utilizes TensorFlow Object Detection API, from which a pre-trained EfficientDet-D0 model was fine-tuned on the training dataset. This API requires data to be TensorFlow Records (TfRecords) format, tensorflow's own binary storage format that increased the speed and performance of pipelines dealing with large datasets. A TFRecord version of the dataset was obtained from Roboflow website and used for this model. The dataset was enhanced by data augmentation techniques to improve the generalization of the model.

The transfer learning approach significantly reduced the training time by using the checkpoints from obtained from training on COCO (common objects in context) as a starting point, and building on them to adjust the weights. The API has a pipeline for each model that can be customized for different use cases. This file was edited to match the requirements of the dataset (i.e number of classes) and also the available resources (batch size, learning rate). Despite D0 being the lightest version of EfficientDet model, it was computationally expensive to fine-tune on our limited access to GPUs. The model was trained for 10400 steps (around 10 hours on A100 GPU), through which the loss was monitored until it reached a low value of 0.5.

The key performance metrics used to assess the model are:

- Intersection over Union (IoU), a comparison between the ground truth bounding box and the predicted box, by calculating the ratio between their intersection and union.
- Average Precision (AP), the mean average precision is calculated over multiple IoUs on different thresholds. The higher the IoU threshold, the more strict the result obtained. Also, the AP is obtained across scales for different object sizes. This indicated that the performance is dependent on the detected object size.
- Average Recall (AR), measures the extent to which the model finds all ground truth objects that exist in the image.

3.2.3. Numerical Results

The model was evaluated on a held-out test dataset that has 501 instances (around 10% of the dataset). The results

are reported in table 1. Both AP and AR are measured on different IoU thresholds, ranging from 0.5 to 0.95 in a step of 0.05.

Table 1. Test Performance of EfficientDet Model

Metric	Value
AP@[IoU=0.50:0.95] (all)	0.240
AP@[IoU=0.50] (all)	0.415
AP@[IoU=0.75] (all)	0.234
AP@[IoU=0.50:0.95] (small)	0.091
AP@[IoU=0.50:0.95] (medium)	0.202
AP@[IoU=0.50:0.95] (large)	0.259
AR (small objects)	0.160
AR (medium objects)	0.405
AR (large objects)	0.507

3.2.4. Interpretation

The performance of the model and as reported in the numerical results, is moderate for this real-world complex underwater imagery. Despite the promising state-of-the-art performance obtained on the COCO dataset that often exceeds a mAP of 0.55, the overall mAP obtained in this case is 0.24. These results can be considered a reasonable start, considering the relatively short training period, the use of the smallest EfficientDet model D0 and the complexity of the task.

When relaxing the localisation requirements, the model has proved to perform much better, obtaining a mAP of 0.415 at an IoU threshold of 0.50, suggesting that the model is effectively learning the visual characteristics of the different types of ocean waste.

The significant drop in performance with a stricter IoU threshold (0.234 at 0.75) indicates that while the model could successfully identify the objects, it struggles in finding the accurate bounding boxes. This localisation accuracy may be sufficient for general area identification tasks, but requires further refinement for the use in tasks that require precise object positioning.

Another important finding is the effect of the object size on the model's performance. The model is performing much better on larger objects compared to smaller ones (0.507 vs 0.16 AR), and the AP significantly drops as the objects become smaller.

3.3. U-Net

3.3.1. Architecture

The U-Net architecture represents a significant advancement in image segmentation tasks, particularly within *medical imaging* and *environmental monitoring* applications. Originally proposed by Ronneberger et al. in 2015 (Ronneberger et al., 2015), the U-Net's distinctive **U-shaped structure** has proven remarkably effective for precise pixel-wise classification. This study explores the integration of three distinct **backbone networks**—*VGG16*, *ResNet50*, and *ConvNeXT*—into the U-Net framework for *ocean waste segmentation*.

The fundamental U-Net architecture comprises an **encoder pathway** that captures context and a **decoder pathway** that enables precise localization. The encoder progressively reduces spatial dimensions while increasing feature depth, effectively capturing hierarchical representations. In contrast, the decoder pathway restores spatial resolution through upsampling operations. A defining characteristic of U-Net is its use of **skip connections**, which concatenate feature maps from corresponding encoder layers to decoder layers, preserving *fine-grained spatial information* that would otherwise be lost during downsampling (Ronneberger et al., 2015).

VGG16 Backbone: Developed by the Visual Geometry Group at Oxford, VGG16 employs a straightforward architecture of *stacked* 3×3 *convolutional layers* followed by max pooling. Despite its simplicity and age, it serves as a strong baseline with approximately **138 million parameters**. Features are extracted from five key layers (`block1_conv2`, `block2_conv2`, `block3_conv3`, `block4_conv3`, and `block5_conv3`) to form skip connections at various resolutions. Previous work has shown VGG16's effectiveness in U-Net frameworks for *semantic segmentation in remote sensing* (Zhang et al., 2023).

ResNet50 Backbone: This backbone introduces *residual connections*, which alleviate the vanishing gradient problem in deep networks. These allow the model to learn *residual mappings* rather than direct transformations, supporting deeper architectures. With around **23 million parameters**, ResNet50 is more parameter-efficient than VGG16. Features are extracted from `conv1_relu`, `conv2_block3_out`, `conv3_block4_out`, `conv4_block6_out`, and `conv5_block3_out`. Its successful integration into U-Net for *biomedical image segmentation* has been confirmed in prior studies (Nande et al., 2023).

ConvNeXT Backbone: A modern architecture inspired by Vision Transformers but implemented using convolutional principles. It features *depthwise separable convolutions*, *inverted bottlenecks*, and *layer normalization*, diverging from traditional CNN designs. With around **28 million parameters**, ConvNeXT achieves state-of-the-art performance. Features are drawn from stages 1–4 of the network. Its application within U-Net pipelines has shown strong

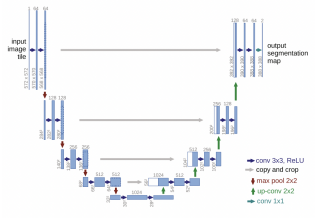

Figure 4. The U-net architecture

results in *multi-scale urban scene segmentation* (Wu & Li, 2025).

Decoder Integration and Adaptation: Integrating these heterogeneous backbones into a unified U-Net architecture poses challenges due to differences in spatial and channel dimensions. Notably, ConvNeXT outputs feature maps with dimensions different from those of VGG16 and ResNet50. To address this, the decoder applies 1×1 **convolutions** to match channel sizes and uses **bilinear interpolation** to align spatial dimensions before skip connection concatenation.

3.3.2. Training Methods

Dataset and Preprocessing This study utilizes the *Ocean Waste Dataset*, which is initially annotated in the YOLO format, representing objects through bounding box coordinates. To facilitate segmentation-based training with the U-Net architecture, these bounding box annotations were systematically converted into binary segmentation masks, where the foreground (waste objects) is labeled as 1 and the background as 0.

The images and their corresponding masks were resized to a fixed dimension of 224×224 pixels to ensure uniformity across the dataset, except for models requiring different input sizes. The dataset was partitioned into training, validation, and testing subsets to enable robust model evaluation.

To enhance model generalization and prevent overfitting, data augmentation techniques were applied exclusively to the training set. These augmentations included horizontal and vertical flips, random rotations, brightness and contrast adjustments, Gaussian blur, Gaussian noise addition, and color (including RGB shift transformations). All augmentation operations were implemented using the `Albumentations` library, integrated within the custom data generator.

Loss Function and Metrics The segmentation models were trained using a composite loss function that combines Binary Cross Entropy (BCE) with Dice Loss, referred to as BCE + Dice Loss. The BCE component addresses pixel-wise classification errors, while the Dice Loss is particularly effective for managing class imbalance by emphasizing the overlap between predicted and ground truth masks, an important consideration given the small object sizes in the ocean waste dataset.

Performance evaluation was conducted using the following metrics:

- **Dice Coefficient**, measuring the similarity between the predicted segmentation and the ground truth.
- **Intersection over Union (IoU) Score**, also known as the Jaccard Index, quantifying the ratio of the intersection to the union of predicted and actual mask regions.
- **Binary Accuracy**, representing the proportion of correctly classified pixels across the entire mask.

These metrics provide a comprehensive assessment of segmentation performance, particularly for datasets where the foreground class occupies a relatively small portion of the image.

Optimization Strategy The optimization of the U-Net models was performed using the Adam optimizer with an initial learning rate of 1×10^{-4}. To facilitate adaptive learning, a `ReduceLROnPlateau` callback was employed, which reduces the learning rate by a factor of 0.5 if no improvement in validation loss is observed. Additionally, a learning rate scheduler was implemented, applying exponential decay after the first 10 epochs to promote stable convergence.

To prevent overfitting and ensure efficient training, several regularization strategies were incorporated:

- **Early stopping**, monitored on the validation Dice Coefficient with a patience of 10 epochs, allowing the restoration of the best-performing model weights.
- **Model checkpointing**, configured to save the model that achieves the highest Dice Coefficient on the validation set.

The models were trained for a total of 30 epochs with a batch size of 8, balancing training stability with computational efficiency.

3.3.3. Numerical Results

To evaluate the effectiveness of the U-Net architecture with different backbone networks, both basic and advanced variants were assessed across four key segmentation metrics: Loss, Dice Coefficient, Intersection over Union (IoU), and Binary Accuracy. The models were evaluated using held-out test data after completing 30 training epochs. The results are summarized in the table below:

Table 2. Test performance comparison of basic and advanced U-Net models with different backbones.

Backbone	Model Type	Loss	Dice Coefficient	IoU Score	Binary Accuracy
VGG16	Basic	0.3359	0.8518	0.7449	0.9326
VGG16	Advanced	0.3361	0.7429	0.9330	0.8510
ResNet50	Basic	0.2957	0.8704	0.7734	0.9409
ResNet50	Advanced	0.2907	0.7804	0.9402	0.8750
ConvNeXt	Basic	0.3130	0.8794	0.7877	0.9427
ConvNeXt	Advanced	0.3026	0.7723	0.9401	0.8701

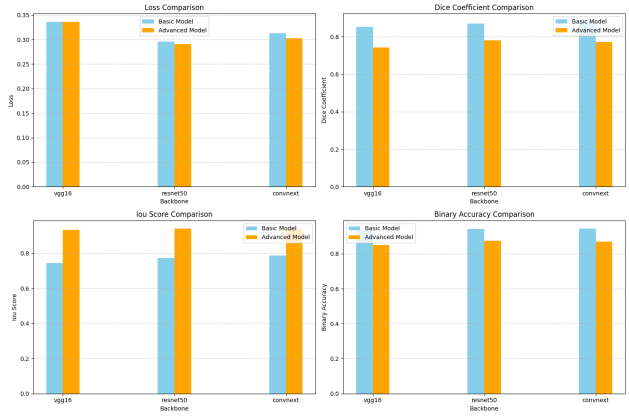

Figure 5. Comparison of segmentation metrics (Loss, Dice Coefficient, IoU Score, Binary Accuracy) across basic and advanced U-Net models with VGG16, ResNet50, and ConvNeXt backbones.

3.3.4. Interpretation

The experimental findings highlight notable trends in the behavior of basic and advanced U-Net configurations with different backbones.

Basic ConvNeXt emerged as the best-performing model overall, achieving the highest Dice Coefficient (0.8794), IoU Score (0.7877), and Binary Accuracy (0.9427), with a competitive loss of 0.3130. This reflects ConvNeXt's strong feature extraction capabilities even in its baseline integration, particularly when trained on moderately sized datasets like ours.

Among the advanced variants, ResNet50 performed best, recording a Dice Coefficient of 0.7804 and IoU Score of 0.9402, indicating that its residual connections help maintain spatial features even under more complex training regimes.

Interestingly, advanced models did not consistently outperform their basic counterparts. In fact, advanced variants for all three backbones - VGG16, ResNet50, and ConvNeXt - showed reduced Dice Coefficients compared to their basic configurations. For instance, the Dice Coefficient for advanced VGG16 dropped by 12.79%, ConvNeXt by 12.18%, and ResNet50 by 10.34%, despite some improvements in IoU or binary accuracy.

This suggests that while advanced models incorporate deeper architectures, regularization (e.g., dropout), and learning rate schedules, these enhancements may not always translate to segmentation accuracy on this dataset. In particular, over-regularization may suppress the network's ability to learn precise boundary information, which is critical for segmenting small, detailed structures such as marine litter.

From a loss perspective, advanced ResNet50 and ConvNeXt achieved the lowest loss values (0.2907 and 0.3026 respectively), reinforcing their stability and consistency during training. However, this low loss did not always correspond to the best Dice performance, once again indicating that loss alone is not sufficient to judge model segmentation quality.

Moreover, binary accuracy appeared to remain relatively stable across models, though basic backbones generally retained a slight edge. This stability may reflect that most pixels in the mask are background, and accuracy remains high even when foreground (waste) segmentation struggles - further emphasizing the importance of using Dice and IoU in unbalanced segmentation tasks.

In conclusion, while backbone choice significantly impacts U-Net performance, the complexity of "advanced" training setups must be weighed against the risk of performance degradation due to overfitting controls. ConvNeXt's basic variant stands out as the most robust and effective in this context, while ResNet50 advanced strikes a balance between generalization and segmentation precision.

4. Limitations and Future Work

Despite the encouraging results achieved in the detection and segmentation models, certain limitations constrain the scope of this study. The dataset used, while diverse, exhibited class imbalance and intraclass visual similarity, occasionally leading to misclassification, particularly between object types such as gloves, masks, and plastic bags. Additionally, the models were evaluated on RGB imagery alone, without access to complementary sensory inputs like depth or sonar, which could enhance detection under challenging underwater conditions.

Future work will focus on expanding the dataset to include more varied underwater scenes, improving label quality for visually ambiguous classes, and incorporating multi-modal data sources to strengthen contextual understanding. Future iterations may explore Vision Transformer based architectures such as SegFormer or SwinUNet to further improve segmentation accuracy. There is also significant potential in optimizing the models for real-time deployment on embedded hardware aboard autonomous underwater vehicles, enabling scalable, and efficient marine waste detection in real-world underwater environments with minimal human

supervision.

5. Conclusion

This study presents a comparative analysis of deep learning models for the task of underwater trash detection, addressing a critical environmental challenge through technological innovation. By evaluating YOLOv8, EfficientDet, and U-Net (with multiple backbone variations), we explored both object detection and semantic segmentation approaches, considering their adaptability to the unique visual and computational constraints of the underwater domain.

Our findings demonstrate that single-stage detectors like YOLOv8, when fine-tuned with class-specific augmentation and carefully chosen hyperparameters, can achieve fast and accurate multi-class detection suitable for real-time applications, such as autonomous underwater vehicles. EfficientDet, while computationally heavier even in its lightweight D0 variant, showed promise in terms of bounding box quality and precision, particularly when evaluated at relaxed IoU thresholds. U-Net-based segmentation models, especially those built on the ConvNeXt and ResNet50 backbones, delivered strong pixel-level segmentation performance, with notable strengths in shape delineation and boundary localization, key features for precise waste quantification and mapping.

A key insight across all models is the trade-off between speed, accuracy, and localization precision. The incorporation of targeted augmentations, loss functions tailored to imbalanced data, and backbone-aware architectural design contributed significantly to model performance across varying object sizes and visual contexts. Furthermore, the results reaffirm the importance of dataset diversity, augmentation strategies, and backbone selection when applying deep learning models to environmental monitoring tasks.

Ultimately, this work illustrates the potential of deep learning not only to detect marine litter effectively but to scale environmental monitoring solutions in a cost-effective and automated manner.

6. Individual Contributions

This project was a collaborative effort, with each team member contributing to distinct but complementary components of the research pipeline:

- **47999** - Focused on implementing and fine-tuning the **YOLOv8 Model** for underwater trash detection. This included dataset preprocessing, applying class-specific data augmentation strategies to address class imbalance, and training the model using both default and advanced hyperparameters. Additionally, led the evaluation and interpretation of YOLOv8's predictions
- **42502** - Focused on researching the project's related work, including various models and frameworks used for object detection. Researched the TensorFlow Object Detection API, then implemented the **EfficientDet model** including setting up the environment to bypass compatibility issues, pipeline configuration file, and the data preparation as well as evaluated the performance of EfficientDet.
- **43263** - Focused on the project's foundational aspects, including researching original **U-Net** papers, researched backbone-specific requirements and defining the overall code structure, Responsible for developing the model training procedures and evaluation metrics and compiling the U-net architecture and methodology documentation.
- **51348** - Handled the technical implementation of the **U-Net** model, integrating VGG16, ResNet50, and ConvNeXT backbones, addressing layer compatibility issues, especially for ConvNeXT. Contributions included debugging the core architecture and compiling results and interpretation.

References

Girshick, R. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014. doi: 10.1109/CVPR.2014.81.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2999–3007, 2017. doi: 10.1109/ICCV.2017.324.

Nande, S., Kulkarni, S., and Singh, M. Evaluation of u-net and resnet architectures for biomedical image segmentation. *International Journal of Engineering Applied Sciences and Technology*, 7(4):151–157, 2023.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016. doi: 10.1109/CVPR.2016.91.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science, Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351: 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Wu, Y. and Li, Q. Convnext embedded u-net for semantic segmentation in urban scenes of multi-scale targets. *Complex & Intelligent Systems*, 11:181, 2025. doi: 10.1007/s40747-024-01735-2.

Zhang, H., Liu, J., Shi, J., Li, X., and Liu, G. Semantic segmentation of high-resolution remote sensing images with improved u-net based on transfer learning. *International Journal of Computational Intelligence Systems*, 16 (1):181, 2023. doi: 10.1007/s44196-023-00364-w.