Marine Object Detetction and Classification by Integrating MLH-CNN with YOLOV8

MSc Research Project
MSc in Data Analytics

Ishita Kundu
Student ID: x22242091

School of Computing
National College of Ireland

Supervisor: Abid Yaqoob

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ishita Kundu
Student ID:	x22242091
Programme:	MSc in Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Abid Yaqoob
Submission Due Date:	12/08/2024
Project Title:	Marine Object Detetction and Classification by Integrating MLH-CNN with YOLOV8
Word Count:	6353
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Ishita Kundu
Date:	15th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Marine Object Detetction and Classification by Integrating MLH-CNN with YOLOV8

Ishita Kundu
x22242091

Abstract

The rapid increase in environmental pollution across the globe has a significant impact on marine life. Marine life is particularly impacted by harmful substances, such as plastic, rubber, metal, and natural wastes. Detecting marine objects accurately can help to reduce underwater debris. Numerous studies use deep-learning models to detect and classify marine objects, however achieving accurate results remains a challenge. This paper tries to resolve this issue by detecting marine objects and classifying them into different classes. The research proposes a modified YOLOv8 model, enhanced by integrating a Multi-Level Convolutional Neural Network in the YOLO architecture's backbone. The aim is to improve the efficiency of marine object detection. The model Proposed in this paper will be trained using the Seaclear Marine Debris Detection & Segmentation Dataset. This dataset has images of forty different classes of marine objects including marine debris. A comparative analysis will also be conducted between the YOLOv5, YOLOv8, and the proposed MLH-CNN YOLOv8 on the Seaclear Marine dataset. This comparative analysis will help determine the effectiveness of the proposed models and identify the best-performing model for object detection and classification. The ultimate goal of this research is to contribute to better marine debris management and reduction of marine pollutants through improved detection and categorization of waste materials.

1 Introduction

Environmental pollution causes damage to marine life and ecosystems. It started to grow rapidly after the industrial uprising. Most common debris such as plastic, glass, rubber, and metal are dumped into the sea or ocean and these pollute the marine environment. Removing Marine debris from the ocean is important for the safety of marine life. Various techniques have been proposed and implemented to control marine waste. However, pre-existing debris must be handled. The unexplored wastes cause major issues in the marine environment. Therefore, detecting and classifying unexplored debris is essential. Marine debris, primarily composed of plastics, causes severe threats to marine life and human health due to its environmental persistence Agamuthu et al. (2019). Previously some traditional methods like satellite sensors and sonar techniques were followed in underwater object detection. However, these methods have limitations in generating high-resolution images. Therefore, Autonomous underwater vehicles (AUVs) and Unmanned aerial vehicles (UAVs) were introduced to overcome the challenge and generate detailed images of marine debris. Machine learning and deep learning models showed impressive results in marine object detection Moniruzzaman et al. (2017). This research proposes YOLO models (YOLOv5 and YOLOv8) that can detect and classify objects efficiently. Integrating a Multi-Level Hybrid Convolutional Neural Network (MLH-CNN) with the YOLOv8 model might provide better accuracy in classification. A comparative analysis among the standard YOLOv5, YOLOv8, and proposed models will be conducted to identify the most effective model.

1.1 Research Background and motivation

This research aims to address the crucial problem of marine debris management such as detecting and classifying marine debris accurately. A dataset containing high-resolution images of underwater objects is essential to achieve this. However, existing object detection techniques face several challenges. Therefore the availability of publicly accessible data is very limited which is a major challenge in debris detection. Recent advancements in UAV and AUV technologies have improved the underwater image-capturing process. While numerous deep learning algorithms exist for marine object classification, they have certain limitations. Some deep learning methods provide high accuracy but cannot perform in real time. Few other algorithms have longer processing times despite good performance. Considering these drawbacks, achieving high accuracy in marine waste detection and classification remains an area to be improved.

As the globally increased environmental pollution has an impact on Marine life that causes Marine environmental pollution, it become a crucial issue to be addressed. Therefore, this study is conducted based on the major problem of marine pollution. Pollutants like plastic, glass, and metal land in underwater ecosystems and cause harmful effects on underwater species. Traditional methods of capturing marine objects were not efficient. Currently, technologies like UAVs and AUVs provide high-quality images that help implement deep-learning modules for debris classification. Accurate classification of detected objects is crucial for managing marine waste effectively. Existing researchers employing various deep learning methods continue to struggle with the efficient detection and classification of marine objects. Therefore, this research aims to use YOLO models which are already proven very effective in object detection Redmon et al. (2016) and also integrate a Multi-Level Hybrid Convolutional Neural Network (MLH-CNN) with the YOLOv8 algorithm to make the detection efficient.

1.2 Research Question

The research question for this paper is
"How Yolov8 can be extended for marine object detection and classification?"
This paper has proposed implementing YOLOv5 and YOLOv8 and integrating MLH-CNN with the YOLOv8 model. A comparative analysis will be done among the YOLOv5, YOLOv8, and the integrated MLH-CNN YOLOv8. Considering the efficient performance of YOLO, the model integrating the MLH-CNN with YOLO can achieve more accuracy in object detection and classification.

1.3 Research Objective

This study is conducted to create a robust system for accurate marine object detection and classification. YOLOv8 and YOLOv5 will be implemented initially to achieve this.

Another custom YOLOv8 model, which integrates a multi-level hybrid convolutional neural network (MLH-CNN), will be implemented to improve accuracy. This project aims to improve marine debris management and reduce marine pollution by enhancing the accurate object detection process. Ultimately, this research will try to overcome existing challenges in object detection techniques such as high-resolution images and real-time debris detection.

Section 1 describes the research background, motivation, and research question. The Related work 2 section consists of the academic review of previous papers related to the research question and research background. This section provides an idea of the current status of the research topic and highlights the gaps. It has different subsections starting from the broader aspect of the research logically narrowed down to the research topic. Section 3 includes the research methods, approach to be followed, dataset sources, data collection, data preparation, and model training. Section 4 outlines the detailed design specifications of the project. It includes the architecture of the proposed standard YOLO models and the integration of MLH-CNN with the YOLOv8 model. Section 5 includes the implementation technique of the proposed methods. This section includes the detailed steps taken to clone the repository, set up the environment, and split it into train and test. Section 6 discusses the results of the research. It presents the evaluation metrics and conducts the performance assessment of the proposed system. Section7 summarizes the key findings of the research and its contributions. It discusses the potential areas for future improvement and further research.

2 Related Work

This section examines the background of marine debris, its continuous increase, and its major environmental impact. This section also shows the analysis of the existing models employed by the researchers to detect and classify marine debris.

2.1 Impact of Marine debris on ecosystem

As global environmental pollution including marine pollution continues to rise, several research papers are there regarding the of marine debris impact on the environment. The dominance of microplastics presented by Coyle et al. (2020) is a significant concern for the marine ecosystem. It is important to understand the distribution processes of microplastics to address their ecological impact. Although polymer density influences microplastic distribution, this paper shows that the interaction with marine species is more effective for the presence of buoyant microplastics at depth. It ultimately leads to bioaccumulation. This paper provided details regarding the origin of plastic and what percentage of this ends up in the ocean. Microplastic is present in 92% of the ocean's plastic particles and it is highlighted by the European Union Marine Strategy Framework Directive as a concern due to their impact on marine life.

Marine litter's effect on marine organisms' life proposed by Valente et al. (2020) showed how harmful it is to deep-water habitats and how it affects biodiversity and the ecosystem. This study used a modified protocol for monitoring litter consumption by loggerhead sea turtles. This was done to assess the feasibility of elasmobranch species to monitor macro-litter absorption in marine environments. 122 species including 7 elasmobranch species were assessed in this study and a very small number of cases were revealed with macro-litter ingestion. As the litter stays in the body of the deep-sea creatures shorter than the food, litter detection is challenging. The study highlighted the possibility of litter ingestion by deep-sea creatures which might be missing and showed the need for monitoring underwater litter.

A comprehensive study by Courtene-Jones et al. (2019) was conducted to investigate the microplastics present in deep-sea environments. It mainly focused on microplastic ingestion by two deep-sea species, Ophiomusium lymani and Hymenaster pellucidus. Samples for the study were taken over four decades. Although mass plastic production started in the 1940s-50s, limited data is available on the presence of microplastics in deep-sea species. This study found microplastic ingestion by the 45% of the total examined species. A total of eight types of polymers were found where polyamide and polyester were the highest. This paper highlights the constant presence of microplastics in the underwater ecosystem over the years and their rapid spread in deep-sea environments. The data and findings from this study can help to understand the future effects of microplastics in marine ecosystems.

The presence of microfibers in seawater and sediments is explained by Gago et al. (2018). It highlighted the extensive presence of microfibers and the variability in the sampling method. Microfibers are mainly made from synthetic materials like polypropylene and polyethylene. These are detected with high concentrations in the ocean, particularly in harbors. These synthetic microfibers are a threat to the marine environment because of their small size which makes the ingestion probability by different species very high. This study also discussed the global distribution and challenges in accurately measuring microfibers.

Overall, these papers highlighted the widespread presence and impact of plastic, particularly microplastics, across marine ecosystems. While Coyle et al. (2020) showed the impact of microplastics, Valente et al. (2020) focused on the challenges of monitoring macro-litter ingestion by deep-sea species. A long-term perspective on microplastics provided by Courtene-Jones et al. (2019) and Gago et al. (2018) reviewed the presence of microfibers in marine sediments and waters. In conclusion, all the studies described the need for improved techniques to identify and handle debris in marine ecosystems.

2.2 Techniques for detecting marine objects

The presence of significant amounts of marine debris affects the marine environment and it became crucial to detect and classify debris properly. Initially, marine divers played a crucial role in studying the underwater environment and observing marine life. It provided important information on the underwater environment and confirmed the presence of debris. In response to the increasing debris Kikaki et al. (2024) discussed the challenges of object detection and tracking. While most papers focus on particular pollutants and use classification to differentiate plastic from non-plastic, this paper proposed a holistic approach using remote sensing to detect marine pollutants. This approach included data collection, integration of satellite imagery, and aerial photography.

The challenges of object detection in acoustic image segmentation are addressed by Priyadharsini and Sharmila (2019). The main difficulty was to differentiate between the seafloor, sediments, and objects due to low resolution and specific image characteristics. Traditional segmentation methods, like region-based and edge-based techniques, were not accurate in object detection. This study proposed an edge-based segmentation technique using morphological operations to detect edges followed by an object tracing algorithm. Images used in the study were Real-time images taken from an Edgetech 4125 side-scan sonar. Image processing was done with Wiener and Median filters, and a morphological gradient highlighted the boundary of the object. Finally, Moore's object tracing algorithm was implemented to detect objects.

Satellite image and side-scan sonar techniques were used to address the challenges of the existing marine object detection techniques. Sánchez-Ferrer et al. (2023) proposed the use of AUVs for identifying and classifying marine debris. This method requires minimum human intervention and supervision, so it became popular. AUVs use state-of-the-art techniques based on deep learning models to perform the object detection task. This paper explored the Mask Region-based Convolutional Neural Networks framework to locate and identify objects. This paper also checked the possibility of using synthetic data. The object detection methods tried to predict boundary boxes so that associated labels could be addressed by this.

Although autonomous underwater vehicles are useful in underwater object detection, they have a few limitations. The challenges faced by AUVs that rely on visual sensors are addressed by Fabbri et al. (2018). The performance was affected due to factors like light refraction, color distortion, and absorption in underwater environments. This study proposed using Generative Adversarial Networks (GANs) to improve the image quality captured underwater by restoring images. This study also proposed to improve the accuracy of vision-driven tasks such as segmentation, tracking, and classification. The images processed using this method look more visually appropriate and help in enhancing the performance of a diver tracking algorithm.

All the papers tried to address the challenges present in marine debris detection using various methodologies. A holistic approach with remote sensing and satellite imagery was introduced by Kikaki et al. (2024). Sonar images were used in Priyadharsini and Sharmila (2019) for improving image segmentation. To address the challenges faced by satellite and sensor images, Sánchez-Ferrer et al. (2023) proposed the use of AUVs with deep learning techniques. Finally, the enhancement of AUVs' visual performance through Generative Adversarial Networks (GANs) was proposed by Fabbri et al. (2018) to address challenges and improve accuracy.

2.3 Deep Learning Methods for Classifying Marine Objects

After detecting objects, classification of the objects is essential to manage debris. AUVs were used as an object detection technique in the research paper Fulton et al. (2019). This paper explored the deep-learning models for detecting and classifying marine debris that causes harm to marine environments. Four deep learning algorithms were chosen here such as YOLOv2, Tiny-YOLO, Faster RCNN with Inception v2, and SSD with MobileNet v2. After model evaluation, it was observed that YOLOv2 provided good results, accuracy, and fast runtime. Tiny-YOLO offered the same mAP values as YOLOv2 and also offered faster processing speeds. Faster RCNN and SSD with MobileNet v2 performed better and provided better accuracy, however their processing times were high. Therefore, it was observed that Tiny-YOLO and YOLOv2 provided a balanced result for speed and accuracy. Considering the performance of YOLO models, high accuracy might be achieved by YOLO if a higher YOLO version can be used in object detection.

Other research papers are explored here to further check the performance of updated YOLO models. Ship detection method using PP-YOLO to enhance the accuracy and speed of detecting marine objects was proposed by Liu et al. (2021). Remote sensing images were used in this research, and it evaluated the proposed model against YOLOv3 and YOLOv4 using Synthetic Aperture Radar (SAR) and HRSC2016 datasets. The result of the study demonstrated that PP-YOLO provided better detection accuracy and speed. PP-YOLO used ResNet50d as its backbone by replacing the DarkNet-53 and used MixUp as a data augmentation technique to improve generalization. The results showed that the modified higher-level YOLO model performed better than the standard models.

A YOLOv5-based method underwater object detection technique was proposed by Xia and Tan (2023) to improve the detection speeds and resolve the issue with complex environments. The study introduced the YOLOv5-ShearC3 module. Model parameters and network layers were reduced to enhance detection speed by introducing the ShearC3 network. The proposed model achieved an accuracy of 84.3% in object detection which is 1.2% higher than the standard YOLOv5. The detection speed also increased by 10.16%. This proposed model showed better performance in underwater object detection for accuracy and speed.

Updated versions of YOLO convolutional neural network architecture were introduced by R and M (2023). It focused on enhancing marine object identification to manage marine debris. This study showed the importance of image preprocessing to improve image quality by addressing changes such as low brightness, insufficient light, and light absorption under the sea. This research evaluated YOLO models, mainly YOLOv3, YOLOv5, and YOLOv7 to identify seven different marine animals. Results and evaluation metrics showed that the YOLOv7 performed better compared to the other two YOLO models. The detection accuracy was high for YOLOv7 and achieved mAP of 0.82. Finally, this study highlighted the capability of YOLO models in object classification and further enhancement in YOLO models could provide better results.

These papers compared various versions of the YOLO model for marine object detection and showed how it performed better than traditional methods. Each paper showed unique enhancements on different YOLO models to improve object detection accuracy. The steady performance of YOLOv2 and Tiny-YOLO regarding speed and accuracy was highlighted by Fulton et al. (2019). PP-YOLO performed better than YOLOv3 and YOLOv4 in detection accuracy and speed can be concluded from Liu et al. (2021). YOLOv5 model that introduced the ShearC3 and PixelShuffle modules to improve detection speed and image resolution was introduced by Xia and Tan (2023). Comparison between different YOLO models was conducted by R and M (2023) and showed YOLOv7's superior performance in object identification. Therefore, this paper used the YOLOv8 model integrated with the MLH-CCN and compared it with the standard YOLOv8 to check the model performance.

2.4 MLH-CCN integration with deep learning models

The performance evaluation of the combined MLH-CCN model with deep learning methods was conducted by Funch et al. (2021). It introduced a machine-learning method for classifying glass and metal in consumer trash bags using sound recordings and a metal detector. The aim was to improve waste sorting quality. A custom-built test rig simulated a waste collection truck to generate datasets and this dataset was used in training convolutional neural networks. This model achieved up to 98% accuracy in the classification. Three CNN models using a different number of layers and parameters were evaluated in this study. The first model had five convolutional layers, model 2 was developed with two additional convolutional and fully connected layers, and model 3 was developed using a rectangular kernel in the first layer. Model 1 has provided the highest accuracy here.

A Multilayer Hybrid System (MHS) for automatic waste sorting in urban areas was introduced by Chu et al. (2018). It deployed high-resolution images and sensor data to enhance classification accuracy. In this study, the MHS used a Convolutional Neural Network (CNN) algorithm for feature extraction and a Multilayer Perceptron (MLP) to combine image and other features. The MHS significantly performed better than CNN based approach and achieved classification accuracies of more than 90%. This model could effectively address the challenges in waste classification such as high manual sorting costs and technological limitations.

The multilayer hybrid convolution neural network on the waste dataset was proposed in the paper Shi et al. (2021). This method uses vggNet with fewer parameters and a simpler structure. By adjusting network modules and channels, the model's performance was enhanced in waste classification. The proposed model achieved accuracy of up to 92.6% on the TrashNet dataset which was better than other existing methods by over 4%. This proposed model produced excellent performance accuracy with a simpler structure. Therefore MLH-CNN can be used with other models for better performance.

All the studies proposed different deep-learning approaches to improve the accuracy of waste classification. A CNN model was used by Funch et al. (2021) achieving up to 98% accuracy with five convolutional layers. A Multilayer Hybrid System (MHS) using CNN and MLP achieved over 90% accuracy in Chu et al. (2018). An MLH-CNN inspired by VggNet was proposed by Shi et al. (2021) and provided 92.6% accuracy on the TrashNet dataset. These studies highlighted the excellent performance of MLH-CNN in waste detection. Therefore, this technique can be integrated with other machine-learning models for accurate results.

3 Methodology

This section discusses the complete research process of this project in detail. This step includes data collection, preparation, visualization, and implementation in a systematic order so that the findings can be validated. Figure 1 explains the research methodology followed in this project.

3.1 Data Collection

The dataset used for this project is the SeaClear Marine Debris Dataset and it is sourced from Đuraš et al. (2024). This is an open data repository for science, engineering, and design. It provides a platform for researchers to store and share research data and ensures data is accessible, reusable, and preserved for the long term. It contains 8610 images annotated for object detection and 40 object categories. Figure 2 shows sample images from different categories. This dataset includes marine debris as well as animals and plants. The images for this dataset were captured using remotely operated vehicles (ROVs) for the SeaClear project. These images were taken from various locations, including Bistrina, Jakljan, Lokrum, and Slano in Croatia, and Marseille in France. The annotations are in COCO format in the (.json) file, and images are organized in different folders. All images are in 1920x1080 pixels ensuring uniformity across the dataset.

- **Image dataset and JSON**- 4TU.ResearchData

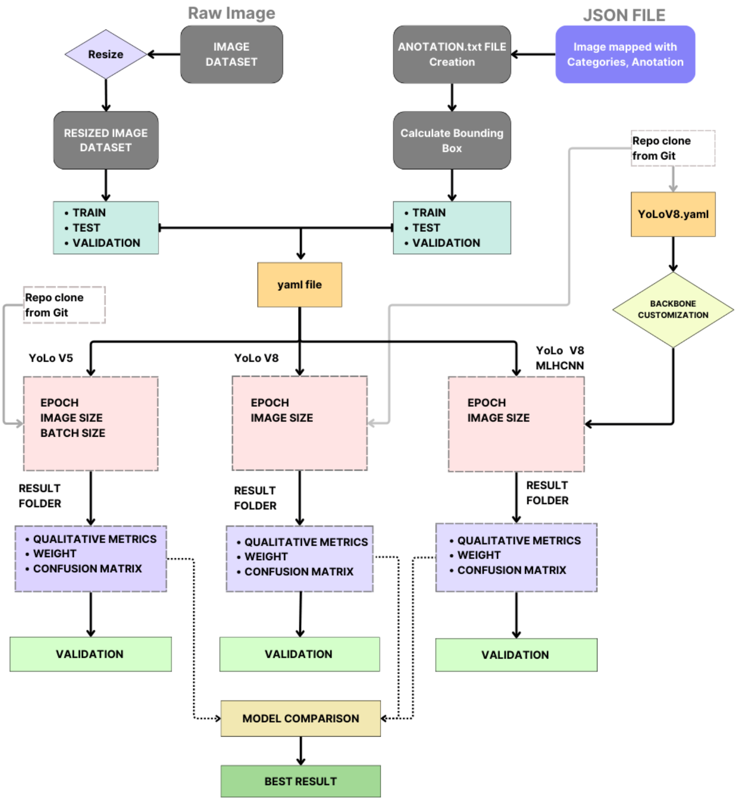

Figure 1: Proposed methodology

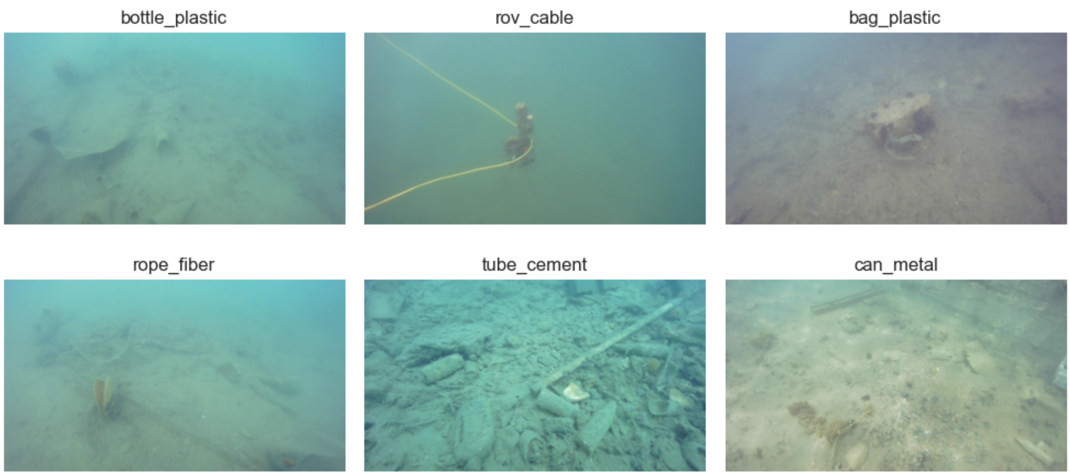

Figure 2: Sample images from different categories

3.2 Data Preparation

3.2.1 Resizing:

This step includes the image resizing from the specified directory to a resolution of 800x800 pixels to standardize input dimensions. This step identifies the .jpg extension and reads each image using OpenCV. This step is important for maintaining consistency across the training and evaluation process. Resizing helps to manage computational resources and ensures consistent object detection. Resizing images can make the algorithm perform quickly and reduce memory usage.

3.2.2 Annotation Conversion:

This step includes the process of creating an annotation.txt file from a JSON file present in the Seaclear Marine Debris Dataset. After extracting data, normalization of bounding box coordinates for each image annotation is performed. Then it gets converted into a format that is compatible with the YOLOv5 and YOLOv8 models. This step involves mapping category IDs and transforming bounding box coordinates from the top-left corner to the center format. Then it normalizes the bounding box values relative to the image dimensions. Finally, the processed data is saved into text files for model training.

3.2.3 Data Splitting:

This section organizes the dataset of images and their corresponding annotation files into training, validation, and test sets using an 80-10-10 ratio. This creates directories for each train, val, and test sets. The training set is used to train the models, the validation set is used to validate the model and the test dataset is used to check the final model performance. This step ensures that the models are trained on a diverse set of images and evaluated on other data.

3.2.4 YAML Configuration:

This step includes YAML file creation for each model to specify dataset paths, class labels, and training parameters. The YAML config file includes paths to the training, validation, and test datasets. It also includes the number of categories and a mapping of category IDs to names. This configuration file is used for the training of the YOLO model to provide necessary directory paths and class information.

3.3 Data Visualization

3.3.1 Bounding Box

This step uses visualization tools to overlay bounding boxes on sample images to verify the accuracy of annotations. This step ensures that the objects in the image data are correctly annotated and that the bounding boxes are properly aligned with the actual objects in the images. Figure 3 shows the proper bounding box creation for all the objects present in the image.

Figure 3: Image with bounding box

3.3.2 Class Distribution

Histograms were generated in this step to visualize the distribution of different object classes and sizes in the dataset. This analysis helps identify any class imbalances in the dataset, such as overrepresenting or underrepresenting classes, which can impact model training and evaluation. Figure 4 shows the class distribution for all the categories and class imbalance can be observed here.

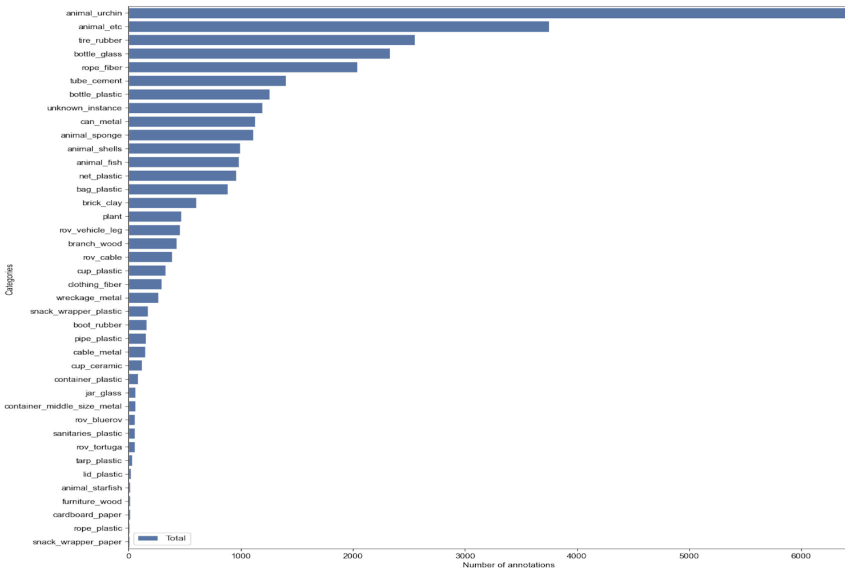

Figure 4: Annotations per category

3.4 Training

3.4.1 YOLOv5 and YOLOv8:

This section includes YOLOv5 and YOLOv8 model training. Configurations for YOLOv5 and YOLOv8 include 20 epochs using image sizes of 215, and batch sizes of 8. The training process includes running the models and monitoring performance metrics such as loss, accuracy, precision, and recall. The YOLOv5 and YOLOv8 training is done with the respective scripts cloned from the Git repository. Model training is done with the pre-processed data and generated yaml file. Training scripts are used with modifications made to make the prediction accurate.

- **YoLov5** - Git
- **YoLov8** - Git

3.4.2 YOLOv8 with MLH-CNN:

This section includes the custom model YOLOv8 with MLH-CNN which is trained with similar configurations but with additional hierarchical feature extraction layers. The training process involves the integration of MLH-CNN into the YOLOv8 architecture. It requires adjustments to the model's backbone to enhance feature extraction. The custom setup aims to improve detection accuracy, especially for objects of varying sizes and complexities.

4 Design Specification

4.1 YoLov5

YOLOv5 is a single-stage object detection model that efficiently detects objects in a single forward pass through the network. Its architecture consists of a backbone, a neck, and a head which can be referred from Xu et al. (2021). This design allows for faster and more accurate object detection.

- **Backbone:**

  The backbone of the YOLOv5 is designed CSPDarknet53 backbone, which uses Cross-Stage Partial connections to enhance the efficiency of feature extraction by reducing redundancy and increasing gradient flow. This architecture helps the model to extract useful features and maintain computational efficiency.

- **Neck:**

  The Path Aggregation Network (PANet) in YOLOv5 is used to combine features from different layers of the backbone by enabling feature fusion across scales. PANet helps to fetch multi-scale features and improves the ability to detect objects of various sizes by utilizing top-down and bottom-up pathways for feature extraction.

- **Head:**

  The detection head in YOLOv5 predicts bounding box coordinates, and class probabilities for each detected object. It utilizes anchor boxes to adjust various object sizes and shapes so that the model's detection accuracy can be increased. This approach improves the model efficiency by aligning predictions more closely with the true shapes and sizes of objects in the image.

4.2 YoLo v8

Figure 5 shows the YOLOv8 architecture which is the enhancement of the YOLOv5 architecture by incorporating advanced features for improved multi-scale feature aggregation. This allows YOLOv8 to achieve higher accuracy and faster detection speeds compared to YOLOv5. Additionally, YOLOv8 introduced different flexible model variants that can be used depending on specific computational resources and requirements. The YOLOv8 model architecture can be referred to Sharma et al. (2023).

- **Backbone:**

  YOLOv8 utilizes a variety of backbones including an enhanced version of CSP-Darknet53 and EfficientNet. The backbone is responsible for extracting meaningful features from the input. It captures simple patterns in the initial layers, such as edges and textures.

- **Neck:**

  The neck is the bridge between the backbone and head, and it performs a feature fusion operation. Neck collects feature maps from different stages of the backbone. YOLOv8 also employs the Path Aggregation Network (PANet) to aggregate multi-scale features and enhance the model's ability to detect objects of various sizes.

- **Head:**

  The detection head in YOLOv8 has few enhancements compared to YOLOv5. It supports both anchor-based and anchor-free detection, so the model's adaptability increases in object detection with different sizes and shapes. It is refined to improve bounding box predictions and class probabilities.

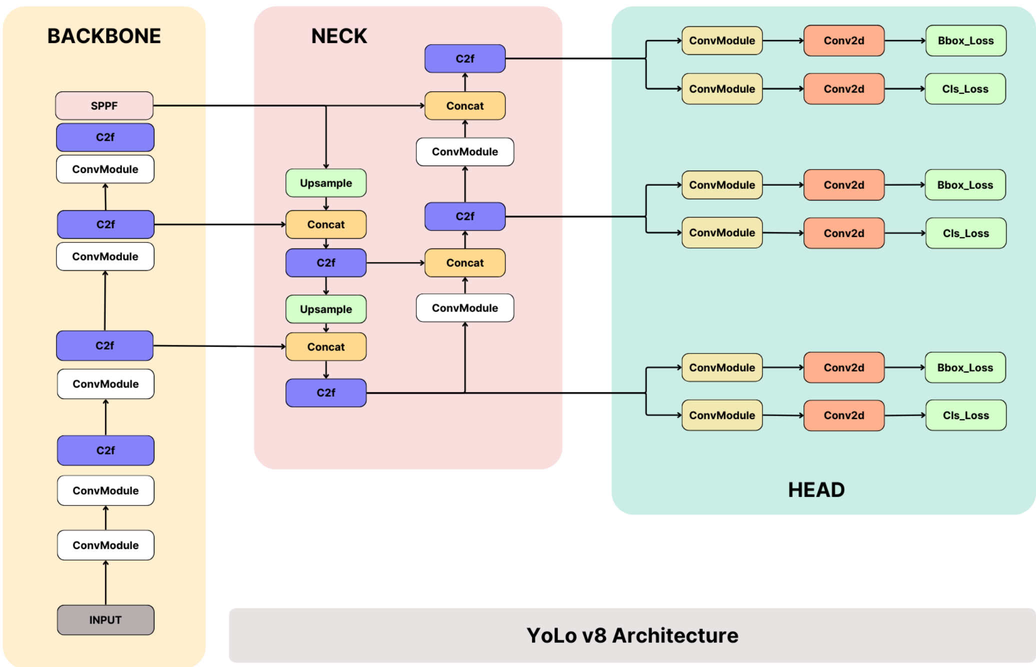

Figure 5: YOLOv8 Model Architecture

4.3 YoLov8 with MLH-CNN

The YOLOv8 model is integrated with Multi-Level Hierarchical Convolutional Neural Networks to capture features at different scales. Figure 6 shows this integration tries to handle variations in object sizes.

- **Hierarchical Feature Learning:**

  MLH-CNN incorporates a multi-level hierarchical structure to enhance the model's performance by learning features at different levels of abstraction. Modifying specific YOLOv8 variant (YOLOv8s) backbone can capture complex features and incorporate Cross-Stage Partial connections. The hierarchical approach also can improve generalization across different object types.

- **Modified Backbone**

  The modified YOLOv8 architecture includes hierarchical layers in the backbone by using BottleneckCSP. This helps the neck to enhance feature extraction and aggregation. Additionally, the Spatial Pyramid Pooling (SPP) layer aggregates features from various spatial scales. This structure allows the model to capture detailed features so that overall detection performance can be improved.

- **Neck and Detection Head**

  The detection head utilizes the hierarchical features from the backbone to refine bounding box predictions and class probabilities. This integration can enhance the model's accuracy in marine object detection.

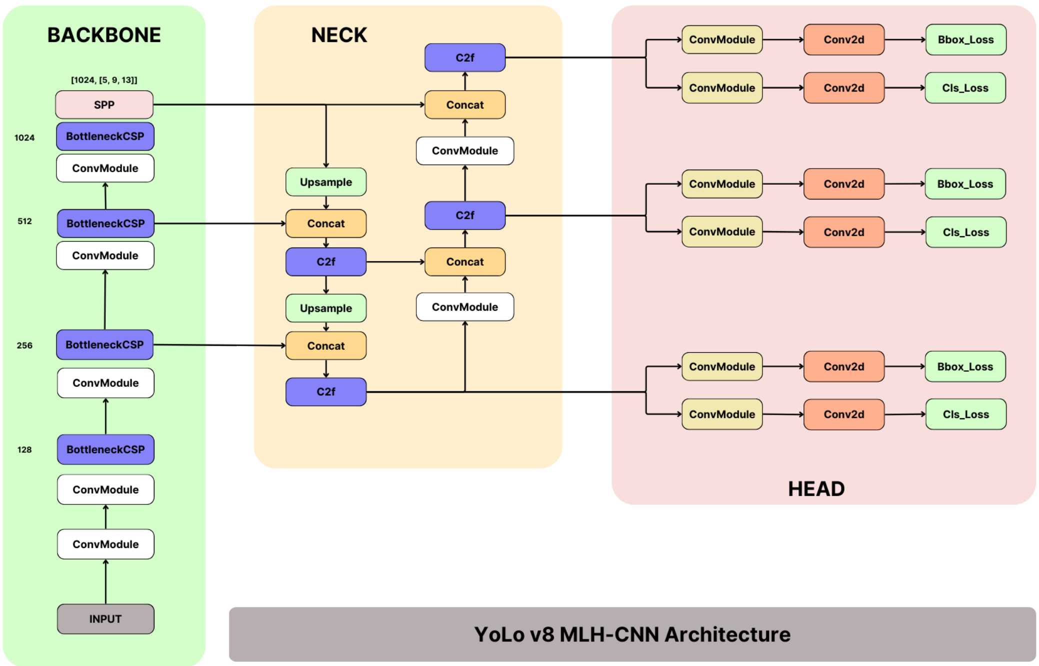

Figure 6: YOLOv8 MLH-CNN Model Architecture

5 Implementation

This section provides an overview of all the implementation steps of the project such as pre-requisite cloning, environment configuration, model training, the model's output, tools, and techniques.

5.1 Setup

5.1.1 Repository Cloning:

YOLOv5 and YOLOv8 models were used after cloning from their respective Ultralytics folder of GitHub repositories. The MLH-CNN was implemented on the YOLOv8 backbone that is cloned from GitHub.

5.1.2 Environment Configuration:

The development environment was set up with Python 3.11, and essential libraries such as PyTorch 1.9.0, OpenCV, pandas, seaborn, and NumPy were installed. Dependency management was handled using conda to ensure a consistent setup. No external GPU was used, and model training and evaluation were performed by adjusting the batch sizes.

5.2 Models

5.2.1 YOLOv5 and YOLOv8:

After the data collection data preparation is done for the YOLOv5 and YOLOv8 models implementation. Data preparation includes creating 'labels' and 'images' directories, bounding box, and annotation creation. After validating the correct Bbox creation, the dataset will be split into train, test, and val within the labels and images folders. Before starting the model training, a custom YAML file is generated using pre-processed data. The configurations included 20 epochs with an image size of 215 and a batch size of 8. The YOLOv5 and YOLOv8 repositories are cloned from their respective Git repositories. YOLOv5s and YOLOv8s files were chosen for the model training depending on the file size, complexity, and runtime. For YOLOv5, hyperparameters included a learning rate (lr0) of 0.01, final learning rate (lrf) of 0.01, momentum of 0.937, and weight decay of 0.0005. For YOLOv8, the AdamW optimizer was used with a learning rate (lr) of 0.000227 and momentum of 0.9. Throughout the training process, key performance metrics such as loss, precision, and recall were monitored in the CSV file generated in the local system to ensure optimal model performance. For the model prediction, generated weights after model training will be used.

5.2.2 YOLOv8 with MLH-CNN:

Implementing YOLOv8 with MLH-CNN required modifying the backbone YOLOv8 architecture to incorporate MLH-CNN layers. Starting from the data preparation, Bbox creation, and data splitting to yaml file creation, these steps will be followed in the same order as followed during YOLOv8 training. The YOLOv8.yaml file needs to be modified where the backbone of YOLOv8 is defined. BottleneckCSP and the Spatial Pyramid Pooling layers were added to the backbone by modifying the existing layers. This modified YOLOv8.yaml was used to train the model with 20 epochs with an image size of 215. The AdamW optimizer was used here with a learning rate (lr) of 0.000227 and momentum of 0.9 same as yolov8. Backbone layers were modified in a way so that they could be compatible with the standard YOLOv8.

5.3 Tools used

The implementation of the model for this project involved several libraries and software.

- **Platform:**

  Jupyter Notebook (Python): It provides an environment for testing the code, implementing models, and visualizing the results. The Jupyter Notebook is also utilized for complete documentation, showing the implementation process and results effectively.

- **Libraries used:**

  PyTorch: It was utilized for training the YOLOv5 and YOLOv8 models considering its flexible and powerful building and optimization capability. OpenCV: It was used for image processing tasks such as resizing and visualizing images with bounding boxes to ensure the accuracy of annotations. NumPy: It was used for numerical operations and handling arrays. It provided support for data manipulation and preparation tasks. Yaml: It was used to create the yaml configuration file to use in model training

- **Visualization Tools:**

  Matplotlib, and Seaborn for creating plots and charts to visualize data and results. Matplotlib and Seaborn were used for generating a variety of plots and charts. It helped to create category-wise data distributions and model results, thus providing comprehensive data analysis.

5.4 Hardware used

The project was completed using the hardware that includes a 4 core 8 threads coupled with an inbuilt Intel UHD Graphics 620 processor, 512 gigabytes of solid state drive and 8 gigabytes of RAM.

6 Evaluation

This section details a complete analysis of the result obtained. The model's performance is evaluated based on a combination of quantitative analysis to provide a complete assessment of the report.

6.1 Quantitative Analysis

This section evaluates the model's performance metrics that have been generated after the model training. This includes precision, recall, F1 score, and model loss. The analysis provides insights into three different models and compares their performance to choose the most effective one.

6.1.1 F1 score

The F1 score is crucial for assessing the model's effectiveness. This F1-score is calculated by taking the average of precision and recall. The three different F1 curves represent the performance of three different models in Figure 7. The F1 score for YOLOv5 shows a peak of 0.47 which indicates a moderate balance between precision and recall. Next, it improves for YOLOv8 significantly with a peak F1 score of 0.60. It reflects better accuracy and consistency across all classes. After incorporating MLH-CNN with YOLOv8 backbone it drops to an F1 score of 0.14. This result indicated that hierarchical feature extraction in MLH-CNN may have introduced complexities that impacted the model's balance.

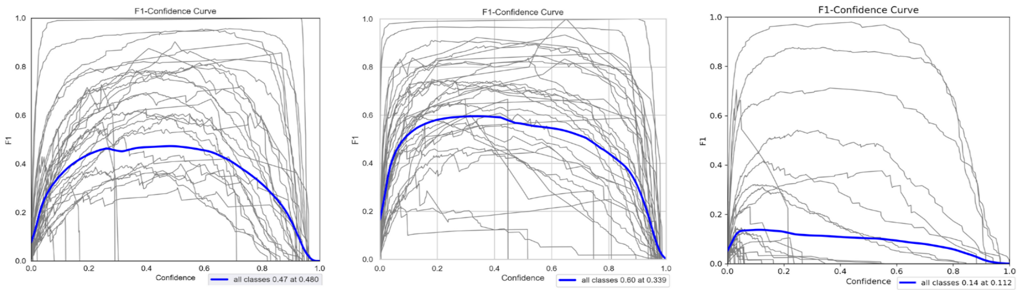

Figure 7: F1 Score for all Models

6.1.2 Precision & Recall

Precision refers to the proportion of images that are correctly predicted by the model to be true. This value increases with each epoch during the model training. As this value increases, the reliability of the model's performance also improves. The precision is calculated using the equation True Positive values and False Positive values. It measures the accuracy of the positive predictions made by the model. Based on the performance metrics shown in Figure 8, Figure9 and Figure10 generated for three models YOLOv5, YOLOv8, and YOLOv8 with MLH-CNN it can be said that YOLOv5 and YOLOv8 show constant improvement in precision. While YOLOv8 shows some initial fluctuations it achieved a precision of 0.83 in the last epoch which is the highest among the three models. After implementing MLH-CNN in yolov8, it shows a precision of 0.53 which is lesser than the standard models. When the MLH CNN with YOLOv8 model training started, precision values were good in a few epochs then it decreased again at the last epoch. Fluctuation is observed here for the MLH CNN with YOLOv8 model training.

The Recall rate refers to the proportion of images in the dataset that the model has correctly predicted. It is the ratio of true positive predictions to the total number of true positives plus false negative predictions. A higher recall rate indicates that the model's accurate prediction rate is high. As for the precision, Figure 8, Figure 9 and Figure 10 show the recall is also high for YOLOv8 which is 0.53, and YOLOv5 which is 0.43. Recall increases for both models ensuring the model's effective performance. The YOLOv8 with MLH-CNN starts with lower values and then gradually improves. It suggests that training on more epochs might have increased the recall value. Based on the precision, and recall values, YOLOv8 performed more effectively.

6.1.3 Log Loss

Training Loss refers to the discrepancy between the actual target values and the model's predicted values during the model training. From the loss metric analysis in Figure 8, Figure 9, and Figure 10, it can be seen that YOLOv8 with MLH-CNN achieved the lowest loss of 0.32. In comparison, YOLOv8 showed an average training loss of 0.35 and YOLOv5 had the highest losses of average 0.45 in training.

6.1.4 YOLOv5 Result

YOLOv5 comprehensive model result can be checked in Figure8. It includes training and validation loss, precision, recall, and mAP.

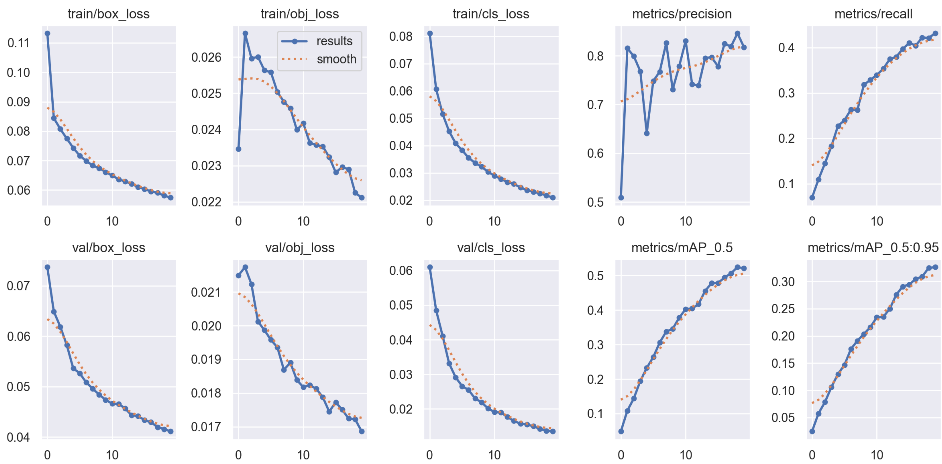

Figure 8: Yolov5 Model Result

6.1.5 YOLOv8 Result

Figure 9 explains the YOLOv8 model's comprehensive result. It also includes training and validation loss, precision, recall, and mAP.

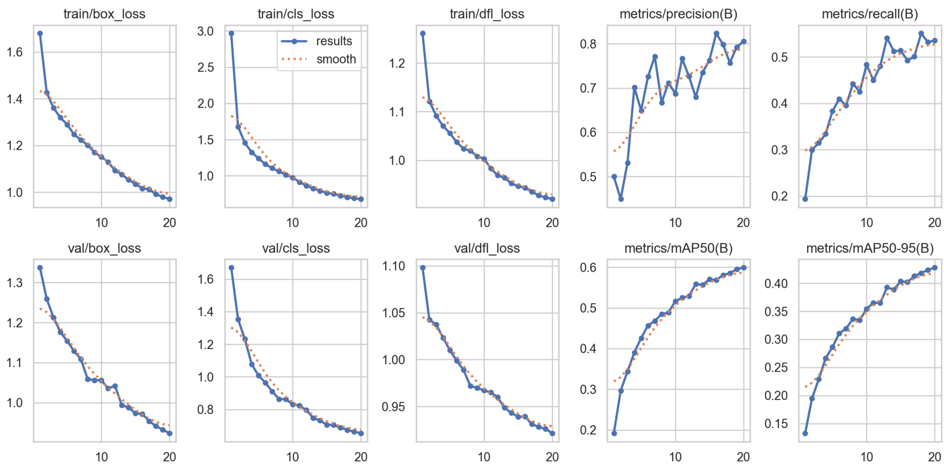

Figure 9: Yolov8 Model Result

6.1.6 YOLOv8 MLH CNN Result

Figure 10 shows the YOLOv8 MLH CNN model's comprehensive result. It shows precision, recall, loss, and mAP.

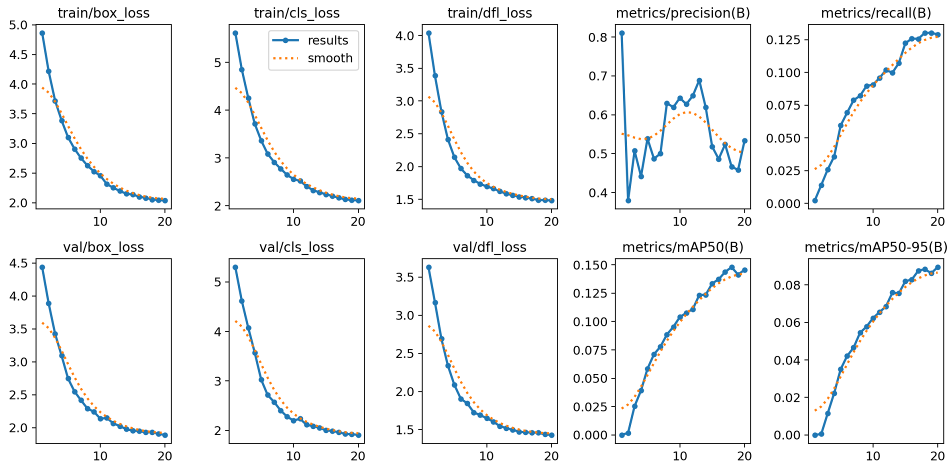

Figure 10: Yolov8 MLH CNN Model Result

6.2 Prediction

In this research, the weights generated from trained YOLOv, YOLOv8, and YOLOv8 MLH-CNN models were utilized to predict objects from the test dataset. Predictions were made by processing images to extract features. This process included predicting bounding boxes, confidence scores, and class probabilities. These predictions are then evaluated for the models' performance. For comparative study, this step is crucial for assessing the models' accuracy. Prediction of objects for different images can be seen in Figure 11 using three different models. All the models could detect objects however there are overlaps in the prediction by YOLOv8 MLH-CNN. YOLOv8 could predict the objects properly without any overlaps for the same image used in YOLOv8 MLH-CNN. YOLOv5 model also could predict the objects correctly however few detections are not relevant. Therefore, the prediction capability is best for YOLOv8 among the three models.

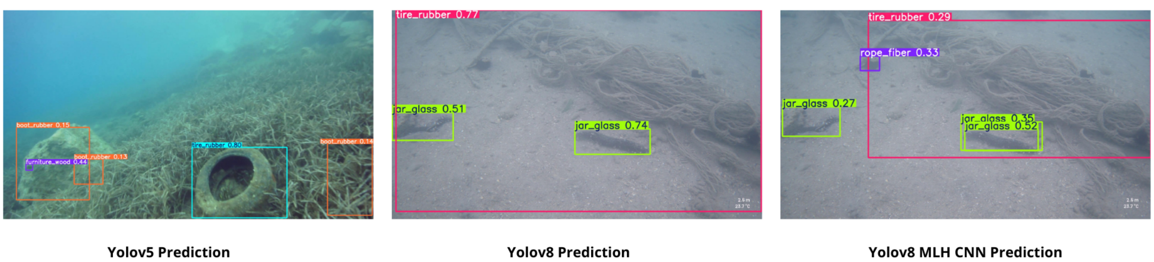

Figure 11: Model Prediction

6.3 Discussion

The evaluation details show that the proposed model could not achieve accuracy as the other two standard models. It highlights the effectiveness of the standard YOLOv5 and YOLOv8 models in detecting marine debris. YOLOv8 showing improved accuracy due to its advanced architecture. The integration of the MLH-CNN added more hierarchy to the feature extraction. The evaluation results indicate that object detection can be done by modifying standard YOLO models however it might require a few more adjustments to achieve better accuracy. These results show that YOLO models can be a useful option in real-time object detection.

Model Comparison

Model	Precision	Recall	mAP	F1 Score
Yolov5	0.81	0.43	0.52	0.47
Yolov8	0.83	0.53	0.59	0.60
Yolov8 MLH-CNN	0.53	0.12	0.14	0.14

7 Conclusion and Future Work

The primary objective of the project was to develop a system for accurate marine object detection and classification. This step was achieved with the implementation of three image classification models: YoLov5, YoLov8, and YoLov8 with MLH-CNN. A comparison of the three models concluded that YoLov8 performed the best out of the three models. YoLov8 achieved higher accuracy with respect to the other models and is capable of accurately classifying marine underwater objects. The model showed a high precision and recall of 0.83 and 0.53 by YOLOv8. Figure 8, Figure 9 and Figure 10 determine the model's capabilities.

Future work from the research findings can open new avenues in research. The use of additional data augmentation and synthetic data generation for more diverse environments can improve the model's accuracy and add to its robustness. Implementing YOLO models from scratch instead of modifying the backbone can improve the accuracy. Dynamic environment handling can be implemented to enable models to retain accuracy even when motion is present. This can also open windows for real-time image processing and model deployment. Real-time detection using edge computing devices can preprocess data from robots and sensors, enabling faster real-time assessment. Cross-domain adaption can also to implemented to make the model more robust by testing the model in different geographical locations and various underwater conditions. This model can also be implemented in different scenarios for object detection and is strictly not limited to marine environments. Self-learning and semi-supervised learning can be implemented to improve the model accuracy with less labeled data. The use of generative adversarial networks (GANs) to generate synthetic data to handle class imbalance can further impact the performance. Integration of additional data sources such as acoustic data, sonar data, etc. can enhance the object detection capabilities.

The research can impact environmental conservation by tracking marine life, assessing the health of coral reefs, and detecting and monitoring marine pollution. The research can be leveraged for studying marine ecosystems. The research can aid in technological advancements by contributing to artificial intelligence and computer vision. This collaboration can lead to technological advancements and impact societal benefits.

References

Agamuthu, P., Mehran, S., Norkhairah, A. and Norkhairiyah, A. (2019). Marine debris: A review of impacts and global initiatives, Waste Management & Research **37**(10): 987–1002. PMID: 31084415.
URL: *https://doi.org/10.1177/0734242X19845041*

Chu, Y., Huang, C., Xie, X., Tan, B., Kamal, S., Xiong, X. and Seeboonruang, U. (2018). Multilayer hybrid deep-learning method for waste classification and recycling, Intell. Neuroscience **2018**.
URL: *https://doi.org/10.1155/2018/5060857*

Courtene-Jones, W., Quinn, B., Ewins, C., Gary, S. F. and Narayanaswamy, B. E. (2019). Consistent microplastic ingestion by deep-sea invertebrates over the last four decades (1976–2015), a study from the north east atlantic, Environmental Pollution **244**: 503–512.
URL: *https://www.sciencedirect.com/science/article/pii/S0269749118330483*

Coyle, R., Hardiman, G. and Driscoll, K. O. (2020). Microplastics in the marine environment: A review of their sources, distribution processes, uptake and exchange in ecosystems, Case Studies in Chemical and Environmental Engineering **2**: 100010.
URL: *https://www.sciencedirect.com/science/article/pii/S2666016420300086*

Fabbri, C., Islam, M. J. and Sattar, J. (2018). Enhancing underwater imagery using generative adversarial networks, 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7159–7165.

Fulton, M., Hong, J., Islam, M. J. and Sattar, J. (2019). Robotic detection of marine litter using deep visual detection models, 2019 International Conference on Robotics and Automation (ICRA), pp. 5752–5758.

Funch, O. I., Marhaug, R., Kohtala, S. and Steinert, M. (2021). Detecting glass and metal in consumer trash bags during waste collection using convolutional neural networks, Waste Management **119**: 30–38.
URL: *https://www.sciencedirect.com/science/article/pii/S0956053X20305432*

Gago, J., Carretero, O., Filgueiras, A. and Viñas, L. (2018). Synthetic microfibers in the marine environment: A review on their occurrence in seawater and sediments, Marine Pollution Bulletin **127**: 365–376.
URL: *https://www.sciencedirect.com/science/article/pii/S0025326X17310275*

Kikaki, K., Kakogeorgiou, I., Hoteit, I. and Karantzalos, K. (2024). Detecting marine pollutants and sea surface features with deep learning in sentinel-2 imagery, ISPRS Journal of Photogrammetry and Remote Sensing **210**: 39–54.
URL: *https://www.sciencedirect.com/science/article/pii/S0924271624000625*

Liu, S., Peng, Y. and Liu, L. (2021). A novel ship detection method in remote sensing images via effective and efficient pp-yolo, 2021 IEEE International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), pp. 234–239.

Moniruzzaman, M., Islam, S., Bennamoun, M. and Lavery, P. (2017). Deep learning on underwater marine object detection: A survey, pp. 150–160.

Priyadharsini, R. and Sharmila, T. S. (2019). Object detection in underwater acoustic images using edge based segmentation method, Procedia Computer Science **165**: 759–765. 2nd International Conference on Recent Trends in Advanced Computing ICRTAC -DISRUP - TIV INNOVATION , 2019 November 11-12, 2019.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050920300235*

R, S. and M, M. (2023). Comparing yolov3, yolov5 yolov7 architectures for underwater marine creatures detection, 2023 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 25–30.

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

Sharma, N., Baral, S., Paing, M. and Chawuthai, R. (2023). Parking time violation tracking using yolov8 and tracking algorithms, Sensors **23**: 5843.

Shi, C., Tan, C., Wang, T. and Wang, L. (2021). A waste classification method based on a multilayer hybrid convolution neural network, Applied Sciences **11**(18).
URL: *https://www.mdpi.com/2076-3417/11/18/8572*

Sánchez-Ferrer, A., Valero-Mas, J. J., Gallego, A. J. and Calvo-Zaragoza, J. (2023). An experimental study on marine debris location and recognition using object detection, Pattern Recognition Letters **168**: 154–161.
URL: *https://www.sciencedirect.com/science/article/pii/S0167865522003889*

Valente, T., Scacco, U. and Matiddi, M. (2020). Macro-litter ingestion in deep-water habitats: is an underestimation occurring?, Environmental Research **186**: 109556.
URL: *https://www.sciencedirect.com/science/article/pii/S0013935120304497*

Xia, H. and Tan, L. (2023). Research on underwater object detection algorithm based on improved yolov5, 2023 2nd International Conference on Sensing, Measurement, Communication and Internet of Things Technologies (SMC-IoT), pp. 7–10.

Xu, R., Lin, H., Lu, K., Cao, L. and Liu, Y. (2021). A forest fire detection system based on ensemble learning, Forests **12**: 217.

Đuraš et al. (2024). Seaclear marine debris detection segmentation dataset.
URL: *https://doi.org/10.4121/4f1dff25-e157-4399-a5d4-478055461689.v1*