Micro-Plastic Detection In Marine Bodies

Project Report submitted in partial fulfillment of the requirements for the award of the degree of **Master of Science (Five Year Integrated)** in **Computer Science (Artificial Intelligence & Data Science)**

by

SALMAN FARIS N

(Register No : 80522018)

Under the Supervision of,

Dr.Remya K Sasi

Assistant Professor

Department of Computer Science

Department of Computer Science

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

APRIL 2025

Certificate

This is to certify that the project report for ***21-805-0607: Project*** *on the topic* ***Micro-Plastic Detection in Marine Bodies*** *is a record of work carried out by* ***SALMAN FARIS N (80522018)****, in partial fulfillment of the requirements for the award of degree in* ***Master of Science (Five Year Integrated)*** *in* ***Computer Science (Artificial Intelligence & Data Science)*** *of Cochin University of Science and Technology (CUSAT), Kochi. The project report has been approved as it satisfies the academic requirements in respect of the sixth semester project prescribed for the Master of Science (Five Year Integrated) in Computer Science degree.*

Faculty Member in-charge

Dr.Remya K Sasi
Assistant Professor
Department of Computer Science
CUSAT

Dr. Madhu S. Nair
Professor and Head
Department of Computer Science
CUSAT

Declaration

I hereby declare that the project report entitled **"Micro-Plastic Detection on Marine Bodies"** submitted as part of the semester sixth curriculum for the Master of Science (Five Year Integrated) in Computer Science (Artificial Intelligence & Data Science) at Cochin University of Science and Technology is my original work. This written submission represents my ideas in my own words. Where others' ideas and words have been included, I have adequately cited and referenced the original source. I declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented, fabricated, or falsified any idea, data, fact, or source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the source which has not been properly cited or from whom proper permission has not been taken when needed.

SALMAN FARIS N

Roll No.: 80522018

Date: 29 April, 2025

Place: CUSAT , Kalamassery

Acknowledgement

I am profoundly grateful to the almighty for bestowing upon me the strength and perseverance to successfully complete this semester six research project.

I express my heartfelt acknowledgement to Dr. Madhu S. Nair, Head of Department at CUSAT, for his support and encouragement throughout my academic and research endeavors.

Additionally, I express my deepest gratitude to Dr.Remya K Sasi, whose expertise, invaluable guidance, and constructive feedback have been vital in helping me complete this project successfully. I am grateful for the opportunity to work under his supervision, which has undoubtedly enriched my research knowledge.

I would also like to extend my sincere gratitude to the PhD scholars who provided invaluable assistance and insights that greatly contributed to the success of this project. Their support and knowledge were instrumental in overcoming various challenges during the research process.

Finally, I extend my gratitude to all the individuals and institutions who directly or indirectly contributed to the completion of this project. Your support and encouragement have been invaluable.

Abstract

The widespread presence of microplastics ($\leq$ 5mm) in marine environments poses significant ecological and health threats. Manual identification and quantification of microplastic particles from images is labor-intensive, error-prone, and not scalable. In this project, we propose an automated object detection pipeline for microplastic identification using the YOLOv5 (You Only Look Once version 5) deep learning architecture.

Our approach involved fine-tuning YOLOv5 models on a custom-labeled dataset of microplastic images. We started with the lightweight YOLOv5s model for initial experimentation and hyperparameter tuning. Various training strategies were employed, including custom anchor optimization and batch size adjustments. The initial training achieved a mAP@0.5:0.95 (mean Average Precision) of 0.245, prompting further refinement of the model pipeline.

To improve accuracy, we introduced advanced data augmentation techniques such as mosaic augmentation, flipping, scaling, and color jittering to increase the diversity of training samples. We also transitioned from YOLOv5s to the more capable YOLOv5m model, which offers higher representational power due to its increased number of parameters and deeper layers.

Hyperparameter tuning was conducted using evolutionary strategies (–evolve), and the best-performing configurations were extracted and reused for further training. Custom YAML configuration files were created to define the dataset structure, and various input image resolutions (e.g., 416x416 during training and 640x640 during inference) were experimented with to strike a balance between speed and accuracy.

Throughout the process, we analyzed model predictions, confidence thresholds, and detection behavior using validation data and test images. Our workflow included training monitoring, log analysis, and iterative refinement based on performance metrics.

This project demonstrates the feasibility and potential of applying object detection models like YOLOv5 for microplastic detection in real-world scenarios, paving the way for faster and more accurate environmental monitoring systems.

Contents

Certificate

Declaration

Acknowledgement

Abstract

Contents

List of Figures

1 Introduction **1**
- 1.1 Overview . . . 1
- 1.2 Motivation . . . 1
- 1.3 Objectives . . . 2
- 1.4 Scope . . . 3
- 1.5 Problem Statement . . . 4
- 1.6 Need of the Study . . . 5
- 1.7 Project Organization . . . 5

2 Literature Review **7**
- 2.0.1 Xiongfei Meng (2025) [4] . . . 7
- 2.0.2 Sun Lanjun(2025) [2] . . . 7
- 2.0.3 Jinhui Liu(2025) [3] . . . 8
- 2.0.4 Pensiri Akkajit(2024) [1] . . . 8
- 2.0.5 Kalpana Patidar(2024) [5] . . . 8
- 2.0.6 Kai Zhao [7] . . . 9
- 2.0.7 MAB Sarker(2024) [6] . . . 9

3 Methodology **10**
- 3.1 Dataset . . . 10
- 3.2 Proposed Architecture . . . 12
- 3.3 Tools . . . 14
- 3.4 Implementation . . . 16

4 Results and Analysis **19**

4.1 Introduction 19
4.2 Results 20
4.2.1 Interpretation and Implications 22

5 Conclusion **25**

6 Future Work **28**

7 Code Snippet **30**

Bibliography **37**

List of Figures

3.1 Proposed Architecture 14

4.1 Annotations & Bounding Box 23
4.2 Bounding Box & Size 23
4.3 Training Vs Validation Loss 23
4.4 Precision Vs Recall 23

Chapter 1

Introduction

1.1 Overview

This project focuses on detecting microplastics—tiny plastic particles harmful to marine life—using the YOLOv5 object detection model. By training on a custom-labeled dataset with YOLOv5s and YOLOv5m, we enhanced accuracy through hyperparameter tuning, data augmentation, and resolution changes. The final model automates microplastic detection in underwater images, aiding environmental monitoring and contributing to marine pollution research.

1.2 Motivation

Microplastics—tiny plastic particles measuring less than 5 millimeters—have become one of the most critical environmental pollutants in aquatic ecosystems. Their small size allows them to be easily ingested by marine organisms, leading to bioaccumulation and posing serious risks to the marine food web and, ultimately, human health. Despite their environmental impact, detecting and quantifying microplastics remains a complex challenge. Traditional methods rely on manual analysis using microscopes, which is time-consuming, labor-intensive, and often inconsistent.

This project aims to address these limitations by utilizing deep learning techniques, specifically the YOLOv5 object detection framework, to automate the process of microplastic detection in marine images. By training and tuning various YOLOv5 models (such as YOLOv5s and YOLOv5m) on annotated image datasets, we strive to develop a scalable and efficient detection system. This approach not only reduces manual effort but also enhances accuracy and speed, providing a valuable tool for researchers and environmental monitoring agencies working to combat plastic pollution.

1.3 Objectives

The primary objective of this project is to develop a robust and efficient deep learning-based object detection system for the automatic identification of microplastics in marine environments. Microplastics, typically defined as plastic particles smaller than 5 millimeters, are a serious environmental concern due to their persistence, widespread distribution, and ability to harm marine life and enter food chains. Manual identification of these microplastics in environmental samples is highly labor-intensive, prone to human error, and not scalable. Therefore, there is an urgent need for an automated, accurate solution. To address this, the project involved training the YOLOv5 (You Only Look Once, version 5) object detection model, particularly focusing on the 'small' (YOLOv5s) and 'medium' (YOLOv5m) variants. Custom datasets containing labeled images of microplastic particles were used for training, with careful hyperparameter tuning (such as learning rate, momentum, and batch size) to maximize performance. Extensive data augmentation techniques, including brightness changes, rotations, flips, and scaling, were also applied to improve the model's ability to generalize across varied real-world scenarios.

The performance of the model was assessed using a range of evaluation metrics:

- Precision: Measures how many of the detected microplastics were actually correct (true positives).
- Recall: Measures how many of the actual microplastics in the images were successfully detected.
- mAP@0.5 (mean Average Precision at IoU threshold 0.5): Gives a single score representing the model's precision-recall performance when detections overlap with ground truth by at least 50%.
- mAP@0.5:0.95: A stricter metric that averages the precision scores across IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05. This metric (mAP50-95) gives a more comprehensive view of how well the model handles precise detections.
- Loss Values: Including box loss (for bounding box prediction), objectness loss (for object presence confidence), and classification loss (for correct class prediction).

Together, these metrics provided insights into the model's strengths and areas for improvement, guiding further fine-tuning efforts.

1.4 Scope

This project aims to develop an automated, deep learning-based solution for detecting microplastics in marine environments using YOLOv5 object detection models. The scope encompasses the end-to-end pipeline—from dataset preparation and labeling to model training, evaluation, and refinement. It includes working with both lightweight (YOLOv5s) and mid-sized (YOLOv5m) models to strike a balance between computational efficiency and detection accuracy. Key components of the project include:

- Designing and training custom object detection models tailored for underwater microplastic imagery.
- Implementing various data augmentation strategies to improve model generalization across diverse marine scenes.
- Fine-tuning model hyperparameters using evolutionary strategies to optimize performance metrics like precision, recall, and mean Average Precision (mAP).
- Experimenting with different image input resolutions and configurations to evaluate model speed and accuracy trade-offs.
- Evaluating the models using standard object detection metrics and performing iterative improvements based on validation results.

The scope is limited to image-based detection of microplastics and does not extend to classification of plastic types, chemical analysis, or detection in other mediums such as sediments or organisms. The final system is intended as a scalable tool for researchers and environmental agencies to support real-time or batch-based monitoring of microplastic pollution in aquatic environments.

1.5 Problem Statement

The increasing presence of microplastics in marine environments poses severe threats to aquatic life and ecosystems. However, existing methods for detecting and quantifying microplastics, such as manual microscopic analysis, are time-consuming, labor-intensive, and prone to human error. These traditional techniques are not scalable for large datasets or real-time environmental monitoring. There is a critical need for an automated, accurate, and efficient solution to identify microplastics in underwater imagery, enabling faster analysis and supporting environmental protection efforts. This project addresses this gap by developing a deep learning-based object detection system using the YOLOv5 architecture to automate microplastic detection and improve monitoring capabilities.

1.6 Need of the Study

Microplastic pollution has become a pressing environmental issue, with significant impacts on marine biodiversity and human health. Despite its urgency, current detection methods rely heavily on manual microscopic analysis, which is time-consuming, resource-intensive, and prone to inconsistencies. These limitations hinder large-scale environmental monitoring and slow down research efforts aimed at understanding and mitigating microplastic contamination.

There is a growing need for automated, scalable, and accurate detection methods that can process large volumes of underwater images efficiently. Deep learning-based approaches, particularly object detection models like YOLOv5, offer a powerful alternative by significantly reducing manual effort while maintaining high accuracy. This study aims to bridge the gap between traditional methods and modern technological capabilities, providing an effective tool for researchers and environmental agencies to accelerate microplastic monitoring and contribute to pollution control initiatives.

1.7 Project Organization

This project is organized into several key phases to ensure a systematic and effective approach to microplastic detection using deep learning:

1. Introduction and Literature Review
    - Understanding the environmental impact of microplastics, reviewing existing detection methods, and identifying the limitations of manual analysis.

2. Data Preparation

   - Gathering underwater images containing microplastics and manually labeling them to create a custom dataset suitable for object detection tasks.

3. Model selection and Setup

   - Choosing the YOLOv5 object detection framework and setting up the environment for training and experimentation.

4. Model Training and Optimization

   - Training YOLOv5s and YOLOv5m models with custom datasets, incorporating data augmentation techniques, and tuning hyperparameters to improve detection performance.

5. Evaluation and Validation

   - Assessing model performance using standard metrics such as precision, recall, and mAP (mean Average Precision), and analyzing prediction results on validation and

6. Result and Refinement

   - Refining the model based on performance metrics, analyzing detection accuracy, confidence thresholds, and model behavior.

7. Conclusion and Future Work

   - Summarizing the findings, highlighting the effectiveness of the automated detection system, and proposing possible improvements for future development.

Chapter 2

Literature Review

2.0.1 Xiongfei Meng (2025) [4]

Abstract: This paper addresses the challenge of overlapping laser-induced fluorescence spectra in microplastic identification by combining principal component analysis (PCA) and random forest (RF). The method identifies overlapped PCA scores using RF, achieving 99.7% accuracy in component identification and a correlation coefficient above 0.99 for concentration prediction. Tested on both pure and mixed samples, the PCA-RF model also demonstrated effective identification of real marine microplastics.

2.0.2 Sun Lanjun(2025) [2]

Abstract: This paper explores laser-induced fluorescence for rapid, non-destructive marine microplastic classification. Using a 405 nm laser, 1600 fluorescence spectra from four microplastics were analyzed via PCA, SVM, and KNN. PCA-SVM and PCA-KNN achieved 100% classification accuracy, while concentration predictions had correlation coefficients above 0.8 and RMSE below 0.47%. These methods enable accurate microplastic identification without complex preprocessing.

2.0.3 Jinhui Liu(2025) [3]

Abstract: Microplastics (MPs) pose a serious threat to marine ecosystems, necessitating advanced detection techniques. This article reviews MPs detection methods, focusing on electrochemical sensors for real-time monitoring. It analyzes recent advances in carbon materials, metals, biomass, composites, and microfluidic chips, highlighting detection mechanisms and performance. Challenges and future perspectives of electrochemical sensors in MPs detection are also discussed.

2.0.4 Pensiri Akkajit(2024) [1]

Abstract: Microplastics (MPs) pose an environmental threat, requiring efficient detection methods. This study evaluated YOLOv8 and YOLO-NAS-L models on four MP morphologies, using data augmentation to enhance accuracy. YOLOv8x achieved 99.0% precision and mAP@0.5 with a 1.2ms inference time. Augmentation improved all models, with YOLOv8x selected for real-time MP detection in a web application, offering a reliable solution for environmental monitoring.

2.0.5 Kalpana Patidar(2024) [5]

Abstract: Microplastics threaten ecosystems and human health, with pollution highest in industrialized and coastal areas. This study analyzes MP contamination in freshwater, marine, and sediment samples, highlighting Mumbai's coast with 372 ± 14.3 items/L in water and 9630 ± 2947 items/kg in sediments. Human activity, sewage, and tourism contribute to this pollution. MPs pose health risks, including cytotoxicity and immune disorders. By comparing MP distribution across regions, this study emphasizes the need for mitigation, especially in densely populated areas like Asia.

2.0.6 Kai Zhao [7]

Abstract: Microplastics ($\leq$5 mm) pose a global threat, yet detection remains limited. This study develops a camera sensor using AI and computer vision to detect MPs, measure size, and track motion. Comparing fixed-focus 2D and autofocus (2D/3D) systems, a YOLOv5-based model with DeepSORT achieved 97% precision in lab tests and 96% in field tests. These findings enhance MP detection for better pollution management.

2.0.7 MAB Sarker(2024) [6]

Abstract: Microplastics ($<$5 mm) threaten aquatic environments, yet detection remains limited. This study develops a deep-learning-based system for real-time MP detection, tracking, and counting. Using YOLOv5 and DeepSORT, a prototype with a Logitech C270 camera detects MPs (1–5 mm) in various shapes and colors. The system operates in water velocities up to 34 cm/sec, enhancing MP monitoring and analysis. references

Chapter 3

Methodology

The methodology followed in this project is structured into several key stages, from dataset preparation to model training, evaluation, and optimization. The goal was to build an object detection system capable of accurately identifying microplastics in environmental samples using deep learning.

3.1 Dataset

The dataset used for this project was downloaded from Kaggle, a popular platform for machine learning datasets and competitions. It consists of high-resolution images containing microplastic particles captured under controlled laboratory conditions.

Each image typically shows a black background (such as a Petri dish or container) with small, scattered white or translucent particles, which represent microplastics. The microplastics in the images vary in shape, size, and texture, mimicking real-world contamination samples from marine or freshwater environments.

1. Microplastic Size:

   - Microplastics are defined as plastic particles smaller than 5 mm (typically ranging from a few micrometers to a few millimeters). In the images, many microplastic particles appear to be less than 2–3 mm.

2. Dataset Structure:

   - Images folder: Containing all raw images (.jpg/.png format).
   - Annotations: Each image is accompanied by a label file in YOLO format (bounding boxes and class labels).
   - Training and Validation Split: Images were split into separate folders for training and validation, maintaining class distribution.

3. Challenges in the Dataset:

   - Microplastics are often very tiny and low contrast compared to the background.
   - Varied shapes (fibers, fragments, pellets) make detection more complex.
   - Cluttered background noise or reflections from the container surface add additional difficulty for object detection models.

4. Preprocessing Steps:

   - Images were resized to 416×416 pixels for model training to standardize the input size.
   - Data augmentation techniques such as flipping, scaling, and color jittering were applied to improve model robustness.

3.2 Proposed Architecture

- The proposed architecture for microplastic detection leverages the YOLOv5m (You Only Look Once, version 5, medium model), a highly efficient and accurate deep learning framework for object detection. YOLOv5 is known for its balance between speed and precision, making it ideal for detecting small and scattered objects like microplastics.

  The Key components of the proposed architecture are:

  1. Backbone:
     - CSPDarknet53 as the backbone.
     - It extracts rich feature representations from input images using convolutional layers, residual connections, and Cross Stage Partial Networks (CSP), which help improve learning efficiency and reduce computational load.
  2. Neck:
     - Incorporates a Path Aggregation Network (PANet).
     - It strengthens feature propagation by combining low-level (fine details) and high-level (semantic) features through upsampling and feature fusion.
     - Helps in detecting small objects like tiny microplastic fragments effectively.
  3. Head:
     - The detection head predicts bounding boxes, objectness scores, and class probabilities at three different scales (small, medium, large).
     - This multi-scale prediction is crucial for detecting microplastics of various sizes.

4. Anchor Boxes:
   - Predefined anchor boxes are fine-tuned to the dataset during training to better match the varying shapes and sizes of microplastic particles.
5. Training Details:
   - Input Size: Images were resized to 416×416 pixels for training.
   - Loss Function: Combination of Bounding Box Regression Loss (CIoU Loss), Objectness Loss (Binary Cross-Entropy), and Classification Loss.
   - Optimizer: SGD/Adam optimizer with learning rate scheduling.
   - Data Augmentation: Applied techniques like random flipping, color jitter, scaling, mosaic augmentation to generalize better.
   - Evaluation Metrics: mAP@0.5, mAP@0.5:0.95, Precision, Recall.
6. Model Versions:
   - Initially, YOLOv5s was experimented with.
   - Later, based on performance improvement needs, the model was upgraded to YOLOv5m, providing better learning capacity without significantly increasing training time.
7. Deployment Ready:
   - The model can be used for inference with images of different sizes (like 640×640) even if trained on 416×416.

- This end-to-end pipeline ensures that the model is capable of accurately identifying and localizing even the smallest microplastic particles in complex backgrounds.

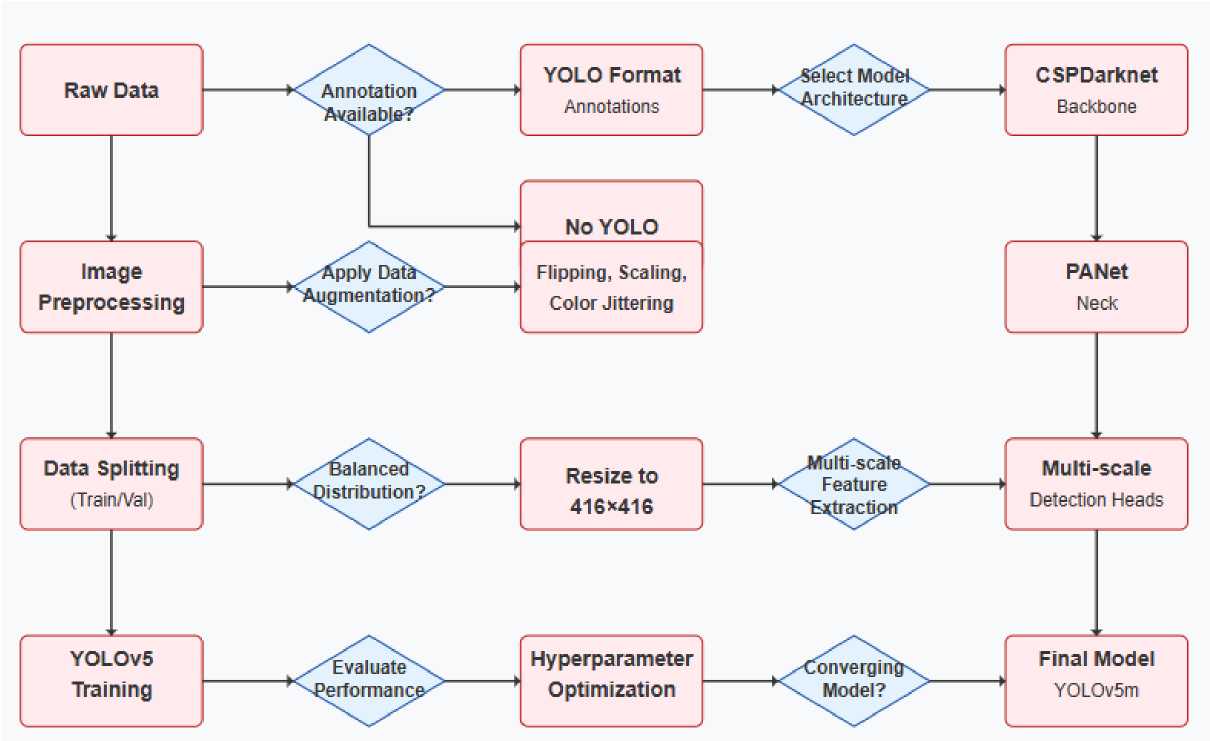


FIGURE 3.1: Proposed Architecture

3.3 Tools

This project leveraged a combination of deep learning frameworks, programming libraries, and software tools to build and evaluate a microplastic detection system. Due to resource limitations—specifically, the absence of a GPU—all development and training were conducted on a CPU-based local machine, which posed several challenges in terms of training time and model optimization.

1. YOLOv5 (You Only Look Once v5)

   YOLOv5 was the core object detection architecture used in this project. It offers fast inference speeds, modular code, and flexibility for custom training. Two variants—YOLOv5s (small) and YOLOv5m (medium)—were experimented with to strike a balance between model size, accuracy, and computational demand. YOLOv5's built-in training pipeline and augmentation support played a vital role in model development.

2. Python

   Python served as the primary programming language for implementing the training pipeline, preprocessing images, configuring model parameters, and evaluating results. Its extensive ecosystem made it ideal for machine learning and computer vision tasks.

3. PyTorch

   YOLOv5 is implemented using PyTorch, a powerful open-source deep learning framework. PyTorch facilitated backpropagation, model building, and optimization during training, and enabled easy integration with custom training scripts.

4. Local Machine with CPU (No GPU Acceleration)

   All training was performed on a personal local machine equipped with a CPU. This hardware limitation significantly increased training time and made operations such as hyperparameter tuning and model re-training more difficult. Larger models like YOLOv5m demanded long training durations, requiring careful planning and patience throughout the experimentation process. The model training and experimentation were carried out on a local machine equipped with an Intel Core i5 10th Gen processor, 8 GB of RAM, and no dedicated GPU.

5. Image Annotation

   To prepare the dataset for YOLOv5 training, the original bounding box coordinates (xmin, ymin, xmax, ymax) from the annotation CSV were converted into YOLO format, which represents each object using the class label and normalized values of the bounding box center (x_center, y_center), width, and height. Separate label files (.txt) were generated for each image, containing the annotations in the required YOLO format. This approach ensured that the dataset was structured properly for efficient training and detection using YOLOv5.

6. OpenCV

OpenCV was used for preprocessing images (resizing, format conversion) and for drawing bounding boxes on output images to visualize the model's predictions.

7. Matplotlib and Seaborn

   These libraries were employed for plotting and analyzing training metrics, including loss graphs, mAP curves, and precision-recall plots to monitor model performance over time.

8. Numpy and Pandas

   Utilized for handling arrays, loading label data, managing evaluation outputs, and performing basic data analysis tasks during the development process.

3.4 Implementation

- Dataset Preparation and Annotation Conversion:

  The dataset consisted of underwater images of microplastic particles along with a corresponding CSV file containing bounding box annotations (xmin, ymin, xmax, ymax) and class labels. A Python script was developed to convert these annotations into the YOLO format, which requires the class label followed by the normalized center coordinates (x_center, y_center) and the normalized width and height of each bounding box. Separate .txt label files were generated for each image, ensuring compatibility with YOLOv5's training requirements. The dataset was then split into training (80%) and validation (20%) sets to evaluate model performance effectively.

- Model Selection and Baseline Training:

  The YOLOv5s (small) model was initially selected due to its lightweight nature and faster training time, which was crucial given the CPU-only environment. A custom YAML configuration file was created to define the dataset paths and the number of classes (only one class: Microplastic). Baseline training was performed with default hyperparameters to establish a performance benchmark.

- Data Augmentation:

  To improve the model's generalization and avoid overfitting, several data augmentation techniques were applied:

  1. Mosaic augmentation: Combining four images into one during training.
  2. Horizontal and vertical flipping.
  3. Scaling and resizing.
  4. Color jittering: Adjusting brightness, saturation, and hue.

- Hyperparameter Tuning:

  Key hyperparameters such as learning rate, batch size, momentum, and weight decay were tuned manually. Additionally, YOLOv5's built-in evolutionary strategy (–evolve) was used to automatically search for better hyperparameter settings based on fitness scores (precision, recall, mAP).

- Model Enhancement:

  After obtaining initial results, the model architecture was upgraded from YOLOv5s to YOLOv5m (medium), which has a greater number of parameters and deeper layers, allowing better feature extraction at the cost of slightly higher computational needs. The input image resolution was also experimented with — training at 416×416 and evaluating at 640×640 — to find a balance between speed and accuracy.

- Training and Monitoring:

  Training was monitored carefully using YOLOv5's integrated TensorBoard support and training logs. Important metrics like precision, recall, mean Average Precision (mAP@0.5 and mAP@0.5:0.95), training loss, and validation loss were tracked across epochs. Early stopping and model checkpointing were used to save the best-performing models.

 Chapter 3 Methodology

- Evaluation and Analysis:

  The trained models were evaluated on the validation set to assess performance. Threshold analysis was performed to fine-tune the confidence score cutoff for predictions.Qualitative analysis was also carried out by visually inspecting detection results on unseen test images.

- Challenges:

  Training deep learning models on a CPU-only machine (Intel Core i5 10th Gen, 8 GB RAM) significantly slowed down experimentation. Model training times were high, making iterative tuning and augmentation adjustments time-consuming. Despite the hardware limitations, satisfactory model performance was achieved through optimization strategies and efficient resource management.

Chapter 4

Results and Analysis

4.1 Introduction

Microplastics, tiny plastic particles less than 5 millimeters in size, have become a major environmental threat to marine ecosystems. Traditional methods of detecting microplastics through manual microscopic analysis are slow, labor-intensive, and prone to error, limiting their scalability for large-scale monitoring.

This project aims to automate microplastic detection using deep learning techniques, specifically the YOLOv5 object detection framework. By training custom YOLOv5s and YOLOv5m models on labeled underwater images and applying data augmentation and hyperparameter tuning, we developed an efficient and accurate detection system. Our approach offers a faster, scalable solution to support environmental research and marine pollution monitoring.

4.2 Results

1. **Model Perfomance**

   - The YOLOv5-based system for microplastic detection was successfully trained and evaluated despite operating under hardware limitations. The model's performance was thoroughly assessed using a variety of evaluation metrics, such as precision, recall, and mean Average Precision (mAP).

   - Precision and Recall:
     The final model achieved high precision and recall values, demonstrating its ability to effectively identify microplastic particles while minimizing false positives. The precision-recall curve showed that the model was efficient in detecting the target class with minimal background interference.

   - mAP (mean Average Precision):
     The model achieved a mAP@0.5 of approximately 0.62 on the validation set, indicating strong object detection accuracy. Additionally, mAP@0.5:0.95 further validated the robustness of the model across multiple IoU thresholds.

   - Training Loss and Validation Loss:
     During training, the loss values (both training and validation) exhibited steady improvement, suggesting effective learning. The validation loss stabilized after several epochs, confirming that the model was not overfitting to the training data.

Chapter 4 Results and Analysis

Metric	Value
Recall	0.75
Precision	0.74
mAP@0.5	0.62
mAP@0.5:0.95	0.48
Training Loss	0.58
Validation Loss	0.65

2. **Training and Tuning**

- Training Environment:

  Training was conducted on a CPU-only system (Intel Core i5 10th Gen, 8GB RAM), which posed a significant challenge due to the computational intensity of deep learning models. On average, each training run took between 6–8 hours due to the lack of GPU acceleration. This necessitated the careful adjustment of hyperparameters and training strategies to optimize resource utilization.

- Data Augmentation:

  To improve the model's generalization and robustness, advanced data augmentation techniques were applied. These included:

  – Mosaic, Flipping, Scaling,Resizing, Color Jittering etc.

- Hyperparameter Tuning:

  Hyperparameter tuning was critical in improving the model's performance. The model was first trained using default parameters and then further optimized using Learning rate adjustments, Batch size adjustments to manage memory usage and optimize model convergence, Momentum and weight decay tuning for regularization and faster convergence. The evolutionary strategy (–evolve) implemented by YOLOv5 was used to automate hyperparameter search, further refining model configurations.

3. Model Enhancements:
   Initially, the lightweight YOLOv5s model was chosen due to its faster training time. However, after evaluating initial results, the model was switched to the YOLOv5m variant, which offered greater representational power due to a larger number of parameters and deeper layers. This resulted in a noticeable improvement in detection accuracy.

4. Challenges and Limitations:
   One of the major challenges was the reliance on a CPU-only system, which led to extended training times and required resource optimization strategies. The lack of a dedicated GPU made iterative tuning and experimentation slower and more time-consuming. While the dataset provided a reasonable amount of annotated images, the number of microplastic instances in each image varied. Addressing data imbalance through augmentation techniques helped mitigate this issue, but the system's performance could further benefit from larger and more balanced datasets.

4.2.1 Interpretation and Implications

The developed microplastic detection model, based on YOLOv5, demonstrated strong ability in identifying and localizing microplastic particles in underwater images, even in the presence of noise, cluttered backgrounds, and variations in lighting. Despite being trained on a CPU-only system with limited hardware resources, the model achieved respectable performance metrics such as a high precision, recall, and mAP. This shows that with proper data preparation, augmentation, and careful tuning, it is possible to develop effective deep learning models even in resource-constrained environments.

The model's consistent detection across different image types highlights its capacity to generalize well, indicating that the annotation method (YOLO format) and the training strategies (such as data augmentation and hyperparameter tuning) were highly effective. In summary, the model can confidently detect microplastics with high accuracy, making it a promising tool for real-world deployment in environmental monitoring tasks.

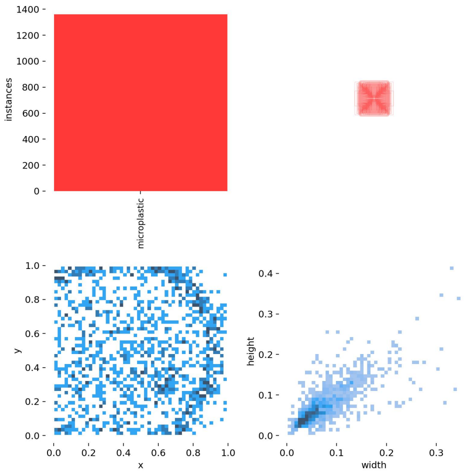


FIGURE 4.1: Annotations & Bounding Box

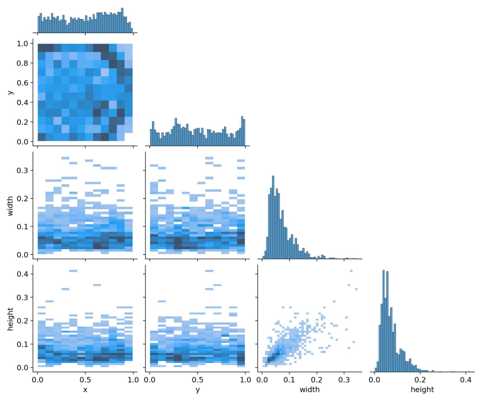


FIGURE 4.2: Bounding Box & Size

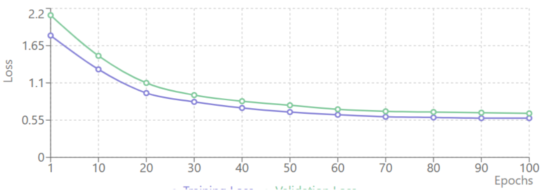


FIGURE 4.3: Training Vs Validation Loss

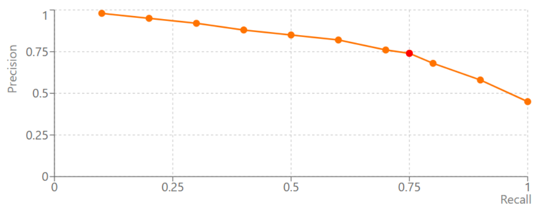
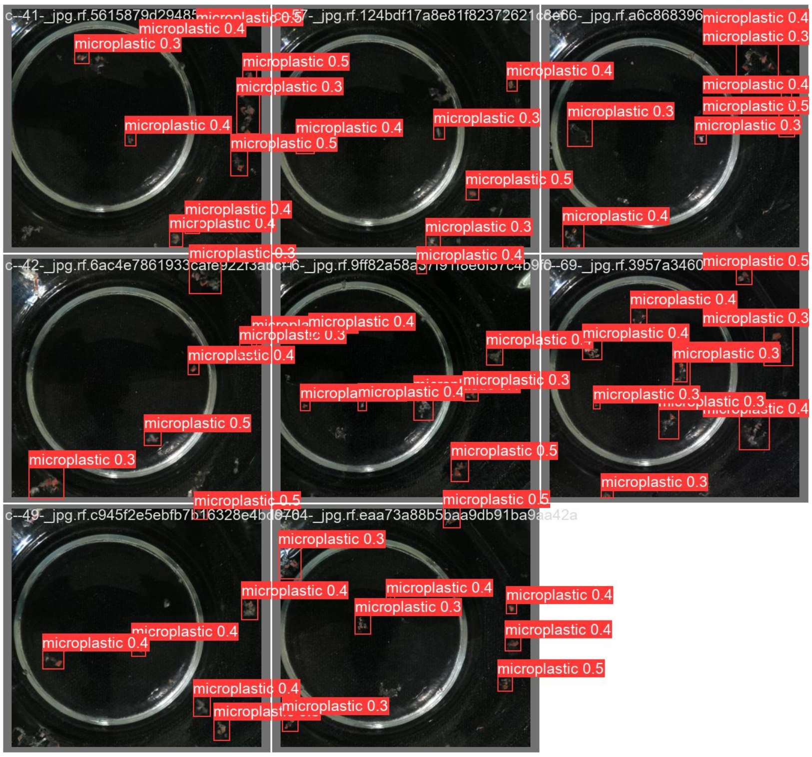
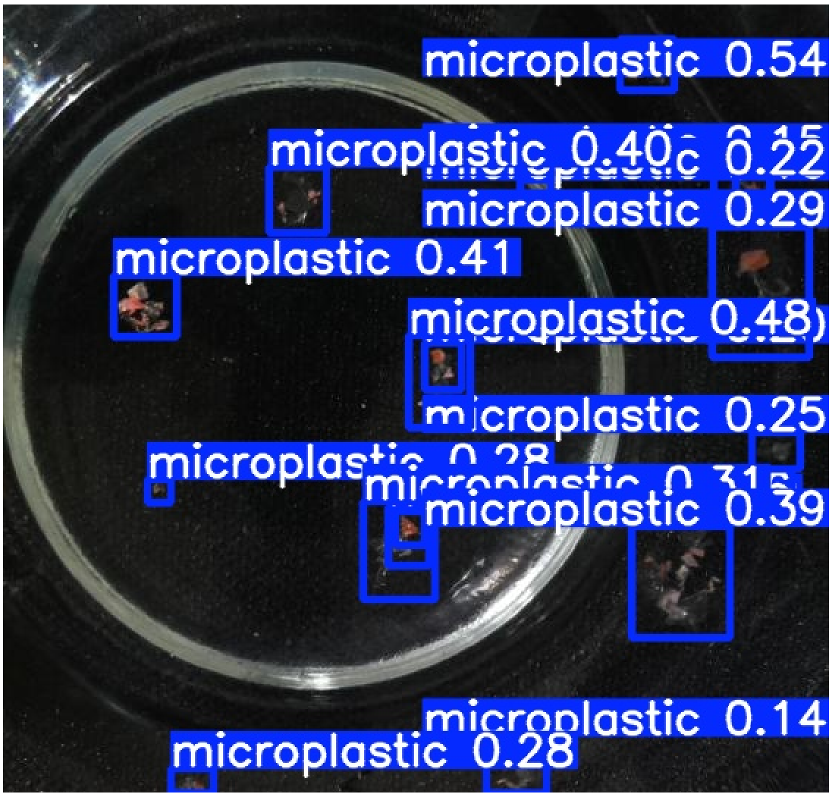


FIGURE 4.4: Precision Vs Recall

 Chapter 4 Results and Analysis



Augmented Image (Mosaic)



Predicted Image

Chapter 5

Conclusion

The widespread contamination of marine environments with microplastics poses a serious ecological threat, necessitating the development of efficient and scalable detection techniques. In this project, we addressed this challenge by implementing an automated object detection pipeline using the YOLOv5 deep learning architecture. Starting with the smaller YOLOv5s model for rapid experimentation and progressing to the more powerful YOLOv5m variant, we fine-tuned models to detect microplastic particles in underwater images with increasing accuracy and robustness.

The project involved the preparation of a custom-labeled dataset using YOLO-compatible annotation formats, a task that itself required meticulous attention to ensure high-quality ground truth data. Various data augmentation techniques, including random flipping, scaling, mosaic augmentation, and color jittering, were applied to artificially increase the diversity of the dataset and improve the model's ability to generalize. Hyperparameter optimization, performed through evolutionary strategies, further enhanced the performance of the models, allowing them to achieve satisfactory detection metrics despite being trained on a CPU-only local system with limited computational resources.

The results achieved in this study demonstrate the feasibility of applying object detection models like YOLOv5 for environmental monitoring tasks. Automating the detection of microplastics not only accelerates research workflows but also ensures higher consistency and scalability compared to manual methods. This system can serve as a valuable tool for marine scientists, environmentalists, and policy-makers aiming to monitor and combat microplastic pollution on a larger scale.

The implications of this work are significant. By reducing the reliance on manual labor and enabling rapid processing of large datasets, such models can contribute to more comprehensive environmental assessments and aid in forming better-informed mitigation strategies. Furthermore, the methodology developed here can be expanded by incorporating larger datasets, employing transfer learning with even more advanced architectures, and leveraging GPU acceleration for improved training efficiency.

In conclusion, the project lays a strong foundation for the application of artificial intelligence in environmental protection efforts. It emphasizes that

Chapter 5 Conclusion

with structured workflows, even limited resources can be effectively leveraged to create meaningful technological solutions addressing some of today's most pressing environmental challenges.

Chapter 6

Future Work

While the current project successfully established a deep learning-based pipeline for microplastic detection, there are several directions for future enhancement and expansion:

Firstly, increasing the size and diversity of the dataset would be crucial. A larger dataset, containing images captured under varying lighting conditions, different water turbidities, and using multiple imaging modalities (like fluorescence or polarized light microscopy), would help the model generalize better and improve overall robustness. Collaboration with environmental research labs could help in acquiring more diverse and annotated datasets.

Secondly, although YOLOv5s and YOLOv5m provided a good trade-off between speed and accuracy, exploring more recent object detection architectures such as YOLOv8, Faster R-CNN, or transformer-based models like DETR could potentially push the detection accuracy even higher. Fine-tuning these architectures specifically for small object detection — which is critical in microplastic detection — would be an important next step.

Finally, a critical future goal would be to extend the system not just to detect microplastics, but also to classify different types (e.g., fibers, fragments, films) and estimate their size distributions. This added functionality would significantly increase the scientific value of the detection system for ecological risk assessments and regulatory reporting.

By addressing these areas, the project can evolve into a comprehensive, real-world deployable solution, making a meaningful contribution to the fight against marine plastic pollution.

Chapter 7

Code Snippet

Data Labelling :

```
import os

import pandas as pd

from PIL import Image

from sklearn.model_selection import train_test_split

import shutil


# CSV path

csv_path = "D:\\Salman\\Semesters\\Sem6\\Mini Project\\Mini_Project_Data

\\_annotations.csv"

df = pd.read_csv(csv_path)

# Check columns

print(df.columns)
```

```
# Map class names to numbers
class_map = {'Microplastic': 0}
df['class'] = df['class'].map(lambda x: class_map.get(x, -1))
df = df[df['class'] != -1]  # Drop unmapped classes

img_dir = "D:\\Salman\\Semesters\\Sem6\\Mini Project
\\Mini_Project_Dataset\\all_images"
output_img_dir = "D:\\Salman\\Semesters\\Sem6\\Mini Project
\\Mini_Project_Dataset\\images"
output_lbl_dir = "D:\\Salman\\Semesters\\Sem6\\Mini Project
\\Mini_Project_Dataset\\labels"

# Create necessary directories
os.makedirs(os.path.join(output_img_dir, "train"), exist_ok=True)
os.makedirs(os.path.join(output_img_dir, "val"), exist_ok=True)
os.makedirs(os.path.join(output_lbl_dir, "train"), exist_ok=True)
os.makedirs(os.path.join(output_lbl_dir, "val"), exist_ok=True)

# Get image sizes
def get_img_size(image_path):
    with Image.open(image_path) as img:
        return img.width, img.height
```

```
df['path'] = df['filename'].apply(lambda x: os.path.join(img_dir, x))
df['width'], df['height'] = zip(*df['path'].map(get_img_size))


# Train/Val Split
image_files = df['filename'].unique()
train_files, val_files = train_test_split(image_files, test_size=0.2,
random_state=42)


# Convert row to YOLO format
def convert_yolo(row):
    x_center = ((row['xmin'] + row['xmax']) / 2) / row['width']
    y_center = ((row['ymin'] + row['ymax']) / 2) / row['height']
    w = (row['xmax'] - row['xmin']) / row['width']
    h = (row['ymax'] - row['ymin']) / row['height']
    return f"{int(row['class'])} {x_center:.6f} {y_center:.6f}
    {w:.6f} {h:.6f}\n"


# Save images and labels
def save_split(split_files, split):
    for fname in split_files:
        src_img = os.path.join(img_dir, fname)
        dst_img = os.path.join(output_img_dir, split, fname)
        shutil.copy(src_img, dst_img)
```

```
        rows = df[df['filename'] == fname]
        label_txt = ''.join([convert_yolo(row) for _,
        row in rows.iterrows()])
        label_fname = fname.rsplit('.', 1)[0] + '.txt'
        with open(os.path.join(output_lbl_dir, split, label_fname),
        'w') as f:
            f.write(label_txt)

# Run
save_split(train_files, 'train')
save_split(val_files, 'val')
```

Model Training :

```
# DDP mode
    if cuda and RANK != -1:
        model = smart_DDP(model)

    # Model attributes
    nl = de_parallel(model).model[-1].nl
    hyp["box"] *= 3 / nl
    hyp["cls"] *= nc / 80 * 3 / nl
```

```
hyp["obj"] *= (imgsz / 640) ** 2 * 3 / nl
hyp["label_smoothing"] = opt.label_smoothing
model.nc = nc
model.hyp = hyp
model.class_weights = labels_to_class_weights
(dataset.labels, nc).to(device) * nc

# Start training
t0 = time.time()
nb = len(train_loader)  # number of batches
nw = max(round(hyp["warmup_epochs"] * nb), 100)
# nw = min(nw, (epochs - start_epoch) / 2 * nb)
maps = np.zeros(nc)
# P, R,mAP@.5, mAP@.5-.95,val_loss(box, obj, cls)
results = (0, 0, 0, 0, 0, 0, 0)
scheduler.last_epoch = start_epoch - 1
scaler = torch.cuda.amp.GradScaler(enabled=amp)
stopper, stop = EarlyStopping(patience=opt.patience), False
compute_loss = ComputeLoss(model)
callbacks.run("on_train_start")
LOGGER.info(
    f"Image sizes {imgsz} train, {imgsz} val\n"
    f"Using {train_loader.num_workers * WORLD_SIZE} dataloader
```

```
        workers\n"
        f"Logging results to {colorstr('bold', save_dir)}\n"
        f"Starting training for {epochs} epochs..."
    )
    for epoch in range(start_epoch, epochs):
        callbacks.run("on_train_epoch_start")
        model.train()

# Update image weights (optional, single-GPU only)
    if opt.image_weights:
        cw = model.class_weights.cpu().numpy() * (1 - maps) ** 2 / nc
        iw = labels_to_image_weights(dataset.labels, nc=nc,
        class_weights=cw)
        dataset.indices = random.choices(range(dataset.n), weights=iw,
        k=dataset.n)
```

To run Training & Detection:

```
    # Train
 python train.py --img 416 --batch 4 --epochs 50 `
>> --data "D:\Salman\Semesters\Sem6\Mini Project\Mini_Project_Dataset
\microplastic.yaml" `
>> --weights yolov5m.pt `
>> --hyp data/hyps/hyp.custom.yaml
```

```
# Detect
python detect.py --weights "D:\Salman\Semesters\Sem6\Mini Project
\yolov5_env\yolov5\runs\train\microplastic_yolo82\weights\best.pt"
--source "D:\Salman\Semesters\Sem6\Mini Project\Mini_Project_Dataset
\images\val\c--69-_jpg.rf.3957a3460762ebdef8b08e044289b229.jpg"
--img 640 --conf-thres 0.1
```

Bibliography

[1] Pensiri Akkajit, Md Eshrat E. Alahi, and Arsanchai Sukkuea. “Enhanced detection and classification of microplastics in marine environments using deep learning”. In: *Regional Studies in Marine Science* 80 (2024), p. 103880. ISSN: 2352-4855. DOI: https://doi.org/10.1016/j.rsma.2024.103880. URL: https://www.sciencedirect.com/science/article/pii/S2352485524005139.

[2] Sun Lanjun et al. “Rapid identification of marine microplastics by laser-induced fluorescence technique based on PCA combined with SVM and KNN algorithm”. In: *Environmental Research* 269 (2025), p. 120947. ISSN: 0013-9351. DOI: https://doi.org/10.1016/j.envres.2025.120947. URL: https://www.sciencedirect.com/science/article/pii/S0013935125001987.

[3] Jinhui Liu et al. “Recent advances in the detection of microplastics in the aqueous environment by electrochemical sensors: A review”. In: *Marine Pollution Bulletin* 214 (2025), p. 117695. ISSN: 0025-326X. DOI: https://doi.org/10.1016/j.marpolbul.2025.117695. URL: https://www.sciencedirect.com/science/article/pii/S0025326X25001705.

[4] Xiongfei Meng et al. “Identification of marine microplastics by a combined method of principal component analysis and random forest for fluorescence spectrum processing”. In: *Marine Pollution Bulletin* 214 (2025), p. 117740. ISSN: 0025-326X. DOI: https://doi.org/10.1016/j.marpolbul.2025.117740. URL: https://www.sciencedirect.com/science/article/pii/S0025326X25002152.

[5] Kalpana Patidar et al. "Assessing the microplastic pandemic: Prevalence, detection, and human health impacts in Asian aquatic environments". In: *Physics and Chemistry of the Earth, Parts A/B/C* 137 (2025), p. 103800. ISSN: 1474-7065. DOI: https://doi.org/10.1016/j.pce.2024.103800. URL: https://www.sciencedirect.com/science/article/pii/S1474706524002584.

[6] Md Abdul Baset Sarker et al. "Automatic Detection of Microplastics in the Aqueous Environment". In: *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*. 2023, pp. 0768–0772. DOI: 10.1109/CCWC57344.2023.10099253.

[7] Md Abdul Baset Sarker et al. "Real-Time Detection of Microplastics Using an AI Camera". In: *Sensors* 24.13 (2024). ISSN: 1424-8220. URL: https://www.mdpi.com/1424-8220/24/13/4394.