SCHOOL OF ELECTRONICS ENGINEERING
Winter Sem 2023-24

Title of the Project
OceanWatch

Student Name : Gourav Anand (21BEC0354)

Project Report
of
BECE 352E – IoT Domain Analyst

Submitted to
Faculty Name : Dr. R. Sujatha **Faculty Signature**
Date: 17/April/2024
Slot : L13+L14

Index

Abstract

"OceanWatch: Revolutionising Ocean Waste Management with YOLOv8"

The OceanWatch initiative, led by IoT OceanCleanse, introduces an innovative approach to monitoring and managing ocean garbage. Leveraging state-of-the-art sensor technology, powered by YOLOv8, the system enables real-time identification and classification of contaminants, providing crucial data to a central hub for informed decision-making. This paper explores the implementation of YOLOv8 in OceanWatch, highlighting its role in facilitating rapid response efforts and reducing environmental damage. Furthermore, the scalability and adaptability of YOLOv8 ensure the effectiveness of OceanWatch on a global scale, making significant strides towards mitigating ocean pollution and fostering sustainable marine ecosystems. Through ongoing research and collaboration, OceanWatch aims to redefine ocean waste management practices, setting a new standard for environmental stewardship and innovation in marine conservation.

Introduction

The growing concern over environmental pollution, particularly in marine ecosystems, has spurred extensive research into methods for identifying and mitigating sources of contamination. Among the most pervasive pollutants are marine debris and trash, which pose significant threats to marine life, ecosystems, and human health. Traditional methods of monitoring and managing oceanic waste have proven inadequate due to the vastness and complexity of marine environments. In response, emerging technologies, particularly those leveraging deep learning techniques, offer promising solutions for automated trash detection and characterisation.

Deep learning, a subset of artificial intelligence, has demonstrated remarkable capabilities in various computer vision tasks, including object detection and classification. One of the most notable deep learning architectures for object detection is YOLO (You Only Look Once), specifically its latest iteration, YOLOv8. YOLOv8 combines efficiency and accuracy, making it well-suited for real-time applications in complex environments.

In this research paper, we propose a novel approach to detecting and identifying trash in ocean environments using the YOLOv8 deep learning framework. Unlike traditional methods that rely on manual inspection or labor-intensive image processing techniques, our approach harnesses the power of deep learning to automate the detection process. By training a YOLOv8 model on a dataset of underwater images annotated with trash labels, we aim to develop a robust and efficient system capable of accurately identifying various types of marine debris.

The unique challenges posed by underwater environments, such as low visibility, varying lighting conditions, and complex backgrounds, necessitate innovative solutions for trash detection. Leveraging the capabilities of YOLOv8, we seek to address these challenges by designing a model that can effectively distinguish between marine debris and natural elements in ocean imagery.

Furthermore, our research aims to contribute to ongoing efforts in marine conservation and environmental monitoring by providing a reliable tool for assessing the prevalence and distribution of oceanic trash. By automating the detection process, our proposed system has the potential to streamline waste management practices, facilitate timely interventions, and ultimately mitigate the impact of marine pollution on aquatic ecosystems.

Through experimental validation and performance evaluation, we seek to demonstrate the effectiveness and reliability of our proposed approach in real-world oceanic environments. Additionally, we intend to explore the scalability and adaptability of our model to different geographical regions and environmental conditions, paving the way for broader applications in ocean conservation and sustainability efforts.

In summary, this research endeavours to leverage the capabilities of deep learning, particularly YOLOv8, to develop an innovative solution for trash detection in ocean environments. By combining advanced computer vision techniques with domain-specific knowledge of marine ecosystems, we aim to contribute to the preservation and protection of our oceans for future generations.

Prototype Implementation

To prototype the implementation outlined in the provided script, follow these steps:

1. **Environment Setup :**Set up a Python environment with necessary libraries such as OpenCV, Matplotlib, and Ultralytics.
   - Ensure GPU support for faster training and inference if available.

2. **Data Acquisition :** Collect or obtain a dataset containing underwater images and videos with labeled trash instances.
   - Organise the dataset into appropriate directories for training, validation, and testing.

3. **Model Configuration :** Define the configuration parameters for training the YOLOv8 model, including the number of epochs, batch size, image size, etc. Prepare a YAML file to specify the dataset paths and class names for training.

4. **Training:** Train the YOLOv8 model for trash instance segmentation using the provided dataset and configuration. Monitor the training progress and adjust hyper parameters as needed.

5. **Inference on Images:** Perform inference on a set of validation images using the trained model. Visualise the predicted segmentation masks and compare them with ground truth labels for evaluation.

6. **Inference on Videos :** Perform inference on a sample video file using the trained model. Save the predicted video frames and visualise them to observe trash detection in real-time.

7. **Evaluation:** Evaluate the performance of the trained model by measuring metrics such as precision, recall, and F1 score on the validation dataset. Analyse the results and identify areas for improvement.

8. **Iterative Development:** Iterate on the prototype by refining the model architecture, adjusting hyper-parameters, and augmenting the dataset to improve performance. Experiment with different pre-processing techniques, loss functions, and optimisation algorithms to enhance model accuracy and robustness.

9. **Documentation:** Document the prototype implementation process, including the dataset used, model architecture, training procedure, evaluation metrics, and results. Provide clear instructions and explanations for each step to facilitate reproducibility and future development.

10. **Deployment(Optional):** If desired, deploy the trained model for real-world applications, such as trash detection in underwater environments using autonomous or remotely operated vehicles.

By following these steps, you can successfully prototype the implementation of trash detection in ocean environments using YOLOv8, as outlined in the provided script.

Proposed System Flow Diagram:

1. **Data Collection:** Raw data is collected from various sources, including sensors, cameras, and other data acquisition devices deployed in the ocean environment.

2. **Preprocessing:** Raw data undergoes preprocessing to clean and enhance its quality, including noise reduction, calibration, and alignment.

3. **Trash Instance Segmentation:** The preprocessed data is inputted into the YOLOv8 model trained for trash instance segmentation.
   - The model identifies and segments trash instances within the input data, producing segmented images highlighting the detected trash.

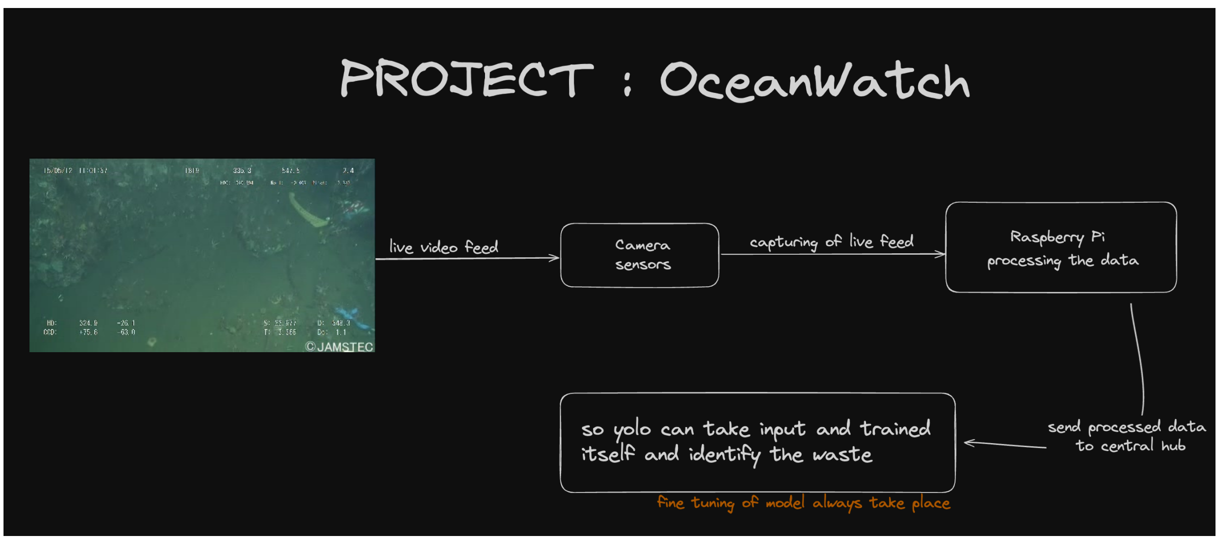
4. **Data Analysis:** Segmented images are analyzed to extract relevant information about the detected trash, including size, type, and location.

5. **Data Transmission:** Processed data, along with analysis results, are transmitted in real-time to a central hub using advanced communication technologies.

6. **Centralised Monitoring and Management:** At the central hub, stakeholders monitor the incoming data and analysis results in real-time. Decision-making algorithms may be applied to interpret the data and trigger appropriate responses.

7. **Response Planning and Execution:** Based on the monitored data and analysis results, stakeholders plan and execute prompt cleanup actions to mitigate environmental damage caused by ocean pollution. Actions may include deploying cleanup vessels, activating underwater drones, or organising beach cleanup initiatives.

8. **Feedback Loop:** Continuous monitoring and analysis of data facilitate iterative improvements to the system, including model retraining, algorithm optimisation, and resource allocation adjustments. Feedback from cleanup actions and environmental changes may also inform future data collection and analysis strategies.Prototype Implementation



Condition applied during training :

- **Task:** Segment
- Mode: Train
- **Model:** yolov8n-seg.pt
- **Data:** trashcan_inst_material.yaml
- Epochs: 5
- Batch Size: 16
- Image Size: 640x640
- Save Model: True
- Model Name: yolov8-seg
- Validation: True
- IOU Threshold: 0.7
- Initial Learning Rate: 0.01
- Momentum: 0.937
- Weight Decay: 0.0005
- Number of Batches: 64
- HSV Hue: 0.015
- HSV Saturation: 0.7
- HSV Value: 0.4

Results

Here, we demonstrate visually how well our YOLOv8 segmentation model for underwater object recognition performs. By presenting graphs, photos, and visualizations, we want to offer qualitative insights on the correctness and efficacy of our method.

1. Model Evaluation Metrics

-Recall Confidence Curve: 0.73
- Precision Confidence Curve: 1
- Precision-Recall Curve (mAP@0.5): 0.388
- F1 Confidence Curve: 0.40 at 0.266
- Recall Confidence Curve: 0.76
- Precision-Recall Curve (mAP@0.5): 0.399

2. Visual Representation

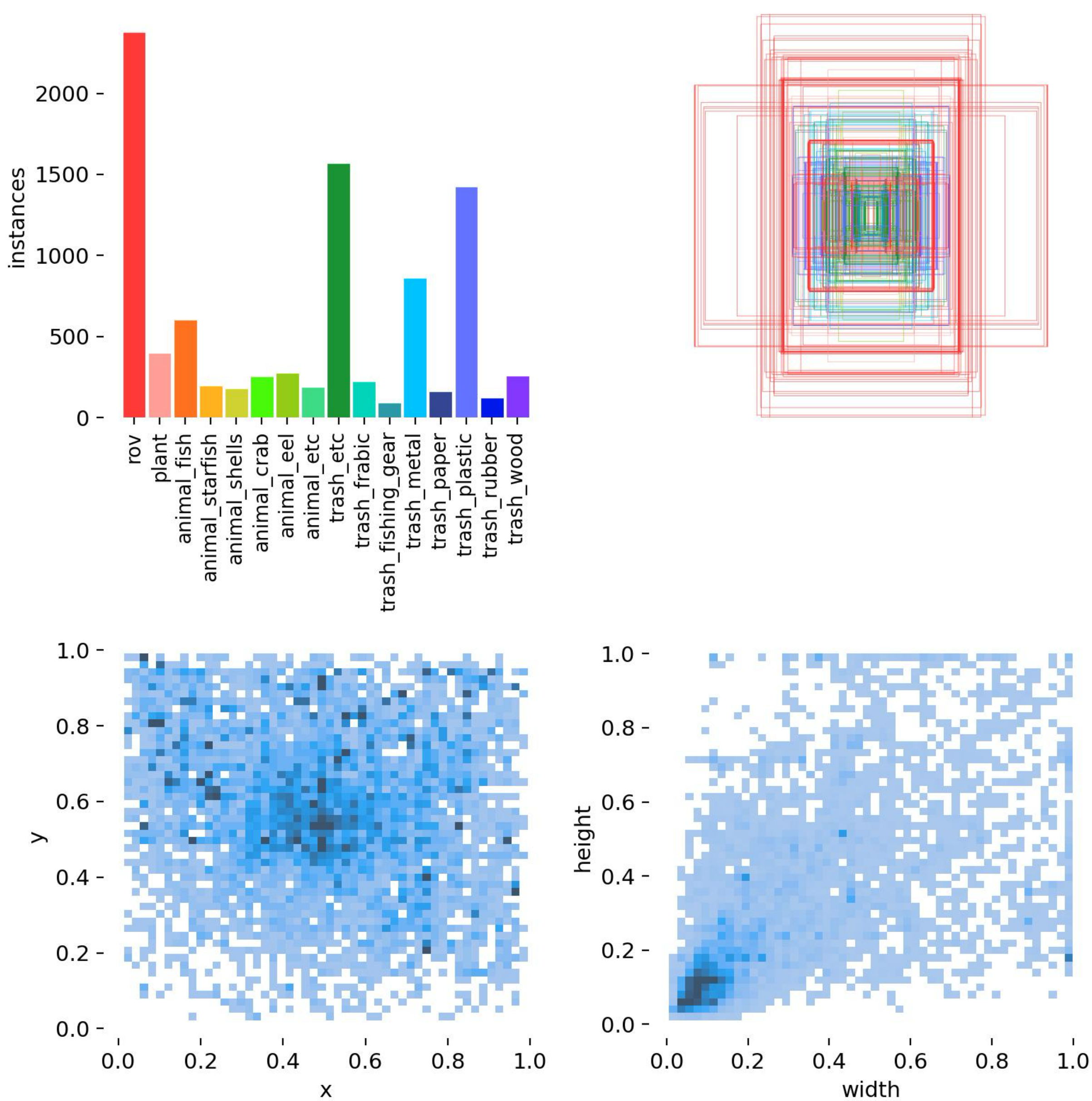
- We present visualizations including graphs and photos to illustrate the performance of our YOLOv8 segmentation model for underwater object recognition. These visual representations provide qualitative insights into the correctness and efficacy of our method.

3. Interpretation

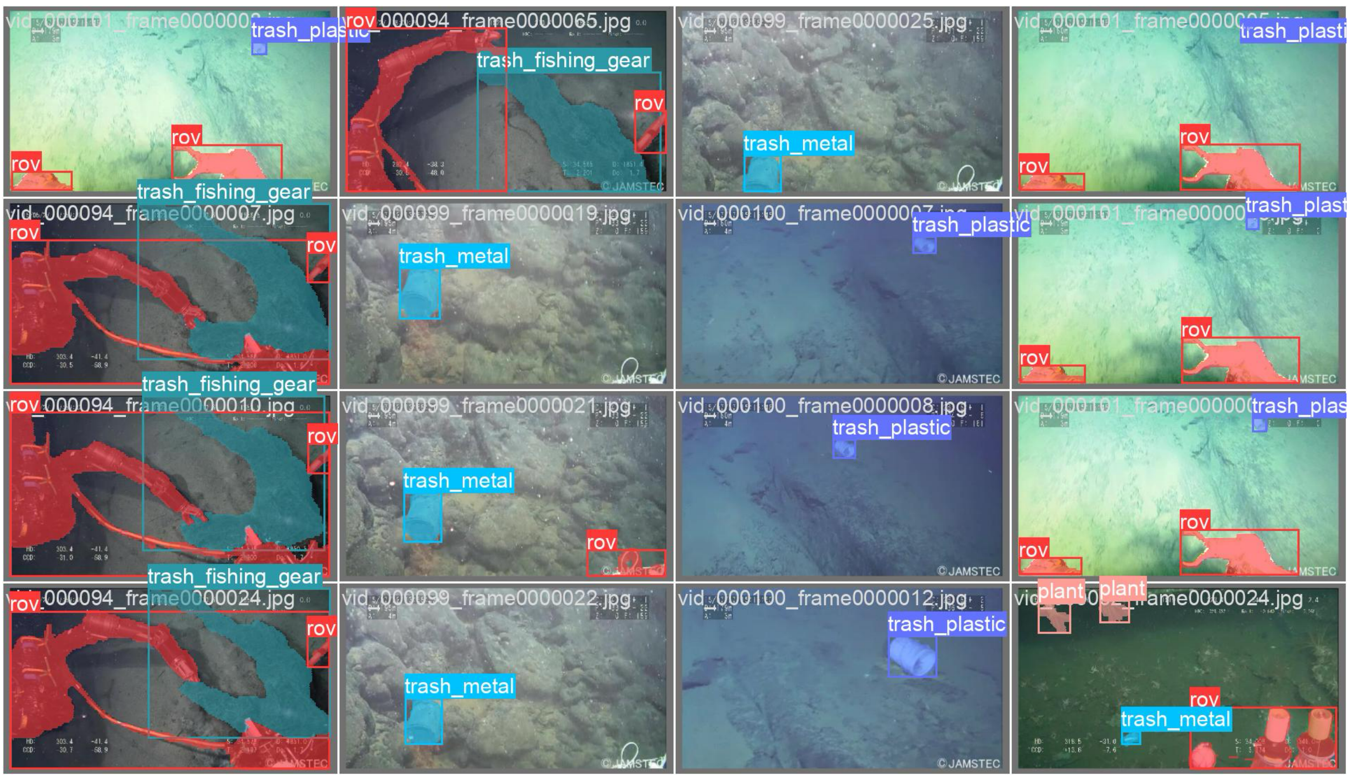
- The model demonstrates a recall confidence of 0.73, indicating its ability to correctly identify relevant instances.
- Precision confidence reached the maximum value of 1, suggesting high precision in positive instance predictions.
- The precision-recall curve yielded an average precision of 0.388 at a threshold of 0.5, reflecting the model's performance across various confidence levels.
- At a confidence threshold of 0.266, the F1 score was 0.40, indicating a balance between precision and recall.
- A recall confidence of 0.76 was observed, showcasing the model's improved ability to capture relevant instances.
- The precision-recall curve also indicated an average precision of 0.399 at a threshold of 0.5, reaffirming the model's consistency in performance.

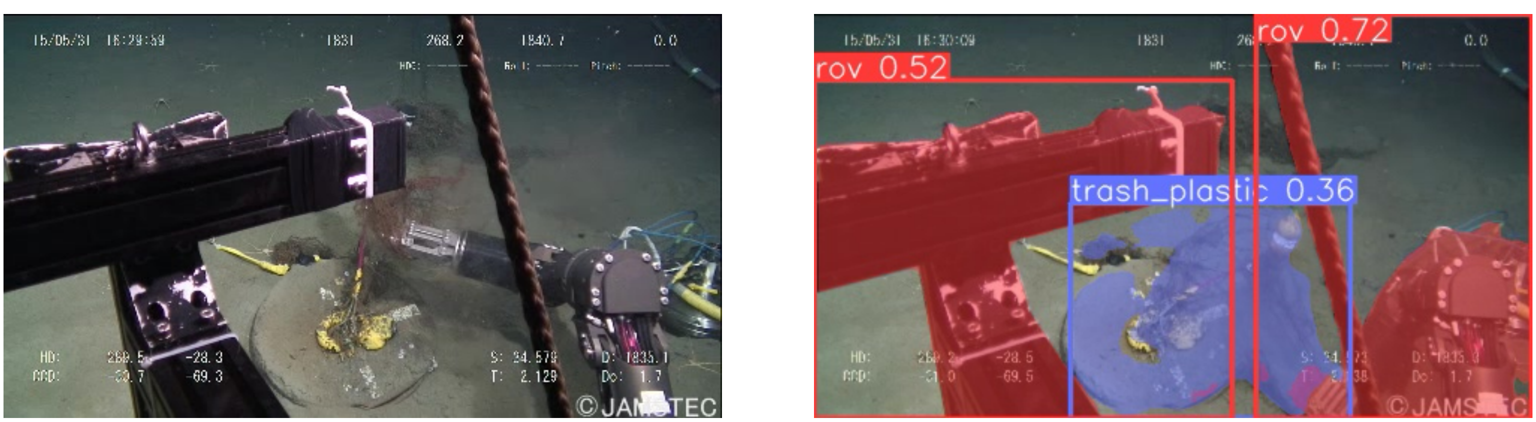
4. Discussion

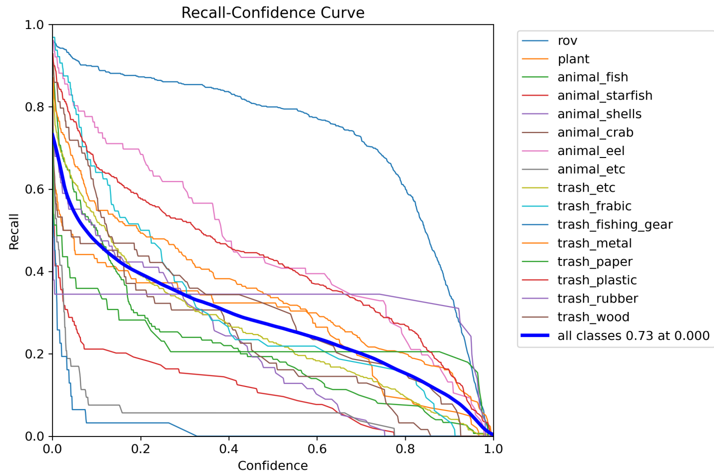
- The combination of quantitative metrics and visual representations offers a comprehensive understanding of the model's performance.
- The qualitative insights provided by the visualizations complement the quantitative results, enhancing the overall evaluation of our YOLOv8 segmentation model for underwater object recognition.

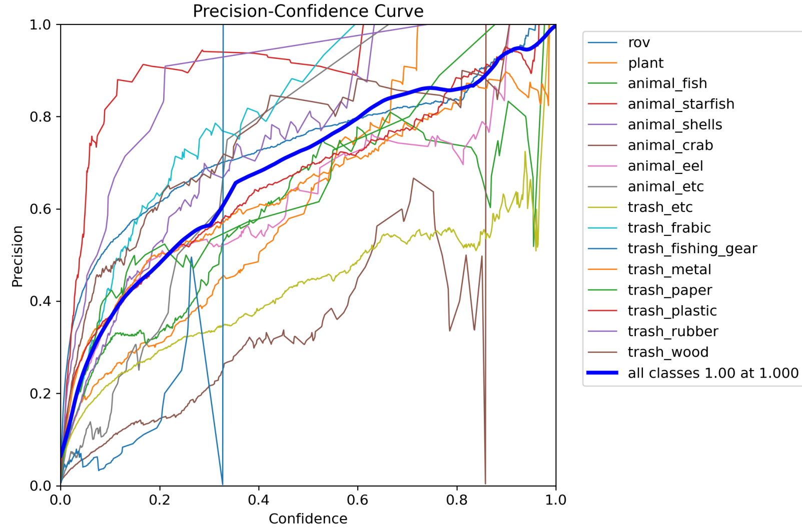


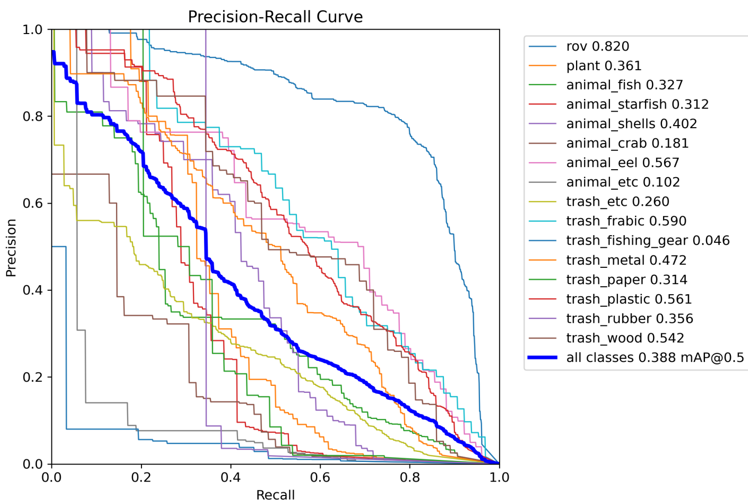
```
optimizer: AdamW(lr=0.0005, momentum=0.9) with parameter groups 66 weight(decay=0.0), 77 weight(decay=0.0005), 76 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/segment/yolov8-seg
Starting training for 5 epochs...

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
        1/5      8.08G      1.213      2.524      3.283      1.251         48        640: 100% 355/355 [03:02<00:00,  1.95it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100% 36/36 [00:15<00:00,  2.28it/s]
                   all       1149       2419      0.268      0.166      0.126     0.0788      0.265      0.163      0.128     0.0694

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
        2/5      5.84G       1.15      2.154      2.377      1.202         52        640: 100% 355/355 [02:55<00:00,  2.02it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100% 36/36 [00:14<00:00,  2.53it/s]
                   all       1149       2419      0.533      0.218      0.207      0.126      0.542      0.211      0.205      0.109

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
        3/5      8.27G      1.116      2.061      2.024      1.181         39        640: 100% 355/355 [02:52<00:00,  2.05it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100% 36/36 [00:14<00:00,  2.55it/s]
                   all       1149       2419      0.443      0.309      0.267      0.166      0.437      0.304      0.259      0.132

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
        4/5      8.28G      1.054      1.932      1.745      1.138         47        640: 100% 355/355 [02:57<00:00,  2.00it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100% 36/36 [00:13<00:00,  2.58it/s]
                   all       1149       2419      0.496      0.371      0.356      0.233      0.492      0.364      0.345      0.183

      Epoch    GPU_mem   box_loss   seg_loss   cls_loss   dfl_loss  Instances       Size
        5/5      8.26G     0.9904      1.819      1.518      1.111         40        640: 100% 355/355 [02:54<00:00,  2.04it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95)     Mask(P          R      mAP50  mAP50-95): 100% 36/36 [00:13<00:00,  2.60it/s]
                   all       1149       2419      0.499      0.391      0.399      0.261      0.502      0.387      0.389      0.206
```

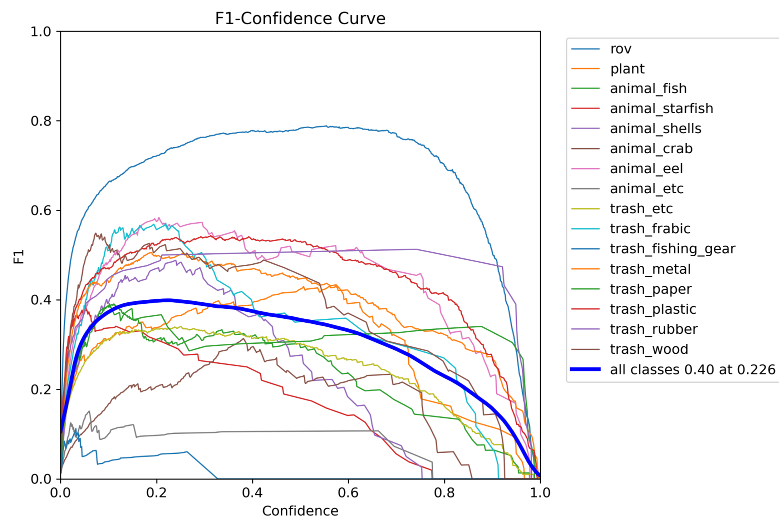


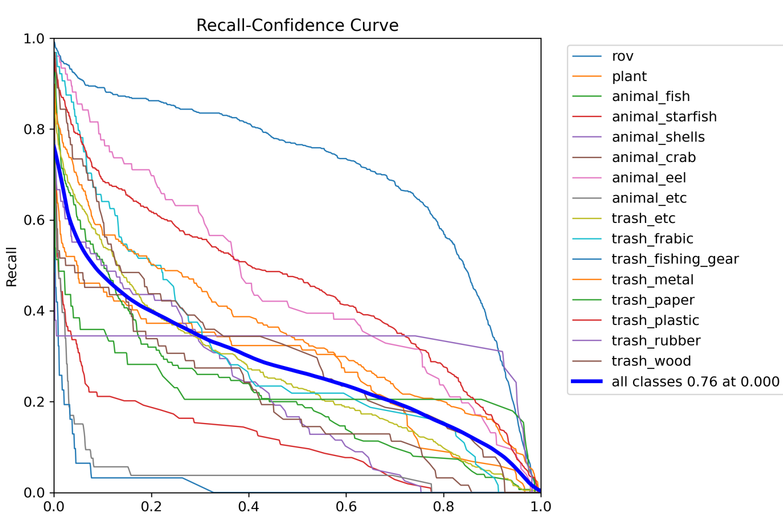


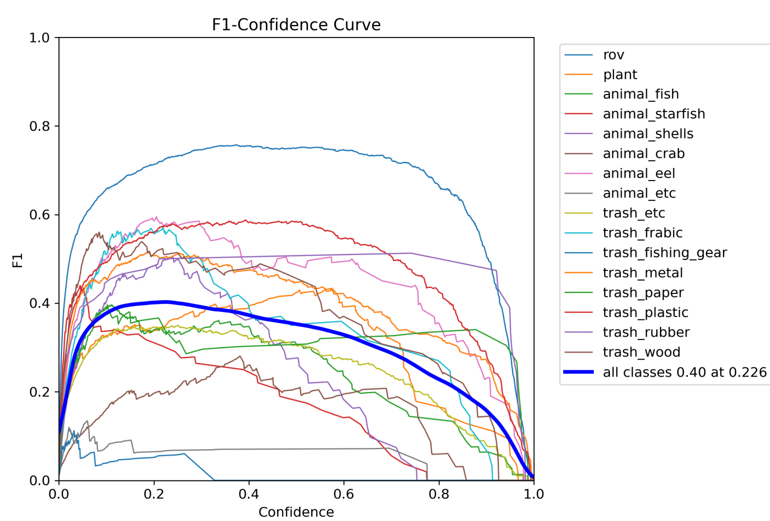

Recall-Confidence Curve
Recall
Confidence
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 0.73 at 0.000

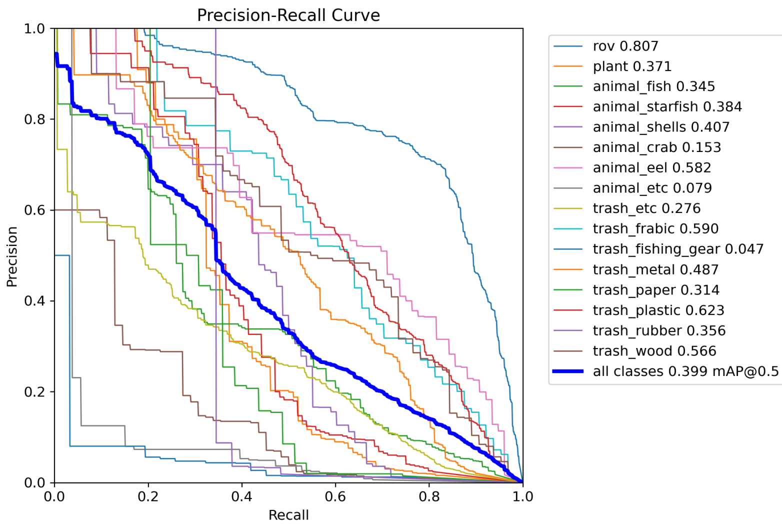

Precision-Confidence Curve
Precision
Confidence
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 1.00 at 1.000

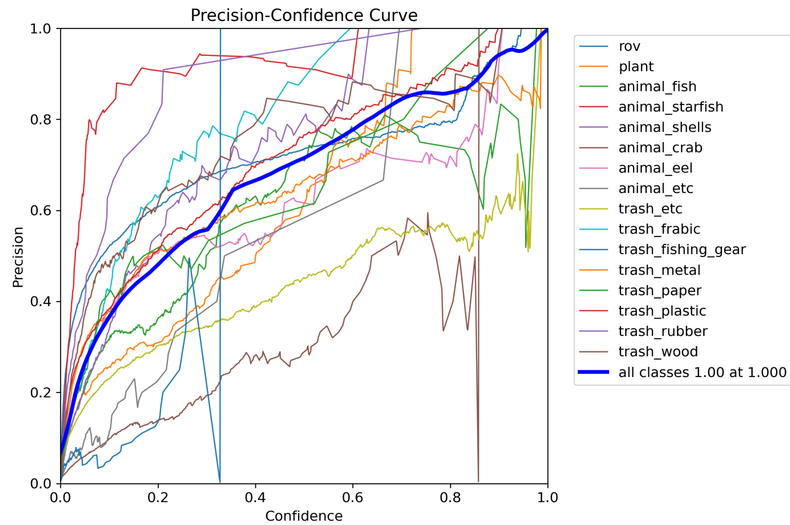

Precision-Recall Curve
Precision
Recall
rov 0.820
plant 0.361
animal_fish 0.327
animal_starfish 0.312
animal_shells 0.402
animal_crab 0.181
animal_eel 0.567
animal_etc 0.102
trash_etc 0.260
trash_frabic 0.590
trash_fishing_gear 0.046
trash_metal 0.472
trash_paper 0.314
trash_plastic 0.561
trash_rubber 0.356
trash_wood 0.542
all classes 0.388 mAP@0.5


F1-Confidence Curve
F1
Confidence
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 0.40 at 0.226

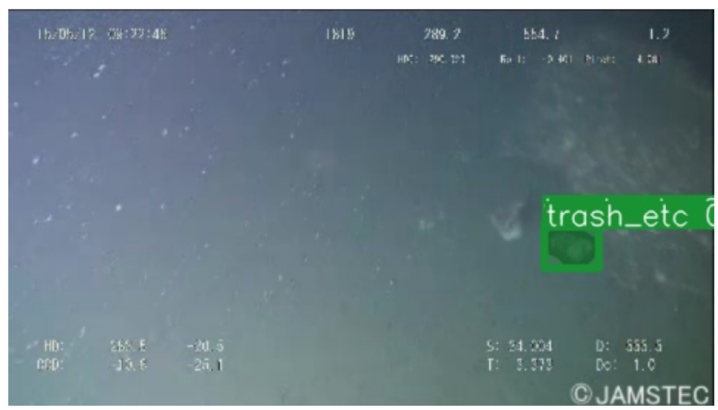

Recall-Confidence Curve
Recall
Confidence
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 0.76 at 0.000


F1-Confidence Curve
F1
Confidence
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 0.40 at 0.226


Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
rov 0.807
plant 0.371
animal_fish 0.345
animal_starfish 0.384
animal_shells 0.407
animal_crab 0.153
animal_eel 0.582
animal_etc 0.079
trash_etc 0.276
trash_frabic 0.590
trash_fishing_gear 0.047
trash_metal 0.487
trash_paper 0.314
trash_plastic 0.623
trash_rubber 0.356
trash_wood 0.566
all classes 0.399 mAP@0.5


Precision-Confidence Curve
Precision
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
rov
plant
animal_fish
animal_starfish
animal_shells
animal_crab
animal_eel
animal_etc
trash_etc
trash_frabic
trash_fishing_gear
trash_metal
trash_paper
trash_plastic
trash_rubber
trash_wood
all classes 1.00 at 1.000


trash_etc 0.28
©JAMSTEC


trash_etc
©JAMSTEC


08/06/09 13:13:40 00°
-53° -24° 2084 F074
091° 3408m 1m
trash_paper 0.87
©JAMSTEC


rov 0.76
rov 0.94
11/08/02 09:31:28
1306
359.1
2056.5
345.6
+25.9
-26.8
-46.3
trash_etc 0.51
©JAMSTEC

Reference

- Sai Chea, Zhongguo Li, Zhou Shi, Miaowei Gao, and Hongchuan Tang, "Research on an underwater image segmentation algorithm based on YOLOv8," *J. Phys.: Conf. Ser.*, vol. 2644, no. 1, p. 012013, [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2644/1/012013/pdf
- Jin Zhu, Tao Hu, Linhan Zheng, Nan Zhou, and Huilin Ge, "YOLOv8-C2f-Faster-EMA: An Improved Underwater Trash Detection Model Based on YOLOv8," [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/2644/1/012013/pdf
- Ms. Neha Singh, Palak Saini, and Deepali Yadav, "Underwater Marine Life Study Using YOLOv8," [Online]. Available: https://www.ijfmr.com/papers/2023/6/10565.pdf
- Sanyam Jain, "DeepSeaNet: Improving Underwater Object Detection using EfficientDet," [Online]. Available: https://arxiv.org/pdf/2306.06075.pdf
- Shubo Xu, Minghua Zhang, Wei Song, Haibin Mei, Qi He, and Antonio Liotta, "A systematic review and analysis of deep learning-based underwater object detection," *Robotics and Autonomous Systems*, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231223000656
- Minghua Zhang, Shubo Xu, and Wei Song, "Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion," [Online]. Available: https://www.researchgate.net/publication/356434469_Lightweight_Underwater_Object_Detection_Based_on_YOLO_v4_and_Multi-Scale_Attentional_Feature_Fusion
- Minghua Zhang, Zhihua Wang, Wei Song, Danfeng Zhao, and Huijuan Zhao, "Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network," *Appl. Sci.*, vol. 14, no. 3, p. 1095, 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/3/1095
- Rong Jia, Bin Lv, Jie Chen, Hailin Liu, Lin Cao, and Min Liu, "Underwater Object Detection in Marine Ranching Based on Improved YOLOv8," *J. Mar. Sci. Eng.*, vol. 12, no. 1, p. 55, 2024. [Online]. Available: https://www.mdpi.com/2077-1312/12/1/55
- Ayush Thakur, "An Improved Underwater Object Detection based on YOLOv8 Segmentation," 2024. [Online]. Available: https://www.researchgate.net/profile/Ayush-Thakur-9/publication/379226556_An_Improved_Underwater_Object_Detection_based_on_YOLOv8_Segmentation/links/65ffc742a8baf573a1d58327/An-Improved-Underwater-Object-Detection-based-on-YOLOv8-Segmentation.pdf
- Yonghui Huang, Qiye Zhuo, Jiyang Fu, and Airong Liu, "Research on evaluation method of underwater image quality and performance of underwater structure defect detection model," *Opt. Laser Technol.*, vol. 65, pp. 1-8, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0141029624003596