International Scientific Journal of Engineering and Management
Volume: 03 Issue: 04 | April – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

Harnessing Deep Learning for Underwater plastic Trash Identification

1st Pradyumna K
dept. Computer Science and Engineering (of Aff.)
Malnad College of Engineering (of Aff.)
Hassan,India
pradyumnasushanth@gmail.com

2nd Ravi Kumar A
dept.Computer Science and Engineering (of Aff.)
Malnad College of Engineering(of Aff.)
Hassan,India
ravikumar.uablr@gmail.com

3rd Palaniswamy I
dept.Computer Science and Engineering(of Aff.)Malnad College of Engineering(of Aff.)
Hassan,India
palaniswamyi9900@gmail.com

4th Pancham H N
dept.Computer Science and Engineering (of Aff.)
Malnad College of Engineering (of Aff.)
Hassan,India
panchamn2@gmail.com

5th Mr. Ravi Kumar D
dept.Computer Science and Engineering (of Aff.)Malnad College of Engineering (of Aff.) Assistant Professor
Hassan,India
rd@mcehassan.ac.in

***Abstract*—Machine and deep learning (DL) offer significant opportunities for exploring and mon- itoring oceans and for tackling important problems ranging from litter and oil spill detection to marine biodiversity estimation. Reasonably priced hardware platforms, in the form of autonomous (AUV) and remote operated (ROV) underwater vehicles, are also becoming available, fuelling the growth of data and offering new types of ap- plication areas. This article presents a research vision for DL in the oceans, collating applications and use cases, identifying opportunities, constraints, and open research challenges. We conduct experiments on underwater marine litter detection to demon- strate the benefits DL can bring to underwater envi- ronments. Our results show that integrating DL in underwater explorations can automate and scale-up monitoring, and highlight practical challenges in enabling underwater operations.**

This project introduces a refined YOLOv8-based algorithm tailored for the en- hanced detection of small-scale underwater debris, to mitigate the prevalent challenges of high miss and false detection rates . The research presents the YOLOv8 algorithm, which optimizes the backbone, neck layer, and C2f module for underwater characteristics and incorporates an effective attention mech- anism. This algorithm improves the accuracy of underwater trash detection while simplifying the computational model. Empirical evidence underscores the superiority of this method over the other conventional network, manifesting in a significant uplift in detection performance. Notably, the proposed method realized a 63% mean average precision (mAP50), a 60% surge in recall (R). Transcending its foundational utility in marine conservation, this methodology harbors potential for subsequent integra- tion into remote sensing ventures. Such an adaptation could substantially enhance the precision of detection models, particularly in the realm of localized surveillance, thereby broad- ening the scope of its applicability and impact.

I. Introduction

Today our environment not only faces pollution such as noise, and air but is also vastly spread to the very essence of life, water. An introduction of harmful materials into the environment is called pollution. Water pollution includes lakes, ponds, rivers, and oceans. A lot of trash objects like plastic and other wastes dumped in our ocean which pollutes and results in climate change as well. There are many cases where the ocean is dumped with hazardous materials by tankers and ships. Though there are control mechanisms available it becomes tough to track these, especially when dumped illegally into the coastal waters.The point to note is that all forms of pollution end up in the water which is a major cause of concern. Underwater Waste is a huge environmental concern that affects our marine and aquatic habitats and has a huge impact on our survival in the long run. The debris found in Marine not only includes non-biodegradable industrial waste, sewage, and radioactive material dumps but also plastic, etc. With the advancement of technology, we have looked at opportunities to utilize them to solve various problems of nature. Identifying objects deep in the bottom of the ocean is very difficult due to the challenges that nature poses in front of us. Using technology for identifying the challenge is a viable option. This is where object recognition plays a major part. Object recognition is the process of detect- ing and recognizing specific objects from image or videos. Machine learning methods are traditionally used to detect

ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

objects, but during recent times, with the advent of powerful Deep learning methods, their usage in object recognition has become widely popular. Histogram of Oriented Gradients (HOG), Support vector machine (SVM), etc. are some of the examples of machine learning algorithms popularly used for object detection. Deep learning, which is also known as deep machine learning, is gaining popularity at a rapid rate with its tremendous success in the object detection and recognition of digital images. Convolutional Neural Networks (CNNs), Regional CNNs, the You Only Look Once (YOLO) model, etc., are examples of some of the popular deep learning algorithms used in the recognition of objects. Underwater object detection is becoming an important area of research for its various applications such as inspection and understanding the underwater scenarios, ocean development in different fields of study. Since there is a drastic difference in the conditions underwater such as lighting and turbidity when compared with the conditions outside, it is of utmost importance to pick the right object detection and tracking technique among the various popular algorithms known, to gain optimal solutions.

II. Dataset Description

Trash-ICRA 19 dataset, the existing dataset which is a labeled dataset for underwater trash will be used to train the deep learning models.This dataset has been created from the J-EDI dataset of marine debris which is a collection of videos of underwater debris, provided by Japan Agency for Marine-Earth Science and Technology (JAMSTEC). The dataset contains a total of 5700 images, comprising various types of debris and water creatures in real-world environments. The dataset may contain a variety of images which has different environment, object sizes, background, or objects. To ensure the evaluation we have selected different subset of images for Faster R-CNN and YOLO algorithm.

III. Methodology

A. YOLOv8 architecture

YOLO (You Only Look Once) is one of the most popular modules for real-time object detection and image segmentation, currently (end of 2023) considered as SOTA State-of-The-Art. YOLO is a convulsional neural network that predicts bounding boxes and class probabilities of an image in a single evaluation.

Despite the undeniable efficiency of this tool, it is important to bear in mind that it was developed for a generalist context, aiming to serve the largest possible number of applications. However, for more specific cases requiring higher quality, speed, handling of non-standard images, among other scenarios.

Main Blocks

The first step to understanding the YOLO architecture is to understand that there are 3 essential blocks in the algorithm and everything will occur in these blocks, which are: Backbone, Neck and Head. The function of each block is described below.

Backbone:

Function: The backbone, also known as the feature ex- tractor, is responsible for extracting meaningful features from the input.

- Activities:
  i. Captures simple patterns in the initial layers, such as edges and textures.
  ii. Can have multiple scales of representation as you go, capturing features from different levels of abstraction.
  iii. Will provide a rich, hierarchical representation of the input.

Neck:

- Function: The neck acts as a bridge between the backbone and the head, performing feature fusion operations and integrating contextual information. Basically the Neck assembles feature pyramids by aggregating feature maps obtained by the Backbone, in other words, the neck collects feature maps from different stages of the backbone.
- Activities:
  i. Perform concatenation or fusion of features of different scales to ensure that the network can detect objects of different sizes.
  ii. Integrates contextual information to improve detection accuracy by considering the broader context of the scene.
  iii. Reduces the spatial resolution and dimensionality of resources to facilitate computation, a fact thatincreases speed but can also reduce the quality ofthe model.

Head:

- Function: The head is the final part of the network and is responsible for generating the network's outputs, such as bounding boxes and confidence scores for object detection.
- Activities:
  i. Generates bounding boxes associated with possible objects in the image.
  ii. Assigns confidence scores to each bounding box to indicate how likely an object is present.
  iii. Sorts the objects in the bounding boxes according to their categories.

Training

After preprocessing the data, 3 folders will be created. We will upload these folders to Google colab (Colab allows anybody to write and execute arbitrary python code through browser, and is especially well suited to machine learning,data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no set up to use while providing access free of charge to computing resources including GPUs). Here we will clone the YOLOv8 algorithm and give path to folders and train the model and it finishesand gives us accuracy. Once we give the pictures to trained model we will receive our results showing what kind of material. For this we need to perform data augmentation. Under water images are not easily available on the internet as

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 04 | April – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

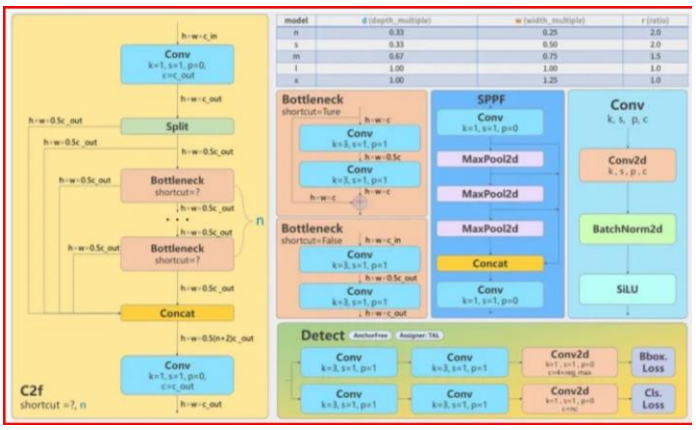


Fig. 1. Main Structures of The Main Blocks

the background of all the images are same, So due to which it is difficult to identify the images as we cannot locate any difference. Because of these factors the training model is not always effectively performed. Therefore, the dataset should be modified and augmented to make the deep learning model make generally used. The data augmentation is mainly based on rotational zooming.

The algorithm has to be configured to detect the trash objects underwater, before the training process. The configuration was done according. The configuration has been made in YOLO algorithm to be detect trash objects in an underwater environment. The number of classes (i.e. number of objects that the algorithm should detect) is 16 as trash_plastic, animal_fish, trash_etc, rov (remotedly operated vehicles), trash_metal etc.

```
attr = {
    'path':cwd+'/trash_inst_material',
    'train':'train/images',
    'val':'val/images',

    'names': {
        0:'rov',
        1:'plant',
        2:'animal_fish',
        3:'animal_starfish',
        4:'animal_shells',
        5:'animal_crab',
        6:'animal_eel',
        7:'animal_etc',
        8:'trash_etc',
        9:'trash_fabric',
        10:'trash_fishing_gear',
        11:'trash_metal',
        12:'trash_paper',
        13:'trash_plastic',
        14:'trash_rubber',
        15:'trash_wood',
    }
}
```

Fig. 2. Image showing no of classes

Validation and Testing

Validation is the process that is carried out after training, where the trained model is evaluated with a testing data set. Validating the algorithm outputs is important to ensure its accuracy to improve the quality and quantity. Without validating the model it is not right to rely on its prediction. Validating the Machine learning algorithm, result in making right predictions.

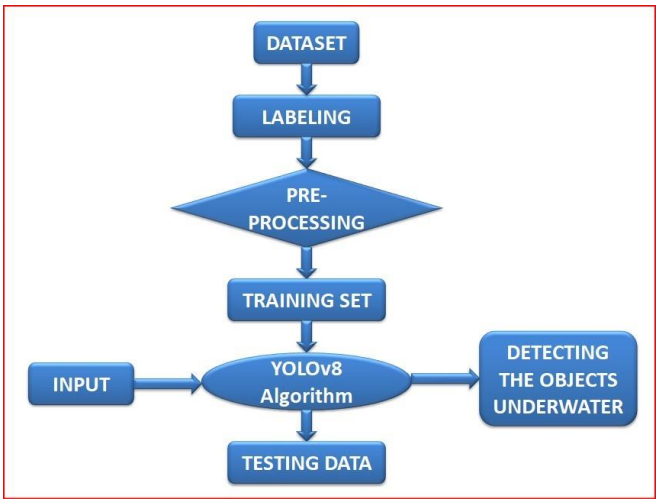
Some advantages of validation of the model are as follows:

- Scalability and flexibility
- Reduce the costs
- Enhance the model quality
- Discovering more errors
- Prevents the model from overfitting and underfitting.

Testing is done by performing experiments for distinct parameters and measuring the performance utilizing key metrics on an evaluation subset of the complete dataset. The preprocessing steps are applied to the testing images that are fed into the engine acquiring exemplar. This intends that several post processing may be needed to assess the exemplar and analyze the outcomes of the detection on the unique image. The test images were collected to evaluate which contained examples of every class in our model were selected for a test set.

The images and objects of videos network which are not to be used for training. So it ensures there are no overlap between the training and test set. Multiple types of plastic objects were categorized and selected, which included grocery bags, plastic bottles, etc. We ensured at least a minimum of 20 images per class were collected for each of these types of objects. We collected a sample of such images which were annotated for testing. The images became examples for each class, in a variety of environments that were collected intentionally and selected to be challenging for detectors, to ensure we provide a realistic evaluation of how these detectors would perform in field conditions.

Fig. 3. The overview of the experiment.



Performance Metrics

Performance metrics are key tools to evaluate the accuracy and efficiency of object detection models. They shed light on how effectively a model can identify and localize objects within images. Additionally, they help in understanding the

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 04 | April – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

model's handling of false positives and false negatives. These insights are crucial for evaluating and enhancing the model's performance. In this guide, we will explore various performance metrics associated with YOLOv8, their significance, and how to interpret them.

Object Detection Metrics

- Intersection over Union (IoU): IoU is a measure that quantifies the overlap between a predicted bounding box and a ground truth bounding box. It plays a fundamental role in evaluating the accuracy of object localization.
- Average Precision (AP): AP computes the area under the precision-recall curve, providing a single value that encapsulates the model's precision and recall performance.
- Mean Average Precision (mAP): mAP extends the concept of AP by calculating the average AP values across multiple object classes. This is useful in multi-class object detection scenarios to provide a comprehensive evaluation of the model's performance.
- Precision and Recall: Precision quantifies the proportion of true positives among all positive predictions, assessing the model's capability to avoid false positives. On the other hand, Recall calculates the proportion of true positives among all actual positives, measuring the model's ability to detect all instances of a class.
- F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a balanced assessment of a model's performance while considering both false positives and false negatives.

i. Class-wise Metrics

  – Class: This denotes the name of the object class, such as "person", "car", or "dog".
  – Images: This metric tells you the number of images in the validation set that contain the object class.
  – Instances: This provides the count of how many times the class appears across all images in the validation set.
  – Box(P, R, mAP50, mAP50-95): This metric provides insights into the model's performance in detecting objects:

1. P (Precision): The accuracy of the detected objects, indicating how many detections were correct.
2. R (Recall): The ability of the model to identify all instances of objects in the images.
3. mAP50: Mean average precision calculated at an intersection over union (IoU) threshold of 0.50. It'sa measure of the model's accuracy considering only the "easy" detections.
4. mAP50-95: The average of the mean average precision calculated at varying IoU thresholds, ranging from 0.50 to 0.95. It gives a comprehensive view of the model's performance across different levels of detection difficulty.

Visual Outputs

The model.val() function, apart from producing numeric metrics, also yields visual outputs that can provide a more intuitive understanding of the model's performance. Here's a breakdown of the visual outputs you can expect:

- F1 Score Curve (F1 curve.png): This curve represents the F1 score across various thresholds. Interpreting this curvecan offer insights into the model's balance between false positives and false negatives over different thresholds.
- Precision-Recall Curve (PR_curve.png): An integral visualization for any classification problem, this curve showcases the trade-offs between precision and recall at varied thresholds. It becomes especially significant when dealing with imbalanced classes.
- Precision Curve (P_ curve.png): A graphical representation of precision values at different thresholds. This curve helps in understanding how precision varies as the threshold changes.
- Recall Curve (R_curve.png): Correspondingly, this graph illustrates how the recall values change across different thresholds.
- Confusion Matrix (confusion_matrix.png): The confusion matrix provides a detailed view of the outcomes, showcasing the counts of true positives, true negatives, false positives, and false negatives for each class.
- Normalized Confusion Matrix (confu- sion matrix normalized.png): This visualization is a normalized version of the confusion matrix. It represents the data in proportions rather than raw counts. This format makes it simpler to compare the performanceacross classes.
- Validation Batch Labels (val_batchX_labels.jpg): These images depict the ground truth labels for distinct batches from the validation dataset. They provide a clear picture of what the objects are and their respective locations as per the dataset.
- Validation Batch Predictions (val _batchX _pred.jpg): Contrasting the label images, these visuals display the predictions made by the YOLOv8 model for the respective batches. By comparing these to the label images,you can easily assess how well the model detects and classifies objects visually.

IV. Results

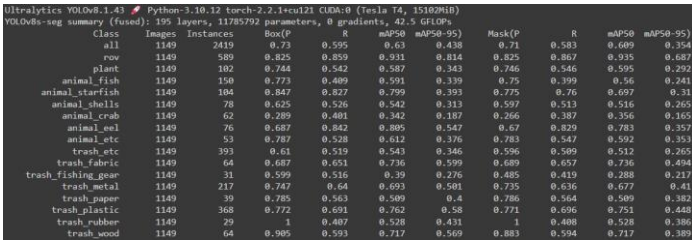
For evaluating the performance of the models, there are multiple standard metrics available such as F1 Score Curve, Precision-Recall Curve, Precision Curve and Recall Curve, Confusion Matrix, Mean average precision(mAP50). Among these metrics, the Mean average precision(mAP50) evaluation technique is the most widely used as it is quick to calculate and easily provides overall assessment of model quality.

The realm of biological detection, YOLOv8 presents initial benchmarks with Precision, Recall, and mAP@50 at modest levels of 73%, 59%, and 63%, respectively. This pronounced

improvement underscores our model's enhanced sensitivityand specificity in detecting biological features even with a small

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 04 | April – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

sample size. Focusing on ROV detection, YOLOv8 achieves a Precision of 83% and Recall of 85%, culminat-ing in a mAP@50 of 93%. Thereby evidencing its superior acumen in discerning ROV attributes. This analytical overview accentuates the bespoke capabilities of our model, particularly in its refined detection of biological components and ROV elements, heralding its versatility and potential applicability across a broad spectrum of marine object detection scenarios.

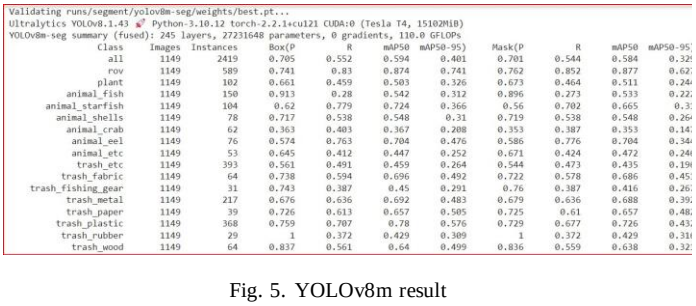


Ultralytics YOLOv8.1.43 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8s-seg summary (fused): 195 layers, 11785792 parameters, 0 gradients, 42.5 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95)
all	1149	2419	0.73	0.595	0.63	0.438	0.71	0.583	0.609	0.354
rov	1149	589	0.825	0.859	0.931	0.814	0.825	0.867	0.935	0.687
plant	1149	102	0.744	0.542	0.587	0.343	0.746	0.546	0.595	0.292
animal_fish	1149	150	0.773	0.409	0.591	0.339	0.75	0.399	0.56	0.241
animal_starfish	1149	104	0.847	0.827	0.799	0.393	0.775	0.76	0.697	0.31
animal_shells	1149	78	0.625	0.526	0.542	0.313	0.597	0.513	0.516	0.265
animal_crab	1149	62	0.289	0.401	0.342	0.187	0.266	0.387	0.356	0.165
animal_eel	1149	76	0.687	0.842	0.805	0.547	0.67	0.829	0.783	0.357
animal_etc	1149	53	0.787	0.528	0.612	0.376	0.783	0.547	0.592	0.353
trash_etc	1149	393	0.61	0.519	0.543	0.346	0.596	0.509	0.512	0.265
trash_fabric	1149	64	0.687	0.651	0.736	0.599	0.689	0.657	0.736	0.494
trash_fishing_gear	1149	31	0.599	0.516	0.39	0.276	0.485	0.419	0.288	0.217
trash_metal	1149	217	0.747	0.64	0.693	0.501	0.735	0.636	0.677	0.41
trash_paper	1149	39	0.785	0.563	0.509	0.4	0.786	0.564	0.509	0.382
trash_plastic	1149	368	0.772	0.691	0.762	0.58	0.771	0.696	0.751	0.448
trash_rubber	1149	29	1	0.407	0.528	0.431	1	0.408	0.528	0.386
trash_wood	1149	64	0.905	0.593	0.717	0.569	0.883	0.594	0.717	0.389

Fig. 4. YOLOv8s result

Encapsulate the empirical evidence of the model's adeptness in the identification of underwater refuse, a task critical for environmental monitoring. Within the ambit of well-illuminated conditions, the model manifests a detection precision approximating 0.89 for such debris. This marks a substantial refinement over the foundational YOLOv8n algorithm. Moreover, this enhanced detection capability is not compromised under the exigencies of low-light environments, where it continues to eclipse the performance of the original YOLOv8.



Validating runs/segment/yolov8m-seg/weights/best.pt...
Ultralytics YOLOv8.1.43 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
YOLOv8m-seg summary (fused): 245 layers, 27231648 parameters, 0 gradients, 110.0 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95
all	1149	2419	0.705	0.552	0.594	0.401	0.701	0.544	0.584	0.35
rov	1149	589	0.741	0.83	0.874	0.741	0.762	0.852	0.877	0.62
plant	1149	102	0.661	0.459	0.503	0.326	0.673	0.464	0.511	0.24
animal_fish	1149	150	0.913	0.28	0.542	0.312	0.896	0.273	0.533	0.22
animal_starfish	1149	104	0.62	0.779	0.724	0.366	0.56	0.702	0.665	0.3
animal_shells	1149	78	0.717	0.538	0.548	0.31	0.719	0.538	0.548	0.26
animal_crab	1149	62	0.363	0.403	0.367	0.208	0.353	0.387	0.353	0.14
animal_eel	1149	76	0.574	0.763	0.704	0.476	0.586	0.776	0.704	0.34
animal_etc	1149	53	0.645	0.412	0.447	0.252	0.671	0.424	0.472	0.24
trash_etc	1149	393	0.561	0.491	0.459	0.264	0.544	0.473	0.435	0.19
trash_fabric	1149	64	0.738	0.594	0.696	0.492	0.722	0.578	0.686	0.45
trash_fishing_gear	1149	31	0.743	0.387	0.45	0.291	0.76	0.387	0.416	0.26
trash_metal	1149	217	0.676	0.636	0.692	0.483	0.679	0.636	0.688	0.39
trash_paper	1149	39	0.726	0.613	0.657	0.505	0.725	0.61	0.657	0.48
trash_plastic	1149	368	0.759	0.707	0.78	0.576	0.729	0.677	0.726	0.43
trash_rubber	1149	29	1	0.372	0.429	0.309	1	0.372	0.429	0.31
trash_wood	1149	64	0.837	0.561	0.64	0.499	0.836	0.559	0.638	0.32

Fig. 5. YOLOv8m result

We have trained 3 versions of YOLOv8 model i.e., YOLOv8s, YOLOv8n, YOLOv8m all of them are trained for 40 epoches and the results come in different time as all three models differ in depth of layers. Therefore YOLOv8n having less number of layer produces result in less amount of time which is used for time critical ROV vehicles where accuracy is not the concern. Where as YOLOv8m takes a lot of time in training as well as getting the output so this model can be used for applications where accuracy is more important.

While analyzing the performance of the models that we have trained we found out that YOLOv8m tops with highest mAP@50 of 0.60

We have developed a dashboard for displaying the output using HTML, CSS, JAVASCRIPT and FLASK the results of the inputs (im- ages or video) will be displayed in a separate window as well as in the dashboard. All the predictions will be saved in a directory for future references the results shows us promising outputs regarding detecting of the classes that we have trained.

TABLE I
COMPARISON OF UNDERWATER PLASTIC TRASH IDENTIFICATION ALGORITHMS

Model	Precision	recall	mAP@50	mAP@50-95
YOLOv8n	0.73	0.53	0.60	0.41
YOLOv8s	0.73	0.59	0.63	0.43
YOLOv8m	0.70	0.55	0.60	0.40

Fig. 6. Output of Yolov8n and Yolov8s

CONCLUSION

The marine plastics is present in the epipelagic layers of the ocean. The results of the experiment show that the recent versions of the YOLO algorithm were able to efficiently predict the ocean plastics with increased speed and accuracy compared to other algorithms when given the image and video feed as input. Both YOLOv7 and YOLOv8 algo- rithms showed similar results in accuracy and speed with the YOLOv8-s showing a significant edge in performance over YOLOv7-n. The real-time results of both the algorithms can be improved by increasing the dataset and parameter tuning while training the algorithms. In the future, the YOLOv7 and YOLOv8 algorithms can be integrated into Deep Learning apps to test the performance, and integrate it with underwater robots or vehicles, aiding them to identify and remove the ocean plastics.

International Scientific Journal of Engineering and Management
Volume: 03 Issue: 04 | April – 2024
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata
ISSN: 2583-6129
DOI: 10.55041/ISJEM01491

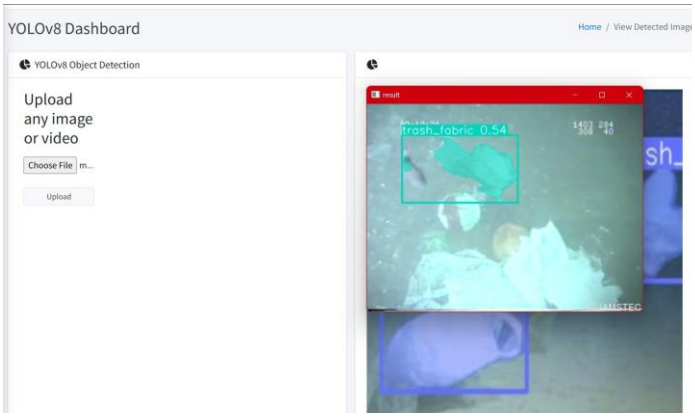


Fig. 7. Sample result using flask

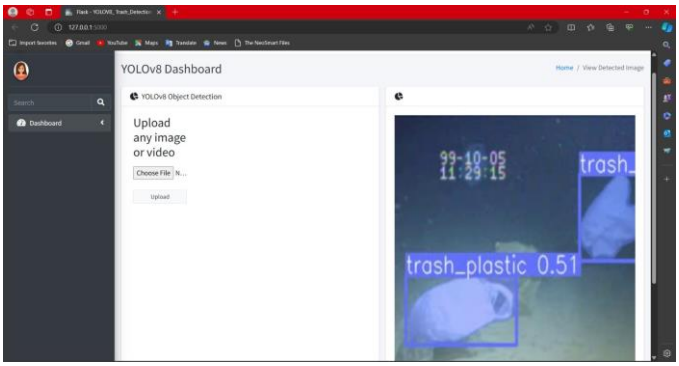


Fig. 8. Sample result using flask

This study is just a small part of the task, the improvised algorithm can be implemented along with other technologies to effectively remove marine plastics across the world

REFERENCES

[1] "Underwater object detection using collaborative weakly supervision", Cai, Sixian and Li, Guocheng and Shan, Yuan, Computers and Electrical Engineering, vol.102, pp.108159, 2022, Elsevier

[2] "Hybridization of deep convolutional neural network for underwater object detection and tracking model", Krishnan, Vijiyakumar and Vaiyapuri, Govindasamy and Govindasamy, Akila, Microprocessors and Microsystems, vol.94, pp.104628, 2022, Elsevier

[3] "Underwater object detection and tracking", Priyadarshni, Divya and Kolekar, MaheshKumar H, Soft Computing: Theories and Applications: Proceedings of SoCTA 2018, pp.837–846, 2020, Springer

[4] "Detection and tracking of objects in underwater video", Walther, Dirk and Edgington, Duane R and Koch, Christof, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol.1, pp.I–I, 2004, IEEE

[5] "You only look once: Unified, real-time object detection", Redmon, Joseph and Divvala, Santosh and Girshick, Ross and Farhadi, Ali, Proceedings of the IEEE conference on computer vision and pattern recognition, pp.779–788, 2016,

[6] "Underwater trash detection algorithm based on improved YOLOv5s", Wu, ChunMing and Sun, YiQian and Wang, TiaoJun and Liu, YaLi, Journal of Real-Time Image Processing, vol.19, num.5, pp.911–920, 2022, Springer

[7] "An optimized YOLO-based object detection model for crop harvesting system", Junos, Mohamad Haniff and Mohd Khairuddin, Anis Salwa and Thannirmalai, Subbiah and Dahari, Mahidzal, IET Image Processing, vol.15, num.9, pp.2112–2125, 2021, Wiley Online Library

[8] "Underwater marine life and plastic waste detection using deep learning and raspberry pi", Hegde, Rahul and Patel, Sanobar and Naik, Rosha G and Nayak, Sagar N and Shivaprakasha, KS and Bhandarkar, Rekha, Advances in VLSI, Signal Processing, Power Electronics, IoT, Communication and Embedded Systems: Select Proceedings of VSPICE 2020, pp.263–272, 2021, Springer

[9] "Real-time instance segmentation for detection of underwater litter as a plastic source", Corrigan, Brendan Chongzhi and Tay, Zhi Yung and Konovessis, Dimitrios, Journal of Marine Science and Engineering, vol.11, num.8, pp.1532, 2023, MDPI

[10] "Submerged marine debris detection with autonomous underwater vehicles", Valdenegro-Toro, Matias, 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA), pp.1–7, 2016, IEEE