**CHAPTER 01 INTRODUCTION**

**1. INTRODUCTION**

**1.1OVERVIEW**

Marine pollution, particularly the presence of diverse types of garbage in oceans, poses a significant threat to aquatic ecosystems and biodiversity. The impact of marine garbage on the environment is far-reaching, affecting not only marine animals but also coastal ecosystems and human health. Efforts to address the issue of marine garbage include clean-up initiatives, recycling programs, and regulations to prevent further pollution. Scientists and researchers are also developing technologies to detect and track marine garbage, allowing for better monitoring and management of the problem. By understanding the sources and impacts of marine garbage. We aim to explore a novel approach to enhance the accuracy and efficiency of marine debris detection by integrating the capabilities of the state-of-the-art You Only Look Once (YOLO) version 8 object detection model with advanced thresholding algorithms. The YOLO V8 model is renowned for its real-time object detection capabilities, making it suitable for dynamic environments such as oceans. However, the complexity of marine debris, which varies in size, shape, and texture, poses a challenge to traditional object detection models. To address this, we propose a fusion methodology that leverages the strengths of YOLO V8 in conjunction with thresholding algorithms to achieve more robust and precise detection across diverse garbage categories. By combining the speed and accuracy of YOLO V8 with the adaptability of thresholding techniques, our approach aims to overcome the limitations of existing methods in detecting submerged and floating marine debris. The research will contribute to the ongoing efforts to mitigate marine pollution by offering an innovative solution for automated detection and monitoring of diverse garbage in oceans. In last term, we focused on the fusion of different thresholding techniques with Thepade’s SBTC method to present a revolutionary technique for marine garbage identification. In order to further increase the detection's precision, we used machine learning algorithms like Random Forest, Random Tree, IBK , Kstar, LWL, ensemble of IBL Kstar, and LWL with Random Forest. We conducted tests on a dataset of underwater photos containing known trash items to assess the effectiveness of our technique. The dataset contained five labeled images. The performance of an ensemble of machine learning classifiers is superior to that of a single classifier. The study encouraged the creation of sustainable practices for maintaining the health and biodiversity of our oceans and adds to continuing efforts to reduce marine pollution.

**1.2 MOTIVATION**

Marine garbage poses a severe threat to marine ecosystems, causing harm to marine life through ingestion and habitat destruction. Water quality, seafood safety, and the general well-being of coastal communities can all be adversely affected by marine waste contamination, which can have far-reaching effects on human health.The financial effects of marine trash are significant and include harm to coastal infrastructure, fishery revenue losses, and tourism earnings. By detecting and monitoring garbage using YOLOv8, we can actively mitigate these impacts, preserving fragile ecosystems and ensuring the sustainability of ocean resources for future generations.YOLOv8's ability to identify and track garbage objects enables environmentalists to evaluate the degree of pollution and put specific conservation measures into place, protecting hotspots for biodiversity and endangered animals from more harm.

**1.3 PROBLEM STATEMENT AND OBJECTIVE**

To design an automated system that can detect and identify the garbage in the ocean to monitor and reduce the impact of marine garbage pollution. The system should be able to detect the presence of different types of garbage, such as plastic, paper, metals, etc. with the fusion of YOLO V8 and thresholding algorithms.

* To perform accurate detection of objects.
* Develop a robust object detection system.
* Develop a real-time system to detect marine debris using YOLOv8.
* Model Training and Optimization using the YOLOv8 to accurately detect various types of marine debris.

**1.4 SCOPE OF WORK**

The scope for the topic of detection and monitoring of marine garbage is broad and encompasses various aspects related to identifying, tracking, analyzing, and mitigating marine debris. Here are some potential areas that could be included within this scope:

* Types of Garbage: Plastic pollution, cardboard,cans,tin,fishing gear, oil spills, etc.
* Environmental Impacts: Impact on ecosystems, wildlife, health, etc.
* Clean-up Strategies: Methods for removing debris from oceans.
* Prevention Efforts: Waste management, recycling, education.
* Technological Innovations: New solutions like machine learning and biodegradable materials.
* Environmental Protection: Safeguard marine ecosystems and wildlife from pollution.
* Resource Managemesssnt: Efficiently allocate cleanup resources based on detected debris concentrations.
* Scientific Research: Contribute valuable data for studying pollution trends and ecosystem dynamics.
* Long-term Sustainability: Establish a framework for continuous monitoring and improvement in marine debris management.

**CHAPTER 02 LITERATURE SURVEY**

**2. LITERATURE SURVEY**

The previous research presents an effective small-object recognition method using an enhanced YOLOv8 network that is specifically designed for underwater photos. The method improves on the YOLOv8 design, which helps it overcome the problems posed by small objects and complicated underwater surroundings, which are a problem for traditional object detection algorithms [1]. The improvements targeted at enhancing the network's underwater small object detection accuracy and efficiency are feature recalibration, context aggregation, and channel-wise attention techniques. The suggested strategy outperforms current approaches, as seen by the experimental results, which also show faster processing rates and more accuracy. This development allows for more accurate and effective detection of small items in difficult aquatic conditions, which has promise for a variety of underwater applications, including environmental monitoring, marine research, and underwater robots.

[2]Another research presents a unique method using Multi-Scale Feature Adaptive Weighted Fusion for small-target water-floating rubbish detection, called APM-YOLOv7. By integrating the Adaptive Feature Pyramid Module (APM) with YOLOv7, the method improves the system's capacity to identify small objects in aquatic situations. By adapting the feature pyramid structure to the target size, APM makes it possible to increase detection accuracy. Furthermore, to optimize information fusion at various scales, the suggested weighted fusion mechanism dynamically modifies feature map weights dependent on object sizes. Based on experimental results, APM-YOLOv7 performs more accurately and efficiently than baseline approaches, especially when it comes to recognizing little waste pieces that float in water. Because it makes it possible to detect and remove water pollutants more effectively, the method has the potential to improve environmental monitoring and management efforts.

[3] An Intelligent Debris Mass Estimation Model (IDMEM) specifically designed for Autonomous Underwater Vehicles (AUVs) is proposed in this research. Because of their limited sensing capabilities and the dynamic nature of underwater environments, AUVs have difficulty precisely calculating the amount of trash. To improve the accuracy of debris mass estimation, IDMEM combines machine learning with sensor data, such as inertial and acoustic measurements. To minimize computing complexity and maximize prediction accuracy, the model makes use of a novel feature selection technique. As compared to conventional approaches, the experimental results reveal enhanced mass estimate accuracy, demonstrating the usefulness of IDMEM. The capabilities of IDMEM have potential applications in environmental evaluation, disaster response, and underwater debris monitoring. By improving AUV's capabilities in managing underwater debris, the article helps to make underwater operations safer and more effective.

[4] The study suggests a novel technique for utilizing a machine vision system with a small dataset to detect marine plastic waste underwater. It makes use of deep transfer learning, a method that uses a tiny dataset to adapt a pre-trained model to a new job. Despite the lack of training examples, the system can recognize plastic garbage underwater by utilizing this strategy. This approach is especially useful in situations when gathering big datasets is difficult. The study highlights the potential for technology to support environmental monitoring and protection by demonstrating the viability of utilizing cutting-edge machine learning algorithms in marine conservation initiatives. To protect marine ecosystems, the research emphasizes how important it is to locate and remove garbage from oceans. Using convolutional neural networks (CNNs), the authors suggest a model that can identify debris in underwater photos with accuracy[5]. They address difficulties unique to detecting underwater debris, like fluctuations in brightness and distorted images, and provide ways to address these problems. The procedure for gathering datasets, annotating techniques, and specifics of model training are described in the paper. The efficacy of the suggested strategy is demonstrated by the evaluation results, which show high debris detection accuracy. The significance of further research in this field is emphasized in the study's conclusion, which addresses environmental issues and offers workable options for monitoring and cleaning marine ecosystems.

[6] Visual Marine Debris Detection using Yolo 5s for Autonomous Underwater Vehicle: The technique presented in this research, which is designed specifically for Autonomous Underwater Vehicles (AUVs), uses YOLO (You Only Look Once) version 5s to detect marine debris underwater. There are serious environmental risks associated with marine trash, and prompt detection is essential to mitigating efforts. Because of YOLO 5s' effectiveness in real-time object identification, it was selected. The AUV's photos are pre-processed as part of the system, and YOLO 5s are then trained on datasets of photographs of maritime trash that have been labeled. During underwater missions, the trained model is subsequently installed on the AUV for real-time detection. The outcomes of the experiments show how well the suggested method works to precisely identify marine debris under various underwater circumstances. By locating and possibly eliminating these, this technique gives autonomous maritime vehicles a useful tool to support environmental conservation initiatives.

[7] A Deep Learning Marine Debris Detection Network: The technology effectively finds trash in underwater photos by utilizing the YOLO (You Only Look Once) architecture, which is important for environmental preservation initiatives. The model's architecture is optimized for real-time processing, making it possible to identify marine litter quickly. By improving detection speed and accuracy over earlier techniques, YOLOTrashCan facilitates the prompt removal and handling of marine garbage. Extensive experimentation on a variety of datasets reveals the better performance of the framework in detecting different kinds of marine rubbish in a variety of underwater habitats, proving its effectiveness. YOLOTrashCan provides a scalable way to track and reduce ocean pollution by automating the identification of debris, supporting international conservation efforts. This idea has the potential to advance efforts to protect aquatic biodiversity and enhance the health of marine ecosystems.

[8]Marine debris detection model based on the improved YOLOv5:The YOLOv5 model is improved and specifically designed for marine debris detection in this research. It tackles the problem of precisely recognizing different kinds of marine debris in photos taken underwater. To better fit the features of underwater images, the suggested model makes enhancements to the YOLOv5 architecture, such as changes to the backbone network, anchor aspect ratios, and augmentation approaches. To aid in model training and evaluation, the research also presents a newly curated dataset designed especially to detect marine debris. The improved YOLOv5 model outperforms baseline techniques, exhibiting greater accuracy and robustness in identifying marine trash objects, according to thorough testing and assessments.The efficacy of the model in practical situations is emphasized, demonstrating its potential uses in environmental monitoring, marine conservation initiatives, and cleanup projects. Overall, the study offers a workable strategy for resolving environmental issues in underwater ecosystems and represents a substantial improvement in the field of marine trash identification.

[9]A Study on the Implementation of Real-Time Marine Deposited Waste Detection AI System and Performance Improvement Method by Data Screening and Class Segmentation .The creation and improvement of a real-time AI system for identifying waste that has been dumped in the ocean is examined in this research. Its main goal is to increase system performance by using techniques for class segmentation and data filtering. The paper begins by outlining the difficulties in detecting marine garbage and emphasizes the significance of real-time detection technologies. It then goes into detail about the process, which includes class segmentation to precisely classify waste categories and data screening to remove unnecessary information. The AI system performs significantly better after applying data screening and class segmentation techniques, according to a number of metrics used to assess its performance. The effectiveness of these methods in improving the precision and efficacy of real-time marine waste detection systems, supporting environmental conservation initiatives and safeguarding marine ecosystems, is highlighted in the paper's conclusion.

[10]Marine Debris Detection in Satellite Surveillance using Attention Mechanisms The study suggests a novel method for employing attention mechanisms in satellite surveillance to detect marine garbage. It tackles the serious environmental issue of marine garbage, which endangers ecosystems and marine life. By prioritizing interesting locations in satellite images, the suggested strategy improves the precision and effectiveness of debris identification by utilizing attention mechanisms. The approach improves overall performance by cutting down on false positives and computational overhead by concentrating on pertinent areas. The approach's usefulness is demonstrated by the experimental findings, which show superior performance above previous methods. Furthermore, interpretability is made possible by the attention mechanism, which facilitates a deeper comprehension of the model's decision-making procedure. All things considered, this study offers a viable method for automatically identifying marine garbage, with potential uses in environmental monitoring and conservation initiatives.

.

**CHAPTER 03 SOFTWARE REQUIREMENTS SPECIFICATION**

**3.SOFTWARE REQUIREMENTS**

**3.1 FUNCTIONAL REQUIREMENT**

**3.1.1.** **Image Acquisition:** a. The system can accept digital images in common formats (e.g., JPEG, PNG) as input for tampering detection.

**3.1.2.** **Preprocessing:** Preprocess the acquired underwater images or video streams to enhance visibility, reduce noise, and optimize them for input to the YOLO v8 model.

**3.1.3. Object Classification:** The system not only detect objects but also classify them into predefined categories or classes, providing information about the type of objects detected. Besides just detecting objects, the system should also categorize them into predefined classes or categories.

**3.1.4. Multiple Object Detection:** The system is able to detect multiple objects of different classes simultaneously within a single image or video frame.

**3.1.5. Object Detection Using YOLO v8:** Implement the YOLO v8 object detection model to identify and localize underwater garbage objects within the processed images or video streams. Ensure that the model is trained and fine-tuned on underwater garbage datasets to improve detection accuracy**.**

**3.1.6. Localization and Size Estimation:** Localize and estimate the size of detected underwater garbage objects to provide additional information for assessment and decision-making.

**3.2 External Interface Requirements**

**3.2.1.User Interface (UI):** The system have a user-friendly interface for users to interact with, allowing them to configure settings, initiate detection processes, and view results. It may include features such as a graphical user interface (GUI) with options for uploading dataset(images or videos), and visualizing detected objects .

**3.2.2.Hardware Interface:** The system is compatible with various hardware components such as, GPUs (Graphics Processing Units), and processors. It should utilize hardware efficiently to accelerate processing speed and optimize performance, especially for real-time applications.

**3.2.3 Software Interface:** The system requires integration with software libraries, frameworks, or APIs (Application Programming Interfaces) for object detection and classification. It should support interoperability with other software components or systems, enabling seamless data exchange and communication. Compatibility with programming languages Python and development environments (e.g., TensorFlow, PyTorch) necessary for building and deploying the system.

# 

# 

# 3.4 Nonfunctional Requirements:

**3.4.1 Performance**: The system is efficient in terms of processing speed and resource usage to deliver results swiftly without overwhelming hardware.

**3.4.2 Reliability**: It provides consistent and trustworthy results, minimizing errors such as false positives or false negatives.

**3.4.3.Scalability**: The system's architecture allows it to grow seamlessly to handle increased loads or requirements.

**3.4.4.Usability**: The system is easy for users to interact with, with clear interfaces and intuitive controls.

**3.4.5.Maintainability**: It is straightforward to update, debug, and enhance, with well-organized code and documentation..

**3.4.6.Compatibility**: It works smo othly across different hardware and software configurations. The system provide insights into its own performance, allowing administrators to monitor and optimize its operation. The system holds to relevant laws, regulations, and ethical guidelines governing object detection and data processing.

**3.4.7.Detection Accuracy:** The system can accurately detect objects within a given dataset or real-time video feed, achieving a predefined level of accuracy in identifying various classes of objects in given image. The system can correctly identify objects within images or video feeds with a specified level of accuracy.

**3.4.8.** **Robustness:** The system is robust and resilient to variations in underwater conditions such as water clarity, lighting, and environmental disturbances, ensuring consistent performance in different underwater environments**.**

**3.5.System Requirements**

**3.5.1. Database requirement:**

To implement marine garbage detection using YOLO v8 (You Only Look Once version 8), we typically used a dataset of images or videos containing marine environments with garbage present. Along with the dataset, you used suitable hardware and software infrastructure for training, deploying, and running the YOLO v8 model.

Here we have used TrashCan 1.0 as a dataset. The dataset contains 6008 training instances and 1204 validation instances. There are a total of 15 classes in the dataset. They are:

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',

**3.5.2.Software requirements:**

* **Deep Learning Framework:** YOLO v8 is typically implemented using deep learning frameworks such as TensorFlow or PyTorch. Ensure you have the necessary framework installed along with any required dependencies.
* **Image Processing Libraries:** Libraries like OpenCV may be necessary for preprocessing images, resizing, augmenting, and normalizing them before training.
* **Development Environment:** Set up a development environment with your preferred IDE (Integrated Development Environment) such as PyCharm, VS Code, or Jupyter Notebook for coding and experimentation.
* **Training Infrastructure:** Access to hardware accelerators like GPUs or TPUs for faster model training. Cloud platforms like Google Cloud Platform (GCP), Amazon Web Services (AWS), or Microsoft Azure provide instances with GPU support.
* **Evaluation Metrics**: Implement evaluation metrics such as precision, recall, and mean average precision (mAP) for assessing the performance of your model during training and testing phases.
* **Documentation Tools:** Use tools like Markdown, LaTeX, or Google Docs for documenting your work, including project specifications, dataset details, model architecture, training procedures, and evaluation results. Hardware Requirements:

**3.5.3. Hardware Requirements:**

**GPU or TPU:** A dedicated GPU or TPU is essential for training deep learning models efficiently. GPUs from NVIDIA (e.g., GeForce, Tesla) or TPUs from Google Cloud are commonly used for accelerating deep learning tasks.

**Memory (RAM):** Sufficient RAM is required to handle large datasets and model parameters during training. Depending on the dataset size and model complexity, a minimum of 16GB RAM is recommended, although more may be necessary for larger projects.

**Storage:** Adequate storage space is necessary for storing datasets, model checkpoints, logs, and intermediate results. SSDs (Solid State Drives) are preferred for faster data access during training and inference.

**CPU:** While model training heavily relies on GPU or TPU acceleration, a decent CPU (Central Processing Unit) is still necessary for managing data preprocessing, I/O operations, and coordinating tasks during training and inference.

**Network Connectivity:** Reliable internet access is required, especially if you're using cloud-based resources for training, downloading datasets, or deploying models.

**3.6. SDLC Model**

For a project like marine garbage detection using YOLO v8, a suitable Software Development Life Cycle (SDLC) model would be the Agile model, specifically tailored to accommodate the iterative and collaborative nature of deep learning projects. Here's how the Agile model can be applied:

* **Requirements Gathering and Analysis**: Define the objectives and scope of the object detection system, including the types of marine garbage to be detected, environmental conditions, and performance requirements. Gather requirements from stakeholders, such as environmental agencies, marine conservation organizations, and end-users. Analyze the gathered requirements to identify key features, constraints, and technical specifications for the system.
* **Design**: Design the architecture of the object detection system, considering factors such as data acquisition, preprocessing, model selection (YOLO v8), post-processing, and integration with other components. Define the data flow, interfaces, and interactions between system modules. Develop detailed design documentation, including diagrams, models, and specifications to guide implementation.
* **Implementation:** Implement the object detection system according to the design specifications, utilizing YOLO v8 as the core detection algorithm. Develop software components for data acquisition, image preprocessing, model training, inference, and result visualization. Write clean, modular, and maintainable code adhering to coding standards and best practices.
* **Testing**: Conduct unit testing to verify the functionality of individual software components and modules. Perform integration testing to ensure seamless interaction between system components. Validate the accuracy and performance of the YOLO v8 model through extensive testing on diverse datasets representing various marine environments and garbage types. Implement regression testing to detect and prevent software regressions during iterative development.
* **Deployment**: Prepare the object detection system for deployment in the target environment. Configure hardware infrastructure computing resources, to support system operation. Install and configure software dependencies, libraries, and frameworks required for running the object detection system. Conduct deployment testing to ensure that the system functions correctly in the production environment and meets performance expectations.
* **Operation and Maintenance**: Monitor the performance and reliability of the object detection system in real-world conditions, addressing any issues or anomalies promptly. Provide user training and documentation to support system operation and usage by operators and stakeholders. Establish protocols for system maintenance, including software updates, bug fixes, and hardware maintenance tasks. Collect feedback from users and stakeholders to identify areas for improvement and optimization.
* **Evaluation and Evolution**: Evaluate the effectiveness and impact of the object detection system in detecting marine garbage and supporting environmental monitoring efforts. Analyze performance metrics, such as detection accuracy, false positive rates, and processing speed, to assess system performance. Iterate on system improvements based on evaluation results, user feedback, and advancements in object detection technologies. Plan for future enhancements, updates, and scalability to ensure the continued relevance and effectiveness of the object detection system over time.

# CHAPTER 04 SYSTEM DESIGN

# 4.SYSTEM DESIGN

**4.1PROPOSED SYSTEM:**

YOLO (You Only Look Once) is a popular set of object detection models used for real-time object detection and classification in computer vision. The model family belongs to one-stage object detection models that process an entire image in a single forward pass of a convolutional neural network (CNN). The key feature of YOLO is its single-stage detection approach, which is designed to detect objects in real-time and with high accuracy. Unlike two-stage detection models, such as R-CNN, that first propose regions of interest and then classify these regions, YOLO processes the entire image in a single pass, making it faster and more efficient. so how the architecture works is a few-step process.

Input image/videos: Being the first step, the input images and videos are taken in the form of frames and are being pre-processed. The image is converted into a grid of cells, each cell predicts bounding boxes.

Non-maximum suppression: Next on the converted image, in YOLOv8, like its predecessors, Non-Maximum Suppression (NMS) plays a crucial role in refining object detection results.YOLOv8 generates multiple bounding box predictions for each cell in the grid. Due to this overlap, there can be redundant and overlapping bounding boxes around the same object. NMS acts as a discerning editor, selecting the most confident and non-overlapping bounding boxes for each detected object. Its goal is to remove redundancy and retain only the most accurate predictions.

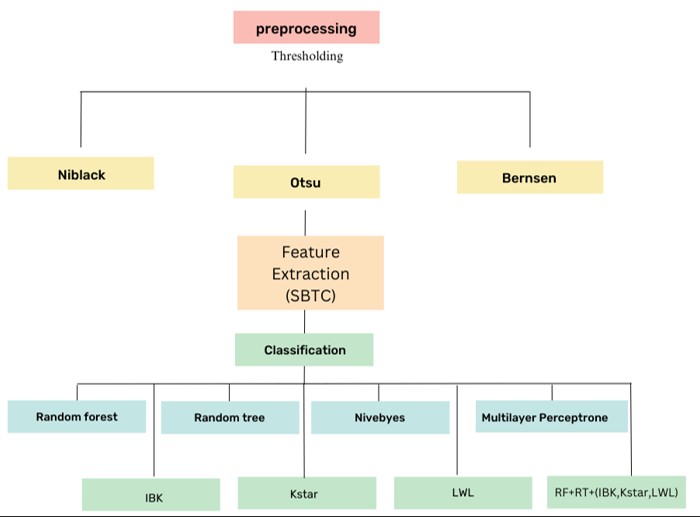
YOLOv8 comes in five variants based on the number of parameters – nano(n), small(s), medium(m), large(l), and extra large(x). All of the variants are used for classification, object detection, and segmentation. In our model, the YOLOv8(x) gives the maximum accuracy.

In computer vision, neural networks typically concentrate on the recognition of edges in the primary layer, forms in the center of the specimen, and work characteristics in the later layers. The beginning and middle stages are the only ones used in deep learning; the latter tiers are just reprogrammed. It makes use of the labelled data from the task that served as its practice environment.

The algorithm for SBTC:

Several of transfer learning's key advantages are the reduction of training stage, improved neural network performance (in most circumstances), and indeed the lack of a considerable quantity of data. Deep learning is helpful in circumstances during which it is typically impossible to obtain the significant quantities of information necessary to train a neural network from scratch.

The system architecture for the SBTC algorithm. The system architecture begins with the pre-processing, Pre-processing is a crucial step which involves transforming raw data into a format that is suitable for further analysis or modeling. Pre-processing can involve a wide range of techniques, such as data cleaning, normalization, transformation, feature selection, and feature extraction. This component would involve the preparation of the input images. We have used three thresholding techniques that are: Bernsen, Niblack and Otsu for the comparison of which technique gives the best result for preprocessing. The next step is featuring extraction, Feature extraction is a process in which relevant and informative features are selected or extracted from raw data, such as images, text, or signals. The goal of feature extraction is to reduce the dimensionality of the data and transform it into a more compact representation. It will include Thepade’s SBTC (Sorted block truncation coding) algorithm for extracting feature. Then for classification here we have used the classification algorithm to classify the extracted feature. We have used the Weka tool for classification of the object using classifiers Random Forest, Random Tree, IBK (image based keyer), Kstar, LWL (Locally weighted learning).  Here we are working on a labeled dataset, which is used for testing the fusion of algorithms. Basically algorithms are fusion of Niblack, Otsu and Bernsen with feature extraction algorithm SBTC. Then different machine learning classifiers are applied to these extracted features. The result analysis is performed by utilizing the different classification algorithm with some ensemble algorithms.



**Fig 1. SBTC Architecture**

**4.2ARCHITECTURE**

Input Images

Preprocess

Input

Videos

Frames

Bounding box

Non Max

Suppression

Final Visuals

**Fig 2. Architecture of YOLO V8 model**

**4.3 Dataset:**

Here we have used TrashCan 1.0 as a dataset. The dataset contains 6008 training instances and 1204 validation instances. There are a total of 15 classes in the dataset.They are:

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’,

Here are some images of dataset:



**Fig 3. Dataset image that shows garbage such as fabric in ocean**



**Fig 4. Dataset image that shows garbage such as wood in ocean**



**Fig 5**. **Dataset image that shows garbage such as plastic pipes in ocean**

****

**Fig 6.** **Dataset image that shows garbage such as tin can in ocean**

**CHAPTER 05 PROJECT PLAN**

**5.PROJECT PLAN**

**5.1RISK MANAGEMENT:**

* Model Performance and Reliability:

Risk: Inaccurate detection of object like marine animals can be detected as garbage.

Mitigation: Regularly evaluate detection performance, implement validation mechanisms, include dataset containing image of marine animals differentiated from garbage.

* Environmental Impact:

Risk: Unintended negative consequences in the ocean or sea that leads to damage of the model.

Mitigation: Conduct assessment for environmental regulations.

* Safety Concerns:

Risk: Deployment in dataset , corruption in data.

Mitigation: Implement safety protocols, provide training, use protective software .

**5.2 RISK ANALYSIS:**

Data Quality:

* + Risk: Inaccurate detections due to poor quality or insufficient training data.
  + Mitigation: Ensure diverse and representative datasets, augment data, and perform thorough preprocessing.

Environmental Conditions:

* + Risk: Adverse weather or water conditions affecting model performance.
  + Mitigation: Design robust algorithms, conduct testing under varied conditions, and implement real-time monitoring.
* Hardware Reliability:
  + Risk: Malfunction or failure of equipment (e.g., drones, sensors).
  + Mitigation: Regular maintenance, redundant systems, and contingency plans.
* Localization Accuracy:
  + Risk: Inaccurate geo-referencing of detected objects.
  + Mitigation: Use precise localization techniques, validate coordinates, and incorporate error estimation.
* Data Privacy and Security:
  + Risk: Breach of sensitive location data or privacy concerns.
  + Mitigation: Encrypt data, anonymize information, and comply with privacy regulations.
* Community Engagement:
  + Risk: Lack of community involvement leading to mistrust.
  + Mitigation: Engage stakeholders, communicate benefits, address concerns transparently, and involve local communities.
* Regulatory Compliance:
  + Risk: Non-compliance with monitoring regulations.
  + Mitigation: Stay informed, obtain permits, and collaborate with regulatory agencies.
* Biases in Detection:
  + Risk: Biases leading to misidentification or underrepresentation of marine garbage.
  + Mitigation: Evaluate model performance, address biases in training data, and employ data augmentation.

**5.2.1 OVERVIEW OF RISK MITIGATION:**

* Data Management:
  + Collect diverse and high-quality data.
  + Keep datasets updated and free from biases.
* Algorithm Design:
  + Build algorithms that work well in different environments.
  + Test algorithms thoroughly under various conditions.
* Equipment Maintenance:
  + Regularly maintain hardware to prevent failures.
  + Have backup systems in place for emergencies.
* Accuracy Verification:
  + Use precise methods to confirm the accuracy of detections.
  + Continuously monitor and adjust for errors.
* Data Security:
  + Encrypt data to keep it safe from unauthorized access.
  + Anonymize sensitive information to protect privacy.
* Stakeholder Engagement:
  + Involve all stakeholders and address their concerns openly.
  + Collaborate with local communities and regulatory agencies.
* Regulatory Compliance:
  + Understand and follow all relevant laws and regulations.
  + Obtain necessary permits and approvals before starting operations.
* Bias Mitigation:
  + Regularly check for biases in the data and algorithms.
* Use techniques to ensure fair and accurate results.

**CHAPTER 06 PROJECT IMPLEMENTATION**

**6.PROJECT IMPLEMENTATION**

**6.1OVERVIEW OF PROJECT MODULES**

Data Collection and Annotation: Gathering diverse datasets of marine environments containing garbage using various sources like drones, underwater cameras, or satellite imagery. Annotating images or videos to label marine garbage objects for training the YOLOv8 model .We have used TrashCan as a dataset

Preprocessing: Enhancing the quality of input data through techniques like noise reduction, image stabilization, and color correction.

Augmenting the dataset to increase diversity and improve model robustness.

Model Selection and Integration: Choosing YOLOv8 as the object detection model due to its real-time performance and accuracy. Integrating the YOLOv8 architecture into the project environment.

Training: Training the YOLOv8 model on the annotated dataset to recognize various types of marine garbage. Optimizing model hyperparameters and architecture for improved performance.

Evaluation: Assessing the model's performance using metrics such as precision, recall, and mean Average Precision (mAP).Validating the model's generalization ability on a separate dataset.

Deployment: Implementing the trained YOLOv8 model for real-time or batch processing of marine garbage detection. Integrating the model into a deployable application or platform.

Monitoring and Alerting: Establishing continuous monitoring capabilities to detect new instances of marine garbage in real time.

Integration with Environmental Systems:

Integrating the marine garbage detection system with existing environmental monitoring networks or platforms. Sharing detected garbage data with relevant organizations for further analysis and action.

Maintenance and Updates:Establishing procedures for maintaining and updating the detection system over time.Addressing issues like model drift and concept shift to ensure continued effectiveness.

**6.2TOOLS AND TECHNOLOGIES USED**

* YOLOv8: This is the main tool for spotting marine garbage in images or video.
* Data Preprocessing Tools: Image/Video Processing: Tools to prepare images/videos for analysis.
* Training Infrastructure: Powerful Computers- Machines with strong GPUs for training YOLOv8. The inference experiments were run on a laptop with an i7 8th generation CPU, 6 GB GTX 1060 GPU, and 16 GB RAM.
* Data Annotation Tools: Labeling Software: Tools for marking garbage in images.

**6.3ALGORITHM DETAILS**

The algorithm we have used here is YOLOv8,it is the latest version of the YOLO system developed by Ultralytics. It is faster and more accurate and it provides an unified framework for training models for performing object detection, image segmentation. YOLO stands for You Only Look Once. It’s a popular set of object detection models designed for real-time object detection and classification. Unlike two-stage detection models (such as R-CNN), which first propose regions of interest and then classify them, YOLO processes the entire image in a single pass. This makes it faster and more efficient.YOLO aims to achieve high accuracy in object detection while maintaining real-time speed.It has evolved from previous versions (YOLOv1, YOLOv2, YOLOv3) with improvements in accuracy and speed.

Key features of YOLOv8 include its single-stage detection approach and efficient processing of the entire image in one pass. The model family belongs to one-stage object detection models that use a convolutional neural network (CNN) for detection. Inan a simplified way how yolov8 works is ,it resizes the input image to a fixed size (e.g., 448 × 448 pixels), runs a single convolutional network on the image. Thresholds the resulting detections based on the model’s confidence score.

YOLOv8 models come in different sizes (YOLOv8n, YOLOv8s, YOLOv8m each with varying accuracy and speed trade-offs. The results have shown that the YOlLv8M has giving the maximum accuracy with and MAP score of 45%.

The system architecture for SBTC begins with the pre-processing, Pre-processing is a crucial step that involves transforming raw data into a format that is suitable for further analysis or modeling. Pre-processing can involve a wide range of techniques, such as data cleaning, normalization, transformation, feature selection, and feature extraction. This component would involve the preparation of the input images. We have used three thresholding techniques that are: Bernsen, Niblack, and Otsu for the comparison of which technique gives the best result for preprocessing. The next step is featuring extraction, Feature extraction is a process in which relevant and informative features are selected or extracted from raw data, such as images, text, or signals. The goal of feature extraction is to reduce the dimensionality of the data and transform it into a more compact representation. It will include Thepade’s SBTC (Sorted block truncation coding) algorithm for extracting features. Then for classification here we have used the classification algorithm to classify the extracted feature. We have used the Weka tool for the classification of the object using classifiers Random Forest, Random Tree, IBK (image-based keyer), Kstar, and LWL (Locally weighted learning).

 Here we are working on a labeled dataset, which is used for testing the fusion of algorithms. Basically, algorithms are a fusion of Niblack, Otsu, and Bernsen with feature extraction algorithm SBTC. Then different machine learning classifiers are applied to these extracted features. The result analysis is performed by utilizing the different classification algorithms with some ensemble algorithms.

**CHAPTER 07 RESULTS**

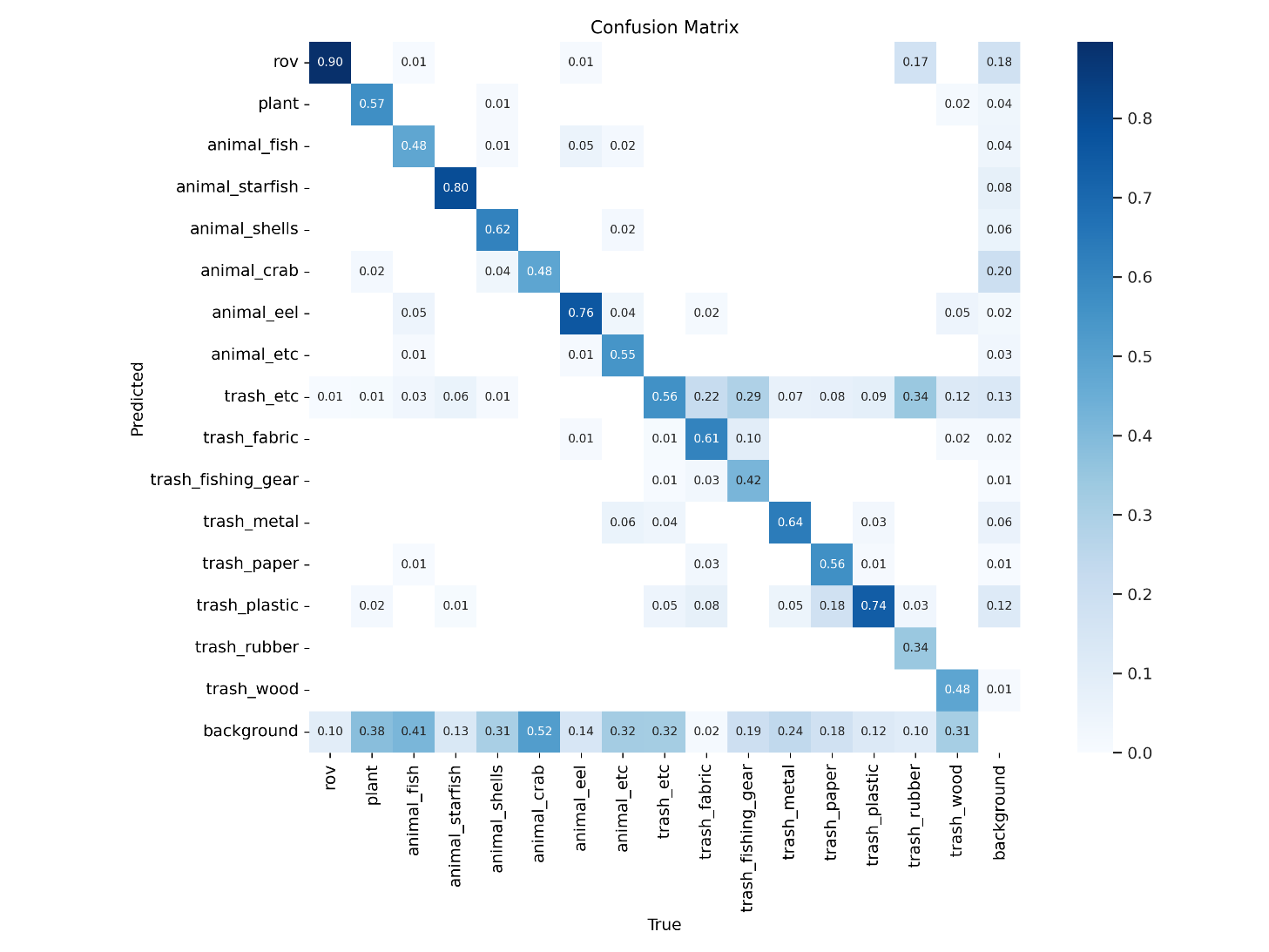
**7**.**RESULTS**

**7.1OUTCOMES**

We have trained the dataset with three different Yolo v8 models that are yolo nano, yolo small and Yolo medium. The Nano model reaches box mAP of 42.6% and segmentation mAP of 34.5% on the last epoch. Whereas, The Small model reaches slightly higher metrics within the same number of epochs. This time, the last epoch’s box mAP is 44.38%, and the segmentation mask mAP is 35.16%. With the YOLOv8 Medium model, we have the highest box mAP yet of 45%. Also, the segmentation mask mAP reaches 36.2%.

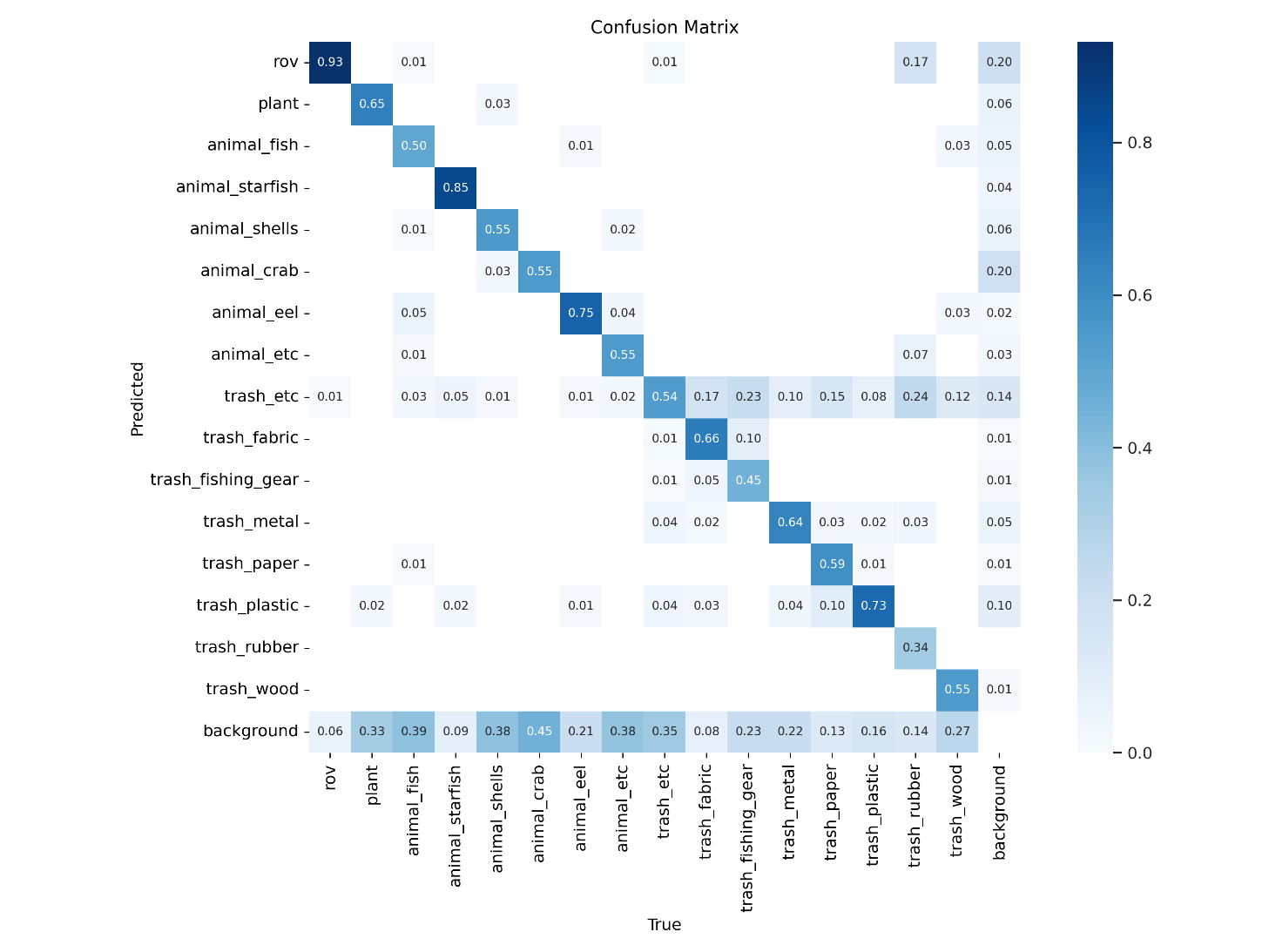
**7.2 RESULT AND ANALYSIS**

After comparison with all the YOLO models the comparison result shows that the YOLOv8m(medium) gives the maximum MAP score of 45%.Analyzing these results, it's clear that as the model size increases from Nano to Small to Medium, there's a corresponding improvement in performance. This improvement is particularly noticeable in the box mAP, where YOLO Medium achieved the highest score. The segmentation mask mAP also shows a similar trend, indicating that the larger model can better segment objects from the background.Overall, these results suggest that increasing the model size and complexity leads to better performance in terms of object detection and segmentation accuracy. However, it's essential to consider factors such as computational resources and deployment requirements when selecting the appropriate model for a specific application.

Nano:

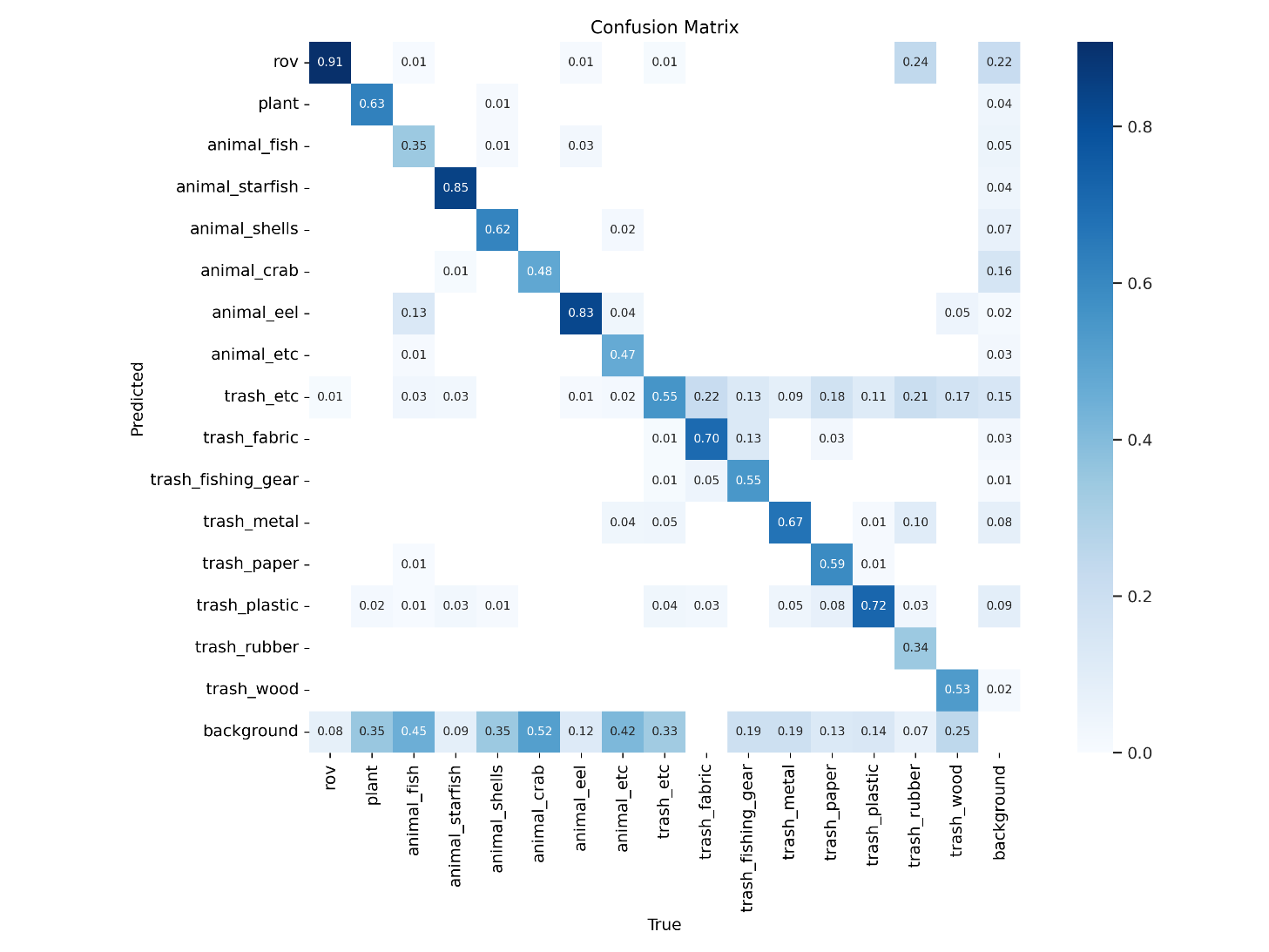
**Fig 7.Confusion matrix of YOLO V8 nano**

Small:



**Fig 8.Confusion matrix of YOLO V8 small**

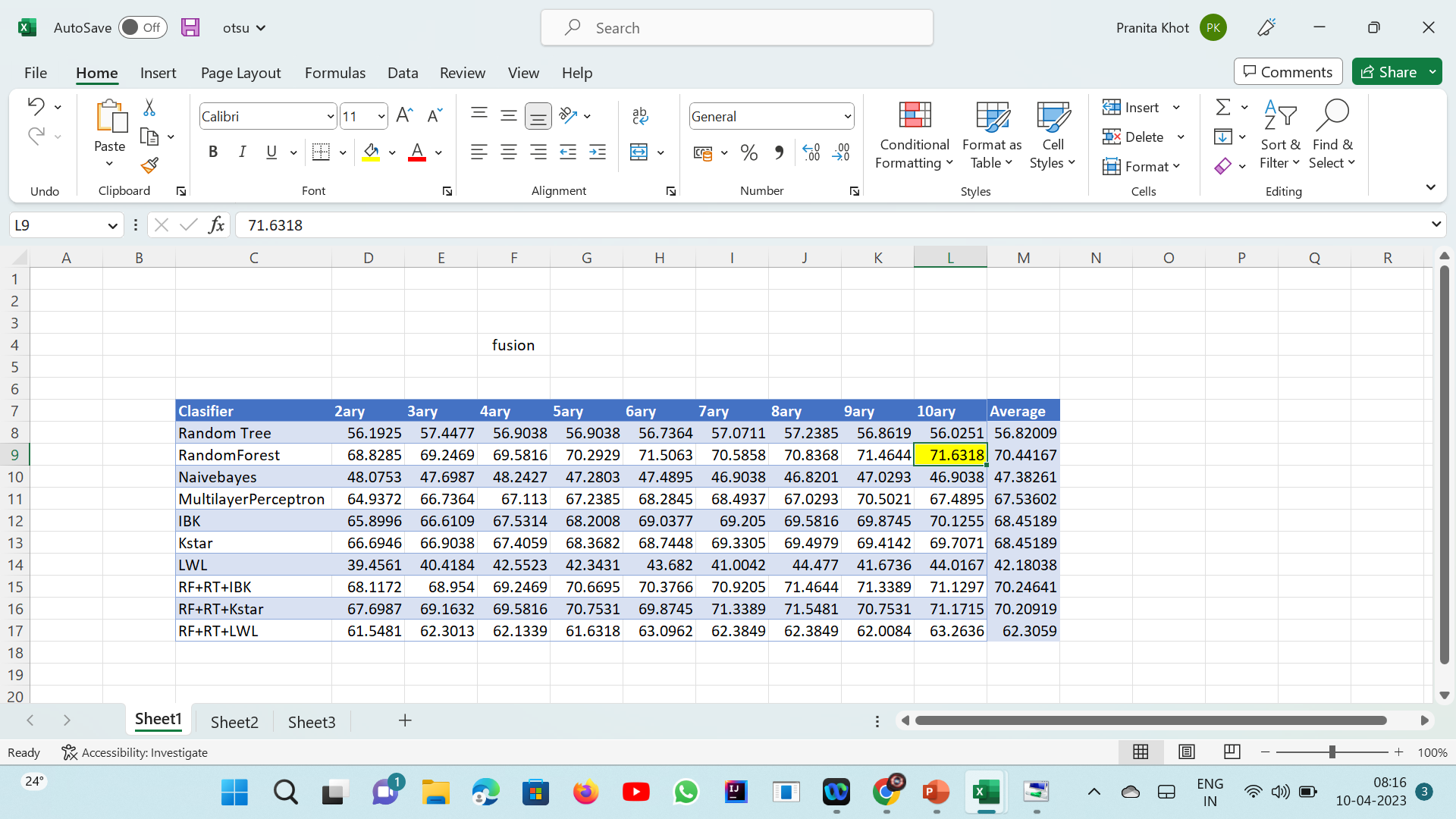
Medium:



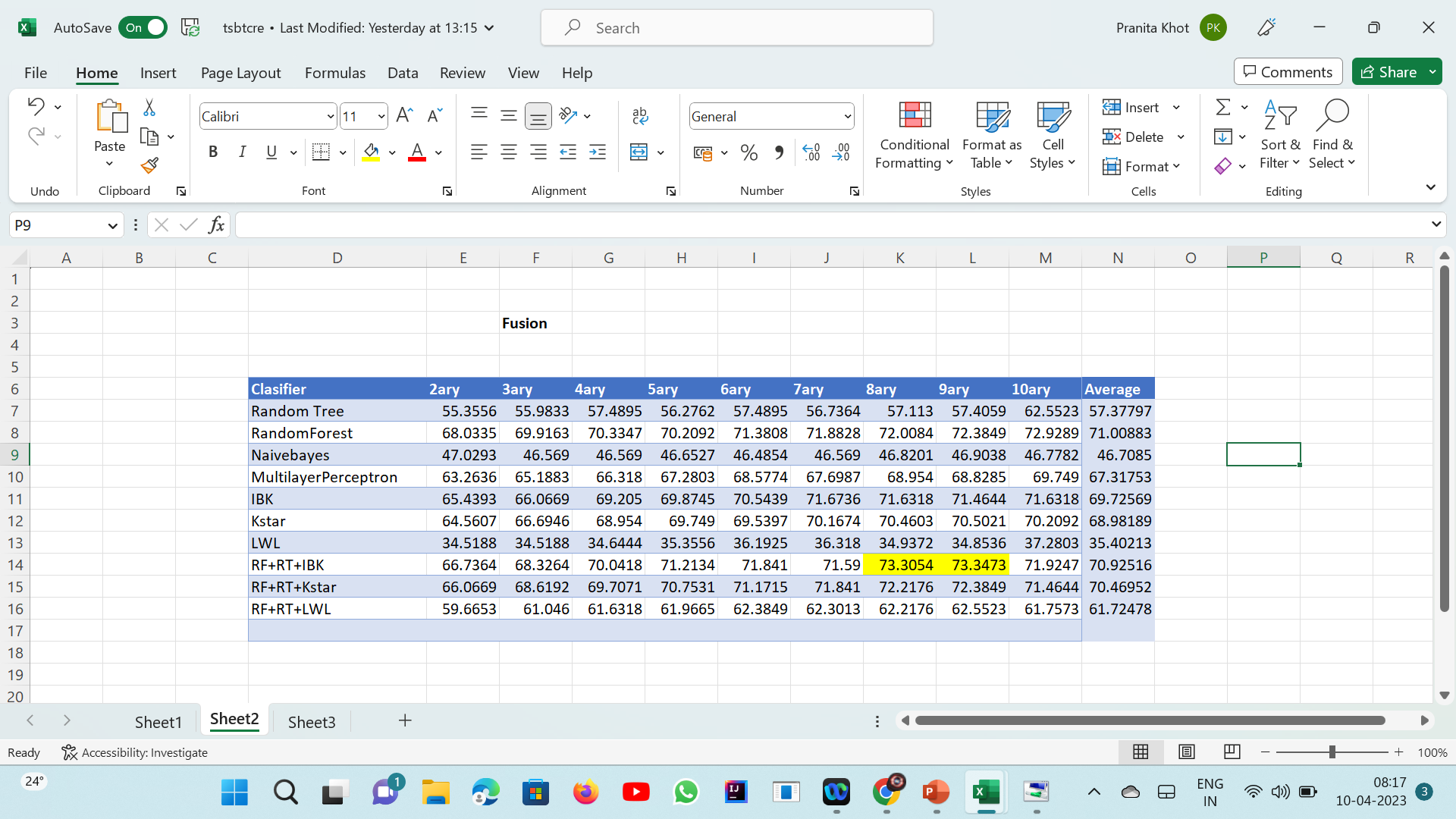
**Fig 9.Confusion matrix of YOLO V8 medium**

Previously, we have done thresholding using Bernsen, Otsu and Niblack followed by feature extraction using TSBTC , where the classification done using classifiers like random forest, random tree , Ibk, Kstar and LWL .We have obtained accuracy for all the classifiers and we got the highest accuracy with the fusion of TSBTC and Niblack with accuracy of 73.47% .

The below table shows the accuracy for fusion between SBTC and Otsu. We can see that 10array is showing the highest accuracy that is 71.63 % with the Random Forest classifier.

****

The below table shows the accuracy for fusion between SBTC and Bernsen. We can see that 9array is showing the highest accuracy that is 73.34 % with the ensemble (Random Forest +RandomTree+ IBK ) classifier.

****

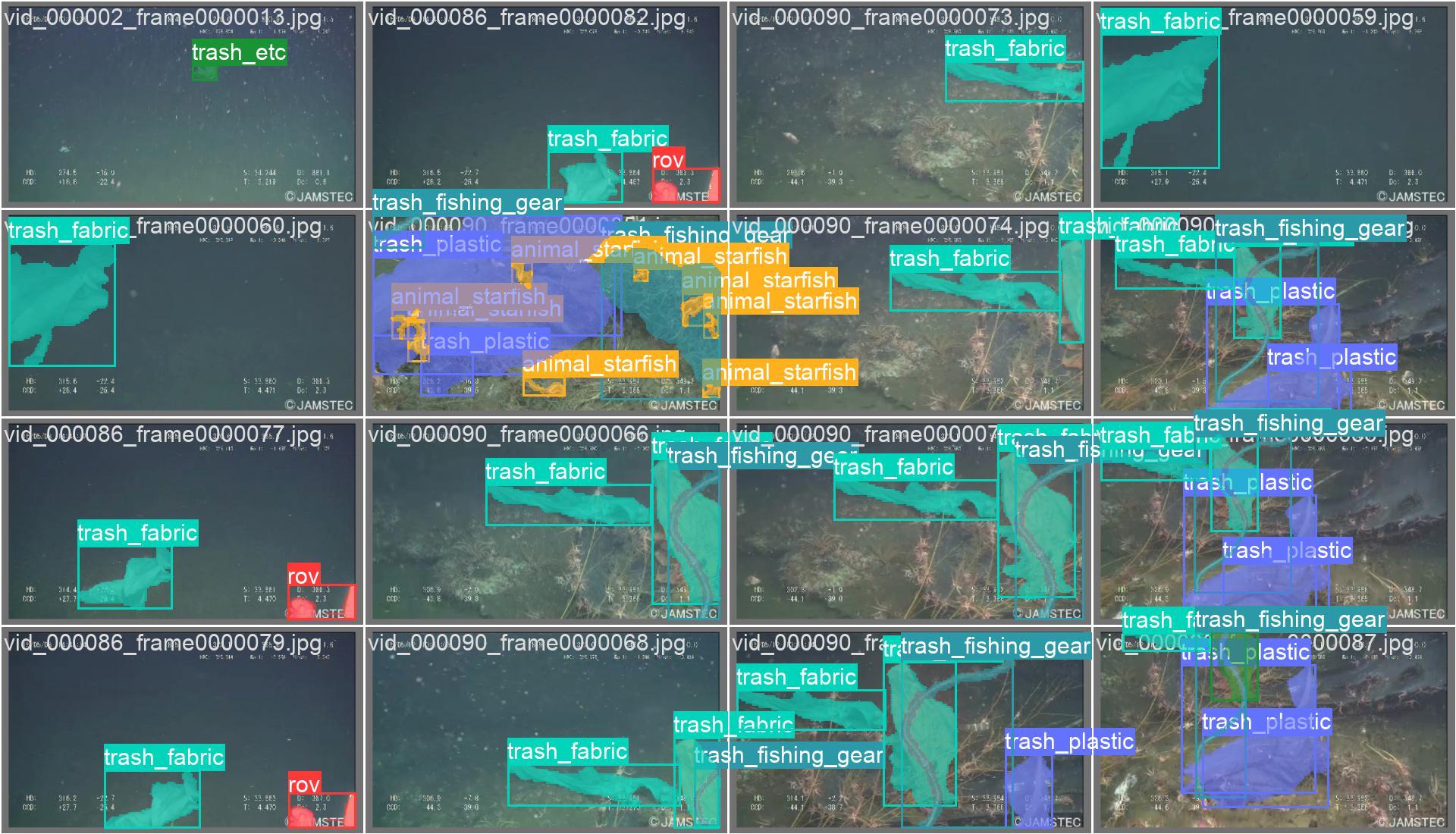
The below table shows the accuracy for fusion between SBTC and Niblack. We can see that 9array is showing the highest accuracy that is 73.47 % with the ensemble (Random Forest +RandomTree+ IBK ) classifier.

**Table

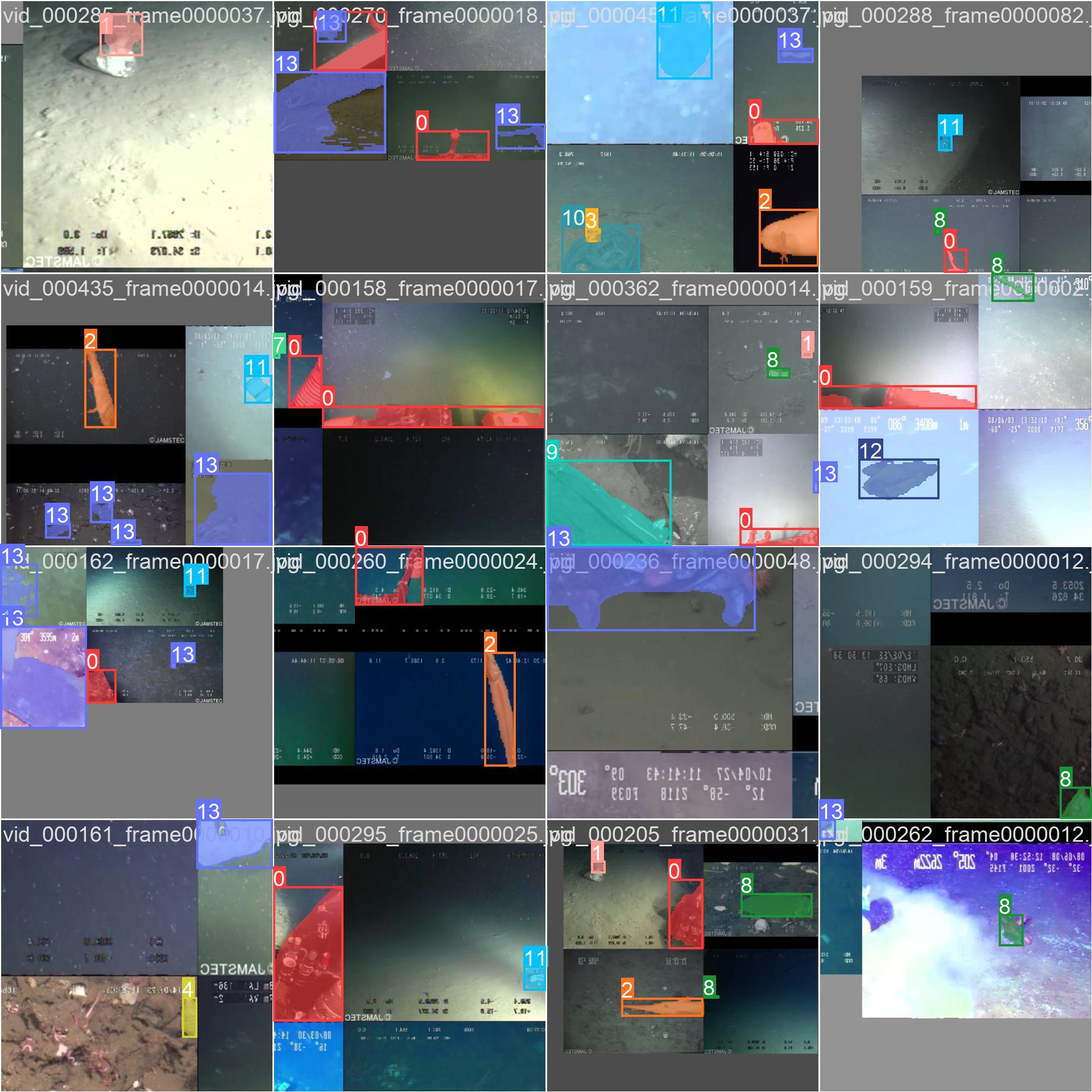
Description automatically generated**

**7.3SCREENSHOT**

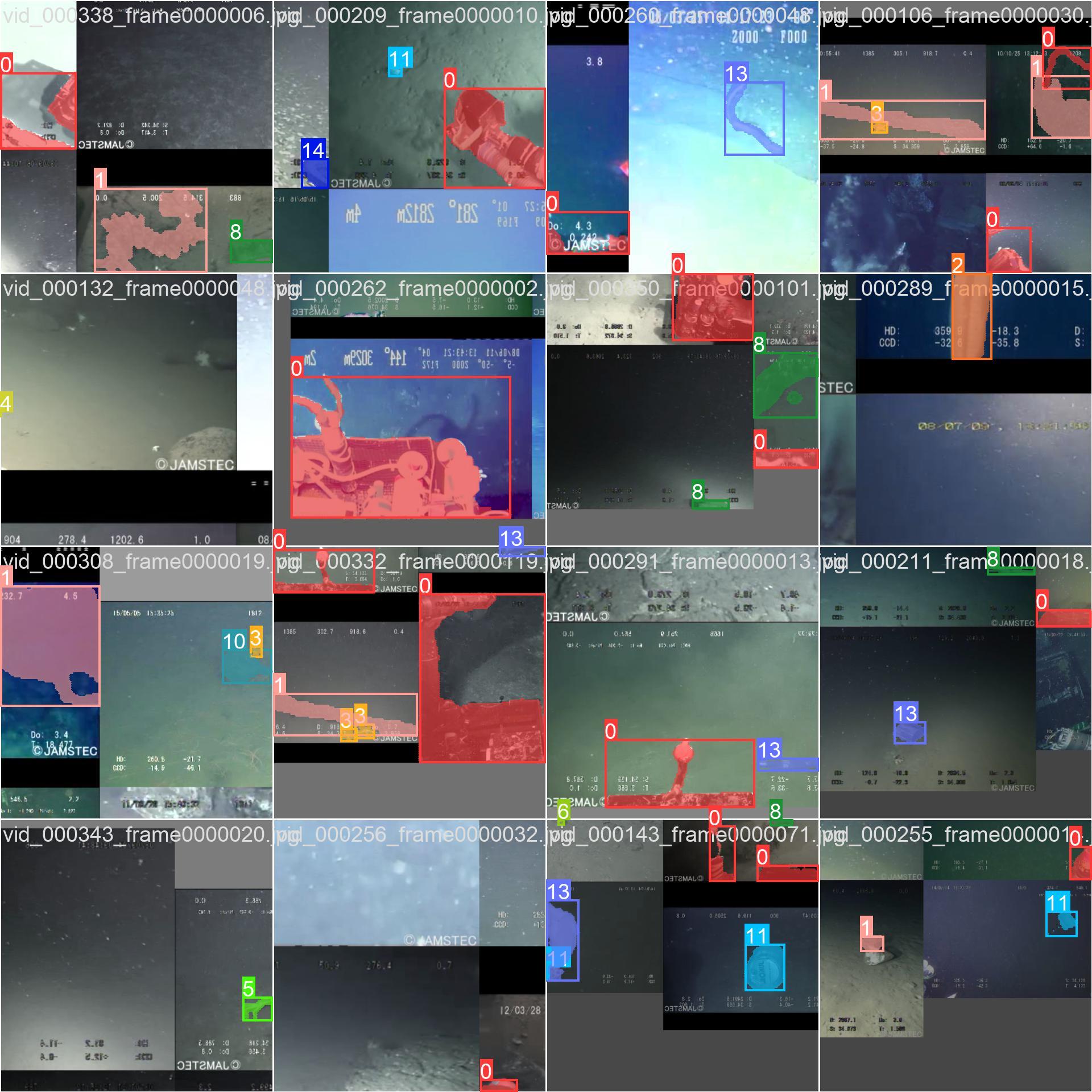
**YOLOv8m(medium)**



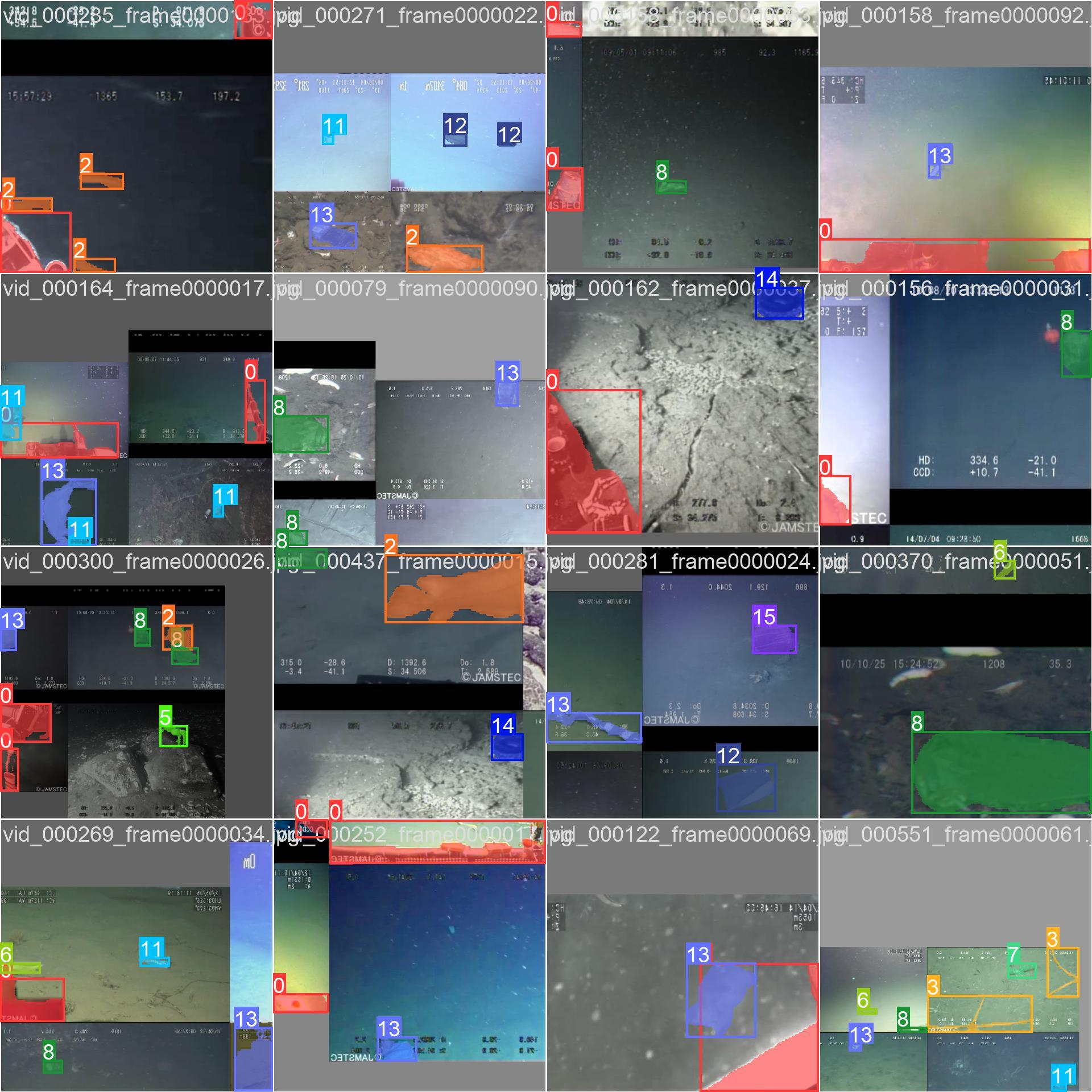
**Fig .10 Output of marine garbage detection using YOLO V8 medium**



**Fig .11 Output of marine garbage detection using YOLO V8 medium**

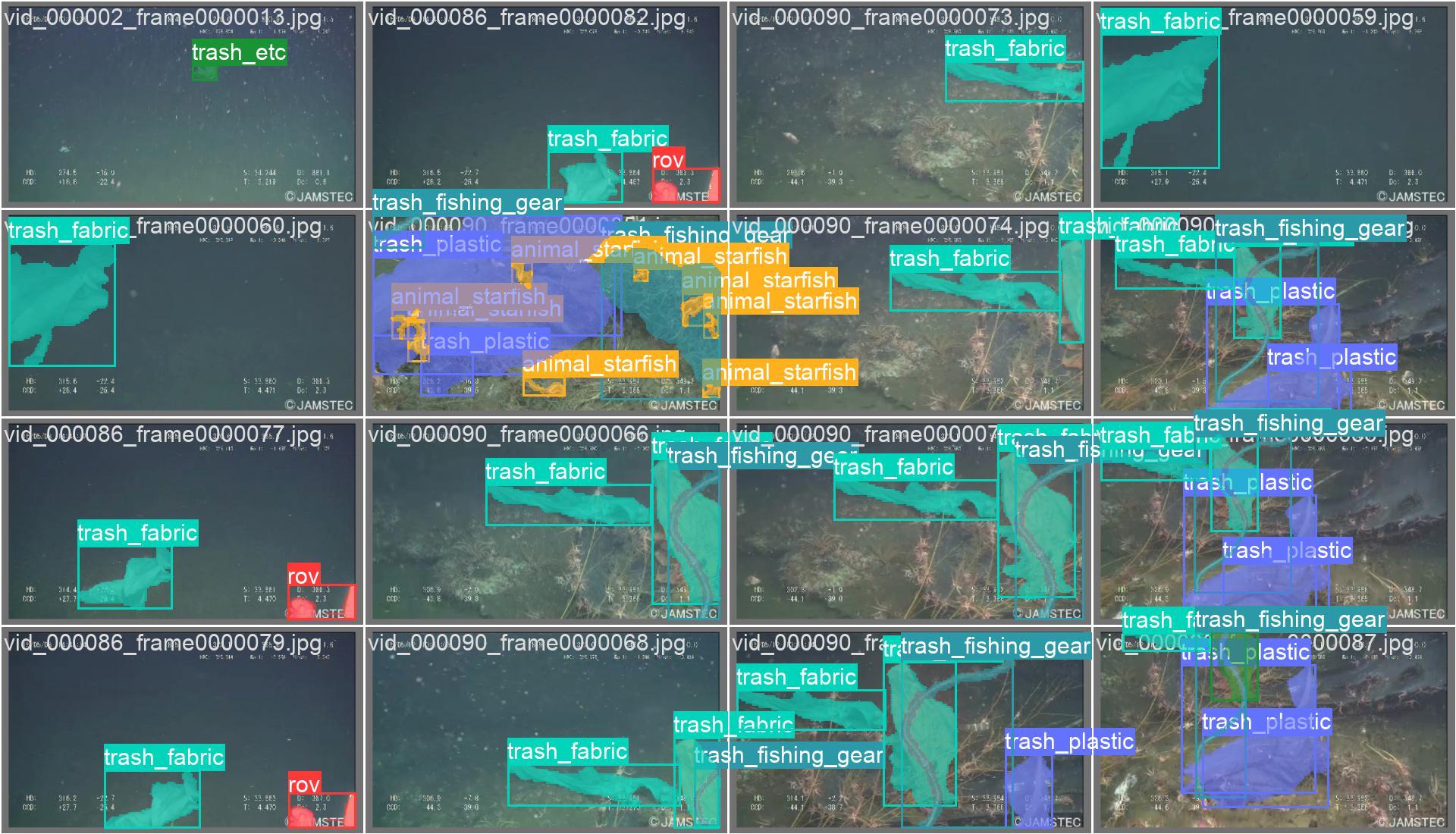


**Fig 12.Output of marine garbage detection using YOLO V8 medium**

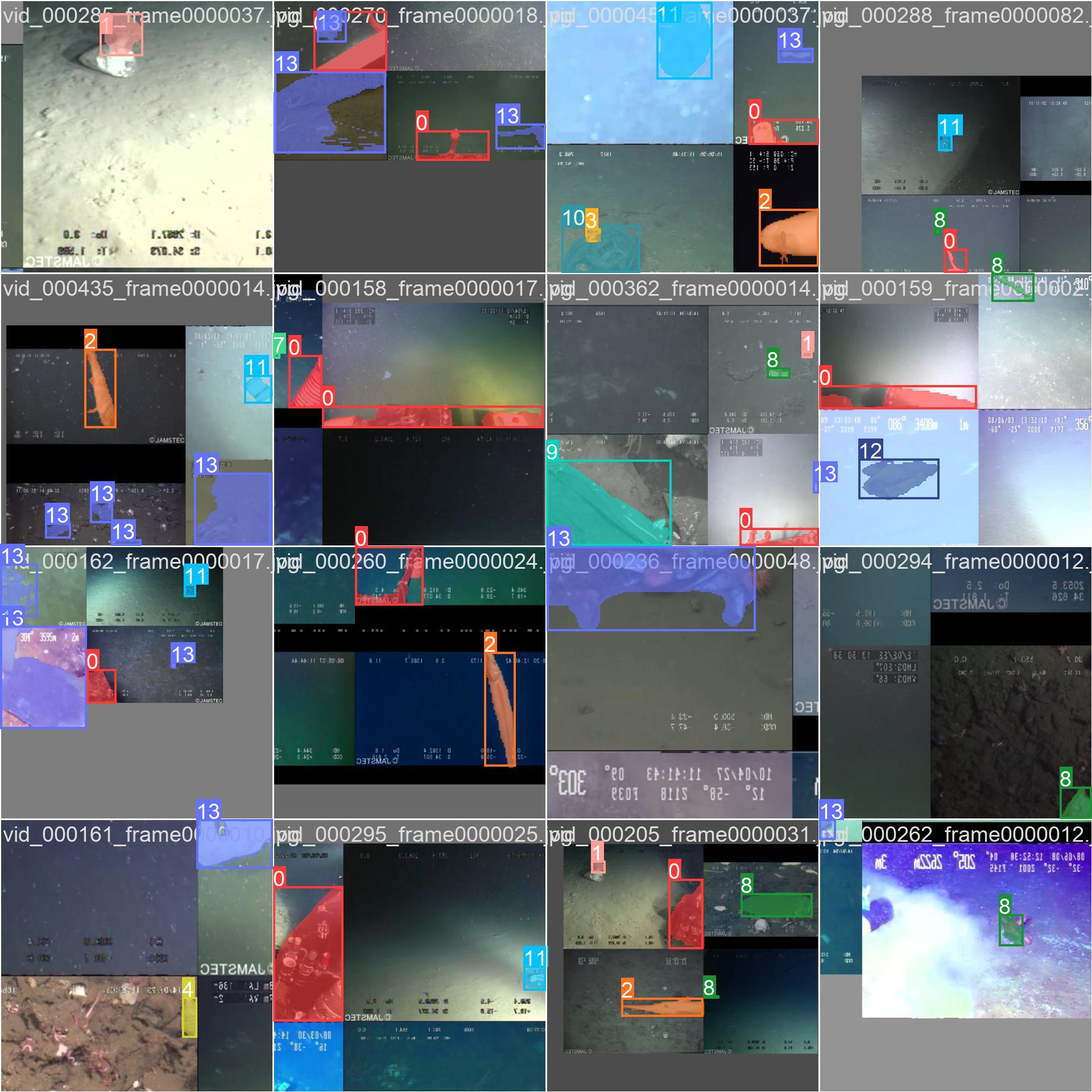


**Fig .13 Output of marine garbage detection using YOLO V8 medium**

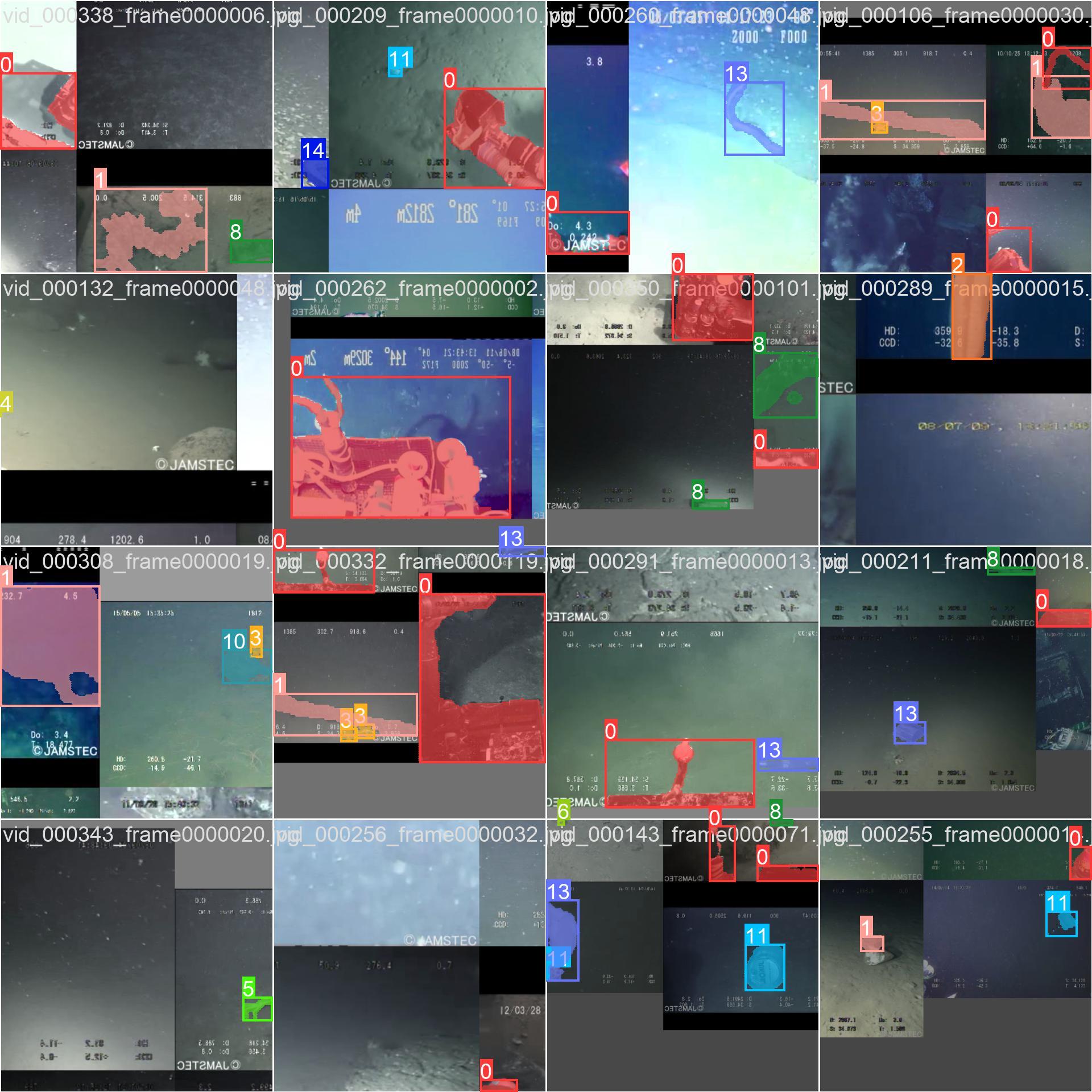
**YOLOv8s(small)**



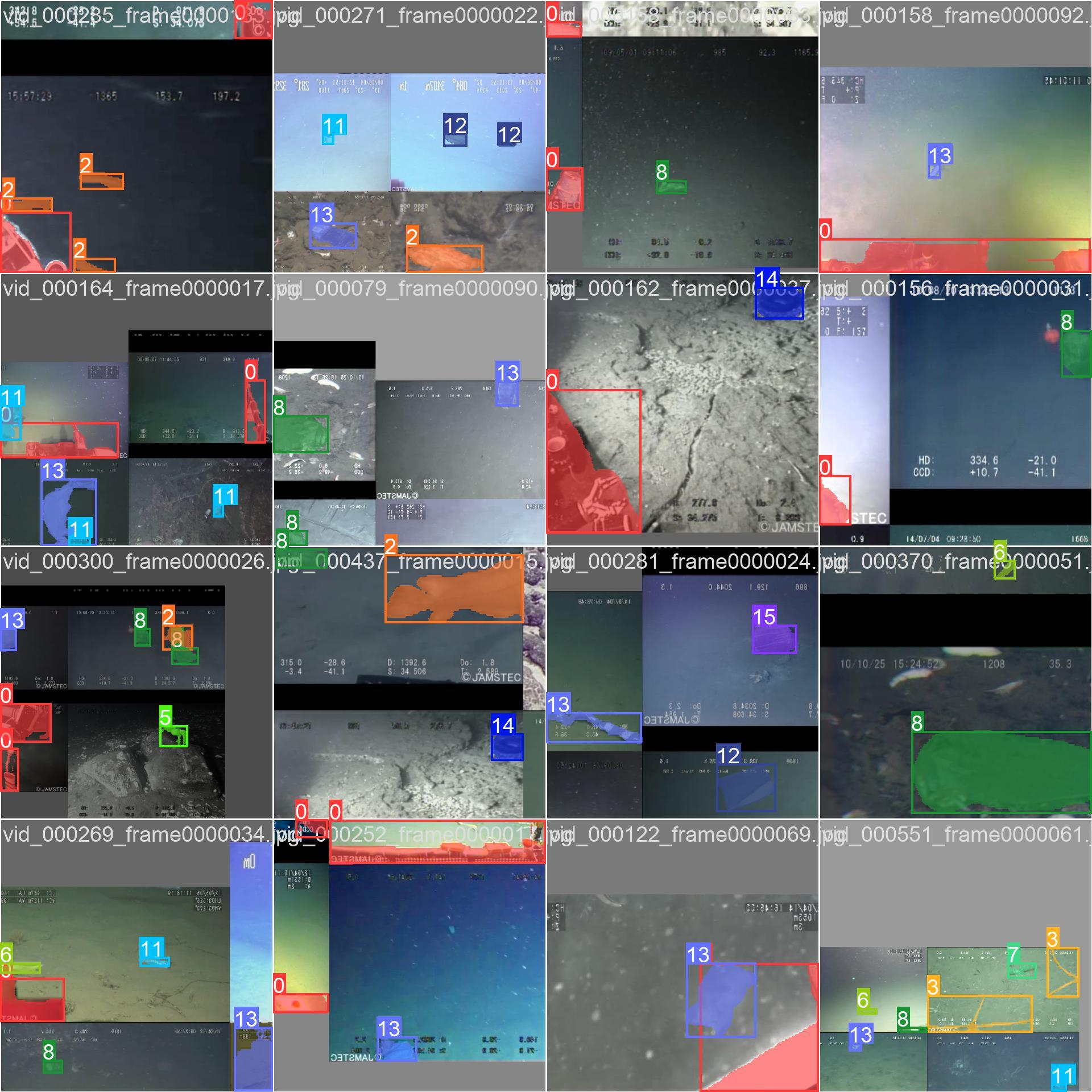
**Fig .14 Output of marine garbage detection using YOLO V8 small**



**Fig .15 Output of marine garbage detection using YOLO V8 small**

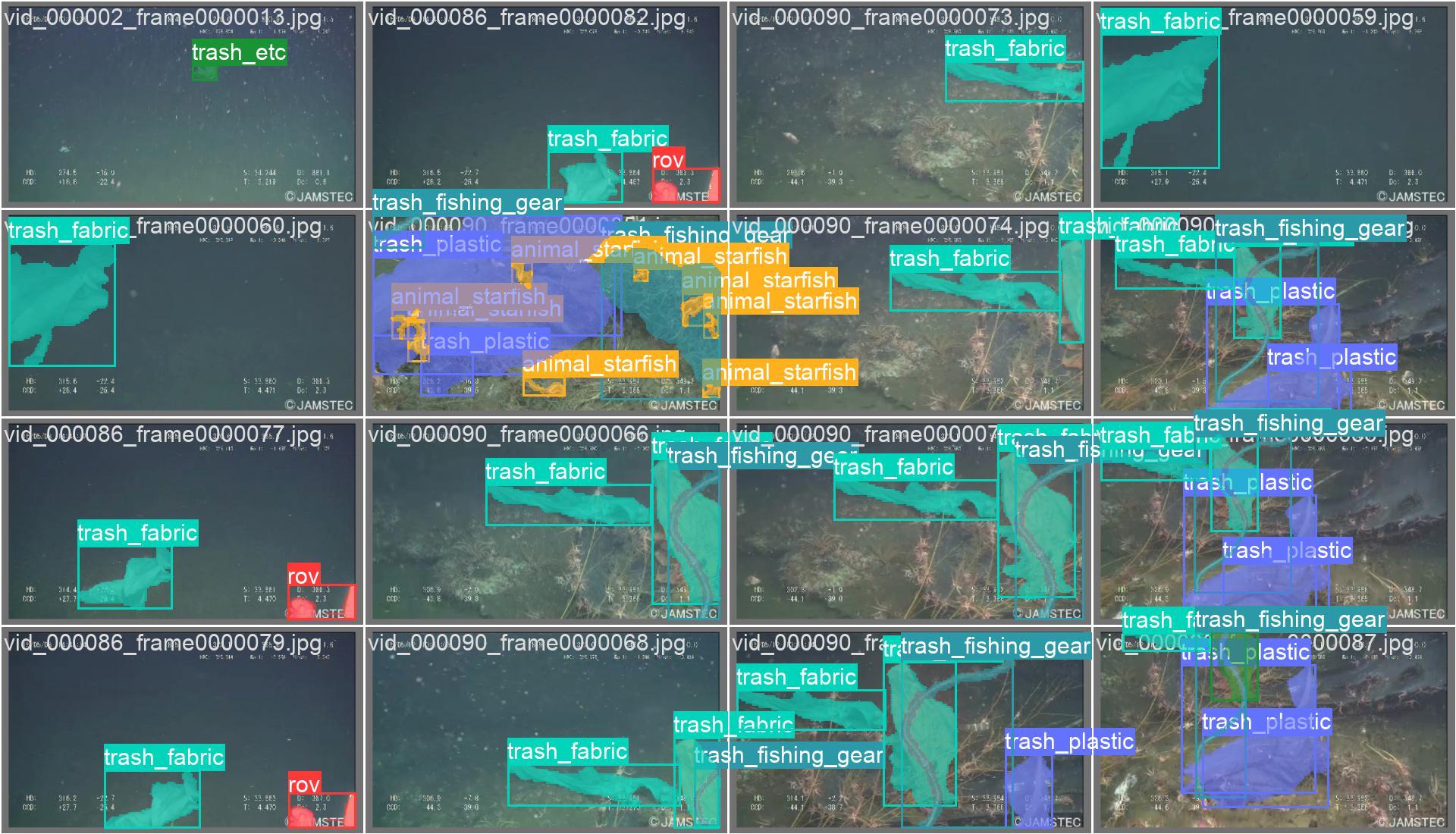


**Fig .16 Output of marine garbage detection using YOLO V8 small**

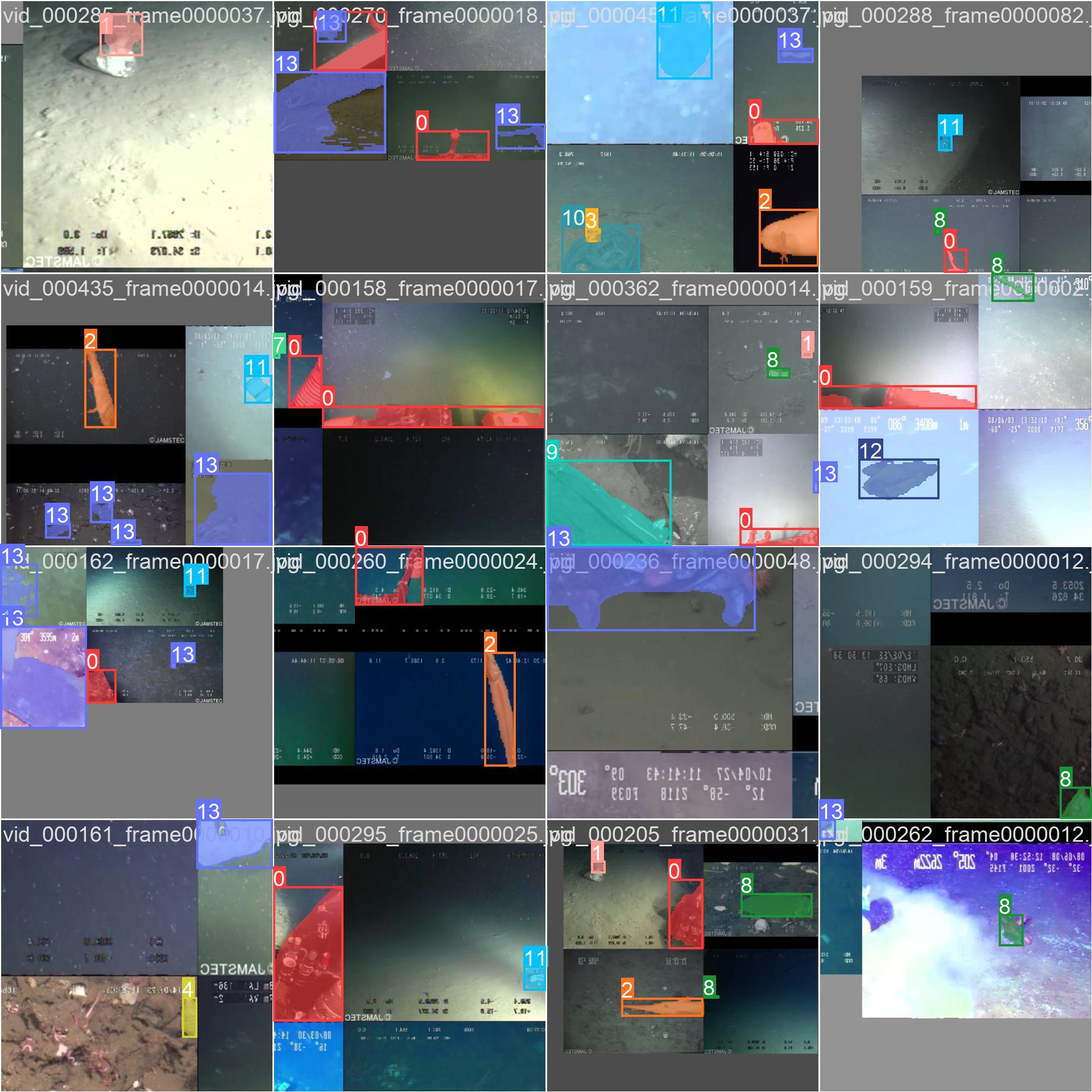


**Fig .17 Output of marine garbage detection using YOLO V8 small**

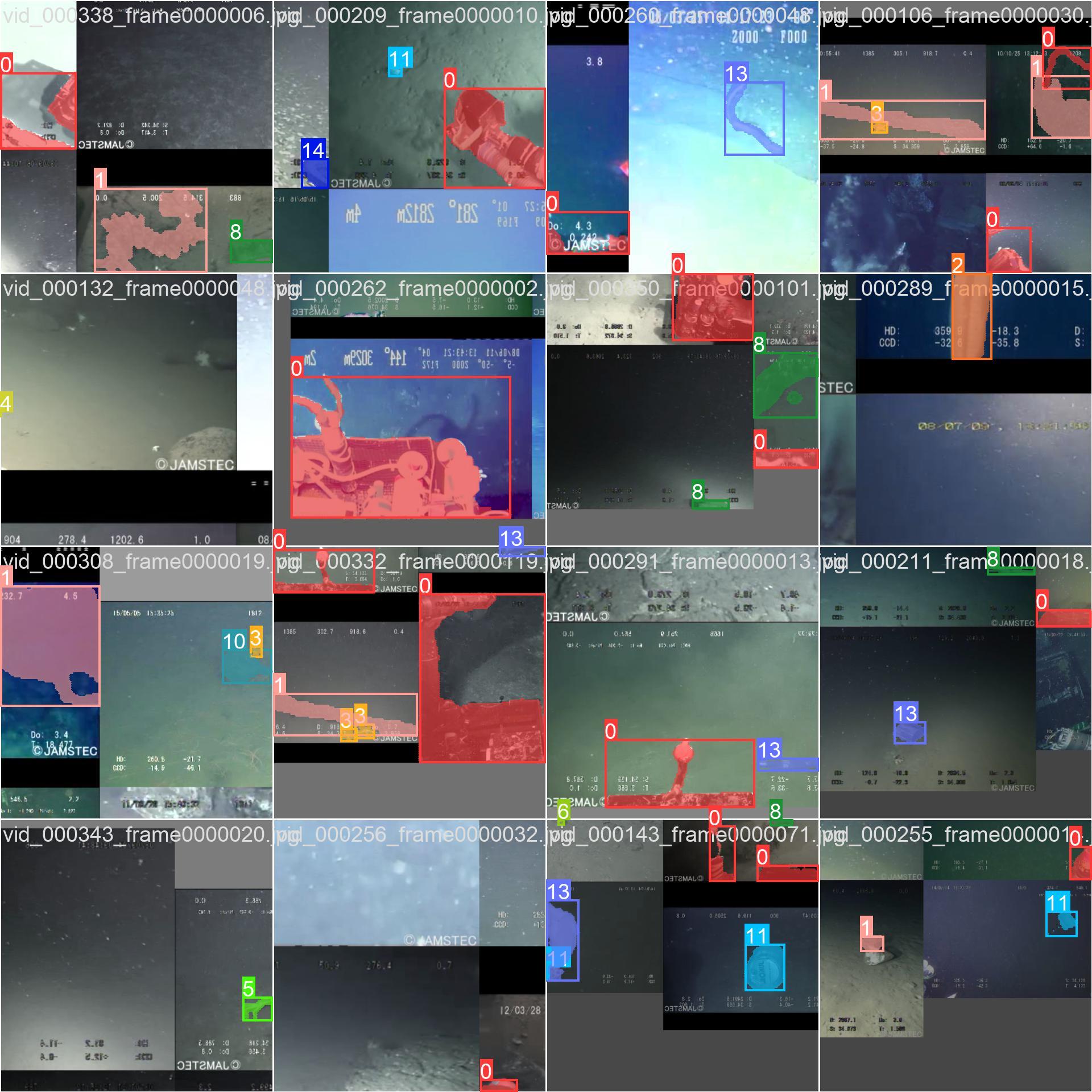
**YOLOv8n(nano)**



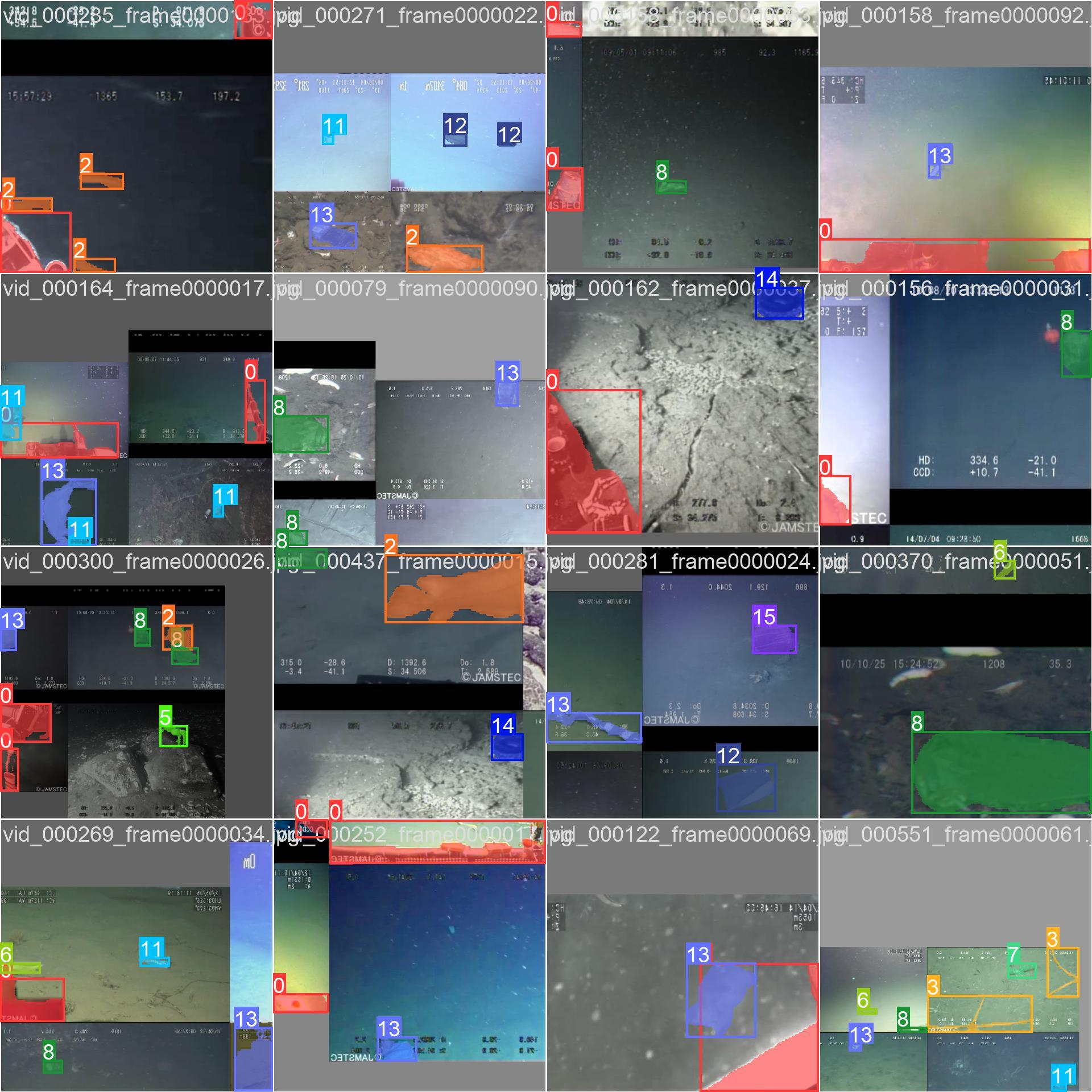
**Fig .18 Output of marine garbage detection using YOLO V8 nano**



**Fig .19 Output of marine garbage detection using YOLO V8 nano**



**Fig .20 Output of marine garbage detection using YOLO V8 nano**



**Fig .21 Output of marine garbage detection using YOLO V8 nano**

**CHAPTER 08 CONCLUSION**

**8.CONCLUSIONS**

**8.1CONCLUSION**

In conclusion, employing YOLOv8 for overall garbage detection yields promising results, showcasing its efficacy in identifying and monitoring marine debris. The performance metrics obtained across different model sizes YOLO Nano, Small, and Medium underscore the significance of model complexity in enhancing detection accuracy. With the YOLOv8 Medium model demonstrating the highest box mAP and segmentation mask mAP, it emerges as the optimal choice for comprehensive garbage detection tasks. These findings highlight the potential of YOLOv8 as a reliable tool for monitoring marine environments and combating pollution effectively. However, further research and deployment considerations are essential to harness its full potential in real-world applications, ensuring scalability, resource efficiency, and integration with decision-making systems for sustainable marine conservation efforts.

**8.2FUTURE WORK:**

Future work for the detection and monitoring of marine garbage using YOLOv8 can focus on several areas to enhance the effectiveness, efficiency, and scope of the application:

* Improved Detection Accuracy: Further refine the YOLOv8 model to achieve higher detection accuracy and reduce false positives/negatives, especially in challenging marine environments with varying lighting conditions, water turbidity, and clutter.
* Object Classification and Segmentation: Enhance the capabilities of YOLOv8 to not only detect marine garbage but also classify and segment different types of debris (e.g., plastics, metals, organic waste), enabling more detailed analysis and targeted mitigation strategies.
* Multi-Sensor Integration: Integrate YOLOv8 with other sensor technologies such as LiDAR, hyperspectral imaging, and acoustic sensors to improve detection capabilities, especially for submerged or partially submerged marine garbage.
* Real-Time Monitoring and Tracking: Develop real-time monitoring and tracking systems based on YOLOv8 to provide timely information on the movement and distribution of marine garbage, facilitating rapid response and intervention strategies.
* Automated Cleanup Systems: Explore the use of YOLOv8 in conjunction with robotic or autonomous systems for automated cleanup of marine garbage, enabling efficient and scalable removal operations in challenging environments.
* Data Fusion and Analysis: Integrate YOLOv8 outputs with other environmental data sources (e.g., ocean currents, weather patterns, marine biodiversity) for comprehensive analysis of marine garbage distribution and its ecological impacts.
* Crowdsourced Data Collection: Develop mobile applications or citizen science initiatives that leverage YOLOv8 for crowdsourced data collection on marine garbage, engaging the public in monitoring efforts and expanding the geographic coverage of observations.
* Long-Term Monitoring and Trend Analysis: Establish long-term monitoring programs using YOLOv8 to track changes in marine garbage distribution over time, enabling trend analysis and assessment of the effectiveness of pollution mitigation measures.
* Collaborative Research and Data Sharing: Foster collaboration between research institutions, government agencies, NGOs, and industry stakeholders to share data, tools, and expertise for advancing the field of marine garbage detection and monitoring using YOLOv8.

**8.3APPLICATIONS:**

The detection of marine garbage using YOLOv8 can have various applications across different sectors and industries. Some of the key applications include:

* Environmental Monitoring and Conservation: YOLOv8 can be used to detect and monitor marine garbage in oceans, seas, and coastal areas, helping researchers and conservationists understand the extent of pollution and its impact on marine ecosystems.
* Marine Cleanup Operations: YOLOv8 can assist in identifying and locating marine garbage for cleanup operations, enabling efficient deployment of resources and prioritization of areas with high levels of pollution.
* Maritime Safety and Navigation: Detection of floating debris such as abandoned vessels, containers, or other hazards using YOLOv8 can enhance maritime safety by alerting ships and vessels to potential obstacles in their path.
* Fisheries and Aquaculture Management: Monitoring marine garbage can help in assessing its impact on fish habitats and aquaculture operations, informing sustainable management practices and mitigating risks to fish stocks.
* Tourism and Recreation: By identifying areas with high levels of marine garbage, YOLOv8 can contribute to the management of tourist destinations and recreational areas, ensuring a safer and cleaner environment for visitors.
* Policy Making and Regulation: Data collected through YOLOv8-based marine garbage detection can inform policy decisions and regulations aimed at reducing marine pollution and promoting sustainable ocean management practices.
* Research and Education: YOLOv8 can support research efforts by providing data on the distribution and composition of marine garbage, facilitating scientific studies on its sources, transport pathways, and ecological impacts. It can also be used as an educational tool to raise awareness about marine pollution and its consequences.

**9.REFERENCES**

1. K. Gu, G. Zhai, X. Yang, W. Zhang and M. Liu, "Subjective and objective quality assessment for images with contrast change," 2013 IEEE International Conference on Image Processing, 2013, pp. 383-387, doi: 10.1109/ICIP.2013.6738079.
2. K. Gu, M. Liu, G. Zhai, X. Yang and W. Zhang, "Quality Assessment Considering Viewing Distance and Image Resolution," in IEEE Transactions on Broadcasting, vol. 61, no. 3, pp. 520-531, Sept. 2015, doi: 10.1109/TBC.2015.2459851.
3. J. Galbally, S. Marcel and J. Fierrez, "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition," in IEEE Transactions on Image Processing, vol. 23, no. 2, pp. 710-724, Feb. 2014, doi: 10.1109/TIP.2013.2292332.
4. H. R. Sheikh, A. C. Bovik and G. de Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics," in IEEE Transactions on Image Processing, vol. 14, no. 12, pp. 2117-2128, Dec. 2005, doi: 10.1109/TIP.2005.859389.
5. S. Bosse, D. Maniry, K. -R. Müller, T. Wiegand and W. Samek, "Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment," in IEEE Transactions on Image Processing, vol. 27, no. 1, pp. 206-219, Jan. 2018, doi: 10.1109/TIP.2017.2760518.
6. Shen, Ao, Yijie Zhu, Plamen Angelov, and Richard Jiang. "Marine Debris Detection in Satellite Surveillance using Attention Mechanisms." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (2024).
7. CS, Sai Ganesh. "Intelligent Debris Mass Estimation Model for Autonomous Underwater Vehicle." *arXiv preprint arXiv:2309.10617* (2023).
8. Zhang, Minghua, Zhihua Wang, Wei Song, Danfeng Zhao, and Huijuan Zhao. "Efficient Small-Object Detection in Underwater Images Using the Enhanced YOLOv8 Network." *Applied Sciences* 14, no. 3 (2024): 1095.
9. Jiang, Zhanjun, Baijing Wu, Long Ma, Huawei Zhang, and Jing Lian. "APM-YOLOv7 for Small-Target Water-Floating Garbage Detection Based on Multi-Scale Feature Adaptive Weighted Fusion." *Sensors* 24, no. 1 (2023): 50.

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Illustration** |
| YOLO | YOU LOOK ONLY ONCE |
| SBTC | Sorted Block Truncation Coding |
| AVC | Automatic vehicle classification |
|  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure** | **Illustration** | **Page No.** |
| 1 | System Architecture | 19-20 |
| 2 | Dataset image | 21-23 |
| 3 | Result | 29-31 |
| 4 | Screenshot | 34-45 |
|  |  |  |
|  |  |  |