



MAELSTROM

MArInE Litter SusTainable RemOval and Management

D3.3

MAELSTROM

Preliminary report on the Cable
robot autonomous control using
Machine learning for litter
identification

06/03/2024



General Information

H2020 call	CE-FNR-09-2020 Pilot action for the removal of marine plastics and litter
Grant Agreement #	101000832
Project Title	Smart technology for Marine Litter Sustainable Removal and Management
Project Acronym	MAELSTROM
Project Duration	January 2021 - December 2024
Project Budget	6 809 461.25 Euros
Related work package	WP3
Related task(s)	T3.6
Lead Organisation	CNRS-LIRMM
Contributing Organizations	
Dissemination Level	Public

Authoring & Approval

Authors	Beneficiary
Marc Gouttefarde	CNRS - LIRMM
Cyril Barrelet	CNRS - LIRMM

Reviewers	Beneficiary
Antonio Petrizzo	CNR-ISMAR

History of Changes

Version	Date	Authors	Description of changes
V0.1	28/11/2023	Marc Gouttefarde	First draft
V0.2	05/12/2023	C. Barrelet	Second draft
V0.3	08/12/2023	C. Barrelet and M. Gouttefarde	Third draft
V0.4	11/12/2023	M. Gouttefarde and C. Barrelet	Fourth draft
V0.5	26/12/2023	M. Gouttefarde and C. Barrelet	Final version, after review by Antonio Petrizzo
V0.6	06/03/2024	M. Gouttefarde	Revision following external reviewer comment (Section 5)

Executive Summary

The MAELSTROM project aims to test and evaluate innovative technologies for the removal of marine litter in different coastal environments, also assessing their impact on the ecosystems in chosen demo sites and evaluating the economic and societal benefits of the MAELSTROM solutions within local economies. Treatments of the plastic litter for their recovery within a circular economy concept are also foreseen.

Deliverable D3.3 deals with the autonomous control of the cable robot of the Robotic Seabed Cleaning Platform. In particular, it reports the on-going work on machine learning for marine litter identification.

Table of content

Executive Summary	4
1 Introduction to the MAELSTROM project.....	7
2 MAELSTROM Consortium.....	9
3 Aims of the Deliverable.....	10
4 Details on the shared autonomy.....	10
4.1 Summary of the litter removal process	11
4.2 Visual servoing	12
4.3 Depth map and high turbidity	12
4.4 Tide estimation and depth compensation	15
4.5 DVL workaround and bathymetry implementation	16
5 Underwater Litter Datasets.....	17
5.1 UNO: Underwater Non-natural Object dataset	19
5.1.1 Acquisition condition	19
5.1.2 Annotation Process	21
5.1.3 Variability Considerations	22
5.2 MORGANE: Marine Observation and Recycling for Garbage Assessment in Nearshore Environments dataset	22
5.2.1 Acquisition condition	23
5.2.2 Annotation Process	25
5.2.3 Variability Considerations	27
5.3 VENICE dataset	27
5.3.1 Acquisition condition	28
5.3.2 Annotation Process	30
5.3.3 Variability Considerations	30
5.4 Comparison between datasets	31
6 Analysis of the datasets.....	34
6.1 Mismatch study	35
6.1.1 Mismatch on raw data	35
6.1.2 Mismatch without the robot class	37
6.1.3 Mismatch on shared classes	38
6.1.4 Mismatch in included classes	39
6.2 Classification task	42
7 Conclusion.....	45

8	References	47
---	------------------	----

List of Figures

Figure 1 - GUI of HMI of the underwater perception system (smart camera, DVL, depth sensors and various data from the cable robot control system).....	11
Figure 2 – First test/implementation of the DVL scan in Port Marghera (Venice).....	13
Figure 3 – DVL mapping at two different resolutions (upper image) and the corresponding bathymetry (lower image).....	15
Figure 4 – The full GUI of the underwater perception system used in Venice during the first cleaning campaign in 2022. The red arrows show objects on the bathymetry image that could be tires.	17
Figure 5 – Random samples of images from the UNO dataset	20
Figure 6 – Class histogram on the UNO dataset	21
Figure 7 – Random samples of images from the MORGANE dataset.....	23
Figure 8 – Class histogram on the MORGANE dataset.....	25
Figure 9 – Example of annotations of a rope. On the left, only one bounding box is used to capture the ropes. On the right, each main part of the ropes has its bounding box. Only the ropes are annotated. Note that the definition of “main part” of a rope rely on the annotator judgement.	26
Figure 10 – Random samples of images from the VENICE dataset	28
Figure 11 – Class histogram on the VENICE dataset	29
Figure 12 – Superclass histogram on UNO, MORGANE, and VENICE datasets.....	32
Figure 13 – Boxplot of the mR in different situations	41

List of Tables

Table 1 – Resume table of the datasets characteristics	33
Table 2 – $mAP@.5$ of YOLOv8 for different train/test datasets on raw data	36
Table 3 – $mAP@.5$ of YOLOv8 for different test/train datasets, data without the robot superclass	37
Table 4 – $mAP@.5$ of YOLOv8 for different test/train datasets, shared classes	39
Table 5 – $mAP@.5$ of YOLOv8 for different test/train datasets, included classes	40
Table 6 – $mAP@.5$ of YOLOv8 for different test/train datasets.....	42
Table 7 – $mAP@.5$ of YOLOv8 for different superclasses and test sets; Training on UNO.....	43
Table 8 – $mAP@.5$ of YOLOv8 for different superclasses and test sets; Training on MORGANE	43

1 Introduction to the MAELSTROM project

MAELSTROM is a project funded under the Topic CE-FNR-09-2020 Pilot action for the removal of marine plastics and litter. MAELSTROM strives to provide answers and diversified solutions to the complex question of the removal and sustainable treatment of marine litter legacy. MAELSTROM contemplates the integration of complementary technologies for marine litter removal in different European coastal ecosystems, compounded with full-fledged circular economy and societal oriented solutions. In particular, the project (i) sets out a reliable multidisciplinary and scientifically sound approach for the assessment of marine debris distribution and impact on marine life in highly valuable ecosystems and protected areas; (ii) designs and manufactures scalable, replicable and automated technologies, co-powered with renewable energy and second generation fuel, to identify, remove and sort marine litter; (iii) evaluates over time the effectiveness of marine litter removal devices along with their impact on local ecosystems; (iv) integrates different technologies to track, sort and recycle all types of collected marine litter into valuable raw materials for future marketisation; (v) assesses the economic and societal impact of the MAELSTROM solutions providing also a comprehensive life-cycle assessment of the technologies and products; (vi) enhances social awareness about the marine litter issue and engages citizens and stakeholders in MAELSTROM activities; (vii) interplays with similar projects to maximize innovation uptake for marine litter removal within and outside the EU.

In particular, MAELSTROM WP3 aims at developing, implementing and integrating the core technologies for the automated system for marine litter removal: The Robotic Seabed Cleaning Platform. The technologies pertain to two blocks: (i) the physical structure (T3.1, T3.2, T3.4) and (ii) the control/automation system (T3.3, T3.5, T3.6). The present deliverable deals with the second block and in particular with the cable robot autonomous control and machine learning for litter identification. In Section 4, the control system of the cable robot of the Robotic Seabed Cleaning Platform is presented. It possesses tools and capabilities

enabling to reduce the complexity of the robot piloting and associated operator fatigue. Then, in Section 5, three datasets are introduced: UNO, MORGANE, and VENICE. They all comprise images of underwater litters captured in natural environments. These datasets were collected from various locations and times, resulting in significant variations in data acquisition conditions and, consequently, in the characteristics of the datasets themselves. Despite their differences, they share the common goal of providing means to detect and classify underwater litter. Finally, Section 6 presents experimental and theoretical analyses of the performances of the detection and classification models on the three datasets presented in Section 5, with the goal of comparing them from a domain adaptation perspective. The underlying idea is to provide different use cases to foster further domain adaptation research on real datasets. We used the YOLOv8 model¹ in the detection and classification experiments. Finally, Section 7 concludes this deliverable.

¹ <https://github.com/ultralytics/ultralytics>

2 MAELSTROM Consortium

	National Research Council	CNR	Project Coordinator Fontina MADRICARDO fontina.madricardo@ismar.cnr.it
	Deltares	Deltares	Frans BUSCHMAN Frans.Buschman@deltares.nl
	University of Malta	UOM	Luciano MULE' STAGNO luciano.mule-stagno@um.edu.mt
	The International Sustainable Development Initiatives	ISDI	Manuel SCARPA m.scarpa@isdigroup.com
	Gees Recycling Srl	GEES	Giorgio BETTETO geesretracking@gmail.com
	Venice Lagoon Plastic Free	VLPF	Davide POLETTI d.poletti@plasticfreevenice.org
	CIMA Research Foundation	CIMA	Isabel GOMES isabel.gomes@cimafoundation.org
	TECNALIA RESEARCH & INNOVATION	TECNALIA	Damien SALLÉ damien.salle@tecnalia.com
	ALPHA Consult	ALPHA UK	Emiliano SPALTRO es@alphacons.eu
	Interdisciplinary Centre of Marine and Environmental Research	CIIMAR	Isabel SOUSA-PINTO ispinto@ciimar.up.pt
	Servizi Tecnici	ST	Nicola FERRARI n.ferrari@stvenezia.com
	The Great Bubble Barrier	TGBB	Philip EHRHORN philip@thegreatbubblebarrier.com
	MAKEEN	MAKEEN	Anders BJORN ABJ@makeenenergy.com
	National Center for Scientific Research	CNRS	Marc GOUTTEFARDE marc.gouttefarde@lirmm.fr

3 Aims of the Deliverable

The aim of Deliverable D3.3 is to report on the autonomous control of the cable robot and, in particular, on the on-going work on using machine learning for litter identification. In fact, Deliverable D3.3 is a preliminary report on the on-going work in WP3, Task T3.6 "Cable robot control system improvement - Machine Learning for autonomous cleaning of the seabed" by the partner CNRS-LIRMM.

The main objectives of Task T3.6 are the following.

In this task, we will design a system able to provide suggestions - in the form of areas/objects highlighted in the interactive image available to the robot operator - regarding which regions or objects on the sea floor should be marked for removal. The system will use Artificial Intelligence machine learning techniques such as Deep learning or Reinforcement Learning. These algorithms will allow the robot to learn from the previous decisions made by the operator during the shared autonomy mode. A dataset for training the machine learning algorithm will be generated based on the human operator who select litters to collect by clicking on the images. The machine learning algorithm will then be continuously improving its ability to identify potential target areas and objects. In all cases, the decision to remove a target will be up to the operator in order to treat appropriately potential archaeological items or even explosives. This task is done in collaboration with the University of Montpellier (UM).

4 Details on the shared autonomy

The overall goal is to endow the control system of the cable robot of the Robotic Seabed Cleaning Platform with tools and capabilities enabling to reduce the complexity of the robot piloting and associated operator fatigue. The content of this section has already been presented in Deliverable D3.2 but it is recalled here for easier reference.

4.1 Summary of the litter removal process

This section summarizes the litter removal process during the cleaning campaign with the Robotic Seabed Cleaning Platform in Venice in 2022. The corresponding demonstration activities have been presented in the following video: <https://youtu.be/16k3-Bp4FCI>

First, a location is selected with the help of the bathymetry provided by the CNR-ISMAR. The objective is to find a zone where big litters can be seen on the bathymetry, and to move the floating barge to the corresponding GPS coordinates.

As soon as the barge arrived to the desired location, a scan is performed with the Doo Doppler Velocity Log (DVL) at sea level. The scan gives an accurate mapping of the zone, offering a reliable and real-time feedback to the operator. Once the scan is done, the robot can move freely within its safe working zone, as every dangerous object (i.e. large objects that could collide with the robot) are detected and projected on the Graphical User Interface (GUI) depth map (Figure 1).

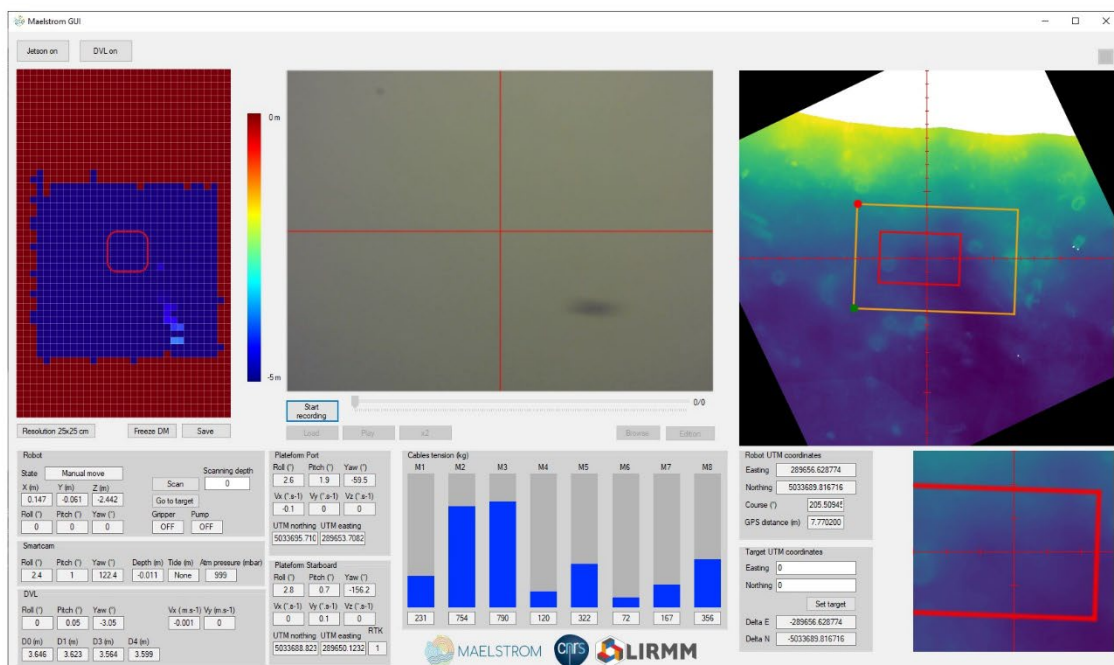


Figure 1 - GUI of HMI of the underwater perception system (smart camera, DVL, depth sensors and various data from the cable robot control system)

A second underwater DVL scan is then performed closer to the seabed, using the prior scan to move close but above the seabed, allowing for better resolution mapping, but also for more visibility on the potentially lying litter. While the robot is scanning the zone for the second time, the operator can click on the image (within the GUI) where he sees macro litter. Each time the operator clicks on the image, the depth (given by the depth map) and the 2D coordinates of the clicked pixel are saved.

Finally, once this last scan is completed, the robot can automatically move a few centimetres above the saved coordinates, giving a complete view of the object and letting the operator grab or suck the macro litter. The grabbed object can then be retrieved on the barge, and the robot can automatically move to the next litter lying on the seabed.

4.2 Visual servoing

First and foremost, the smart-camera is calibrated underwater with a checkerboard. The intrinsic parameters, such as the optical point, the focal length, and the skew, and extrinsic parameters, which transform a 3D world point to the 2D camera coordinates, are then computed.

Then, being given a 2D point on the camera coordinates, and its depth (given by the DVL which is rigidly attached to the smart camera), we can determine the spatial position of an object. This three-dimensional position is then send to the robot, which calculate the trajectory.

Although the visual servoing has been implemented and tested on ground (see video: <https://seafire.lirmm.fr/f/0401ac21e6c74d32ac2b/>), the high turbidity of the water in Venice lagoon made impossible for the operator to identify clearly any litter within the image.

4.3 Depth map and high turbidity

The DVL scan has been tested for the first time in Port Marghera (Venice). Figure 2 shows the first version of the GUI and the resulting depth map (see video: <https://seafire.lirmm.fr/f/2feac9b169b844cfb41f/>).

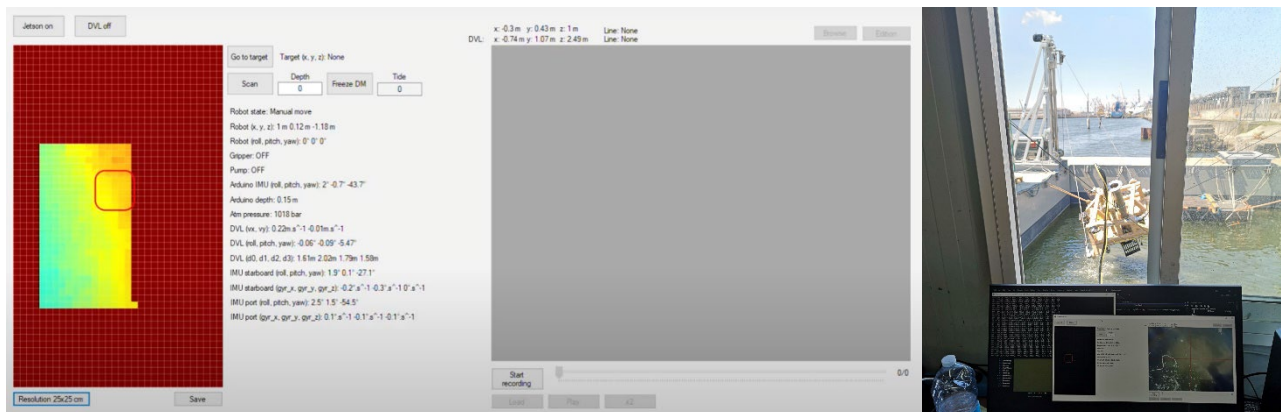


Figure 2 – First test/implementation of the DVL scan in Port Marghera (Venice)

While the operator moves the robot (represented with the red square in the depth map), the GUI software records the four beams signal of the DVL and, knowing their orientations, projects them into the depth map. The bathymetry shown in Figure 2 suggests a light slope, which is confirmed by the barge location. Indeed, the barge was docked on the side of the canal bank, which explains this slope.

The next day of these tests in Venice, the barge moved to the Arsenale to perform further tests before the first cleaning campaign. At this stage, the GUI uses the 2 RTK-GPS antennas (located on both the barge port and starboard sides) to compute the position of the barge with respect to the map origin. The global map origin is defined as the geographic point located at distance chosen to be 50 meters away in the North-West direction, starting from the position of the port GPS antenna, when a new map is requested. The global map axes are in the range [0, 100 m] and are oriented respectively to the East (x-axis) and to the South (y-axis). The update rate of the RTK-GPS is 1 Hz, but this should be increased, if possible, to smoothen the position noise (less than 5 cm during July 2022 experiments in Venice).

Hence, each beam is geo-localized in the global map, allowing the barge to drift while keeping track of the depth map (see video: <https://seafire.lirmm.fr/f/6658db4e6dddf41feae6c/>).

Figure 3 shows the resulting mapping for two different resolutions (5x5 cm and 25x25 cm, upper image). Interestingly, a squared object emerges from the depth map, which suggests that one of the “blocks” lying on the seabed, shown by the red circle on the bathymetry image (lower image of Figure 3) has been flown over by the robot.

The high water turbidity, which leads to very poor visibility, restrains the camera usage since the operator could see barely anything on the image. Although we enhanced the image by applying an adaptive histogram equalization (CLAHE), strong limitations in visibility range were still present.

Therefore, we decided to use a low resolution (i.e. 25x25 to 50x50 cm) for the first scan, and then use a high resolution (5x5 cm) hoping to draw the shape of medium to large objects in the depth map. This idea was tested with the block shown in red in Figure 3.

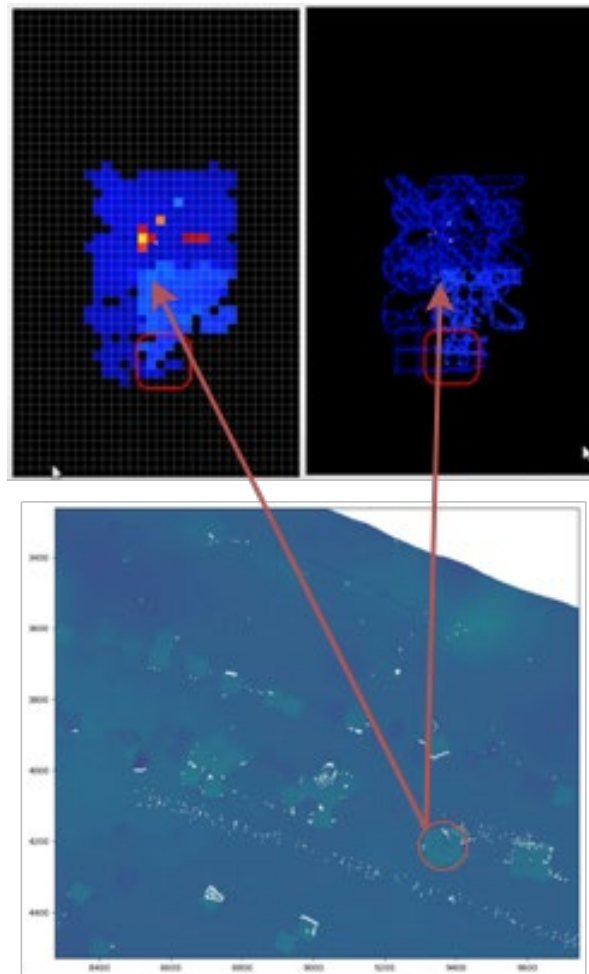


Figure 3 – DVL mapping at two different resolutions (upper image) and the corresponding bathymetry (lower image)

4.4 Tide estimation and depth compensation

As the tide was changing relatively slowly in Venice lagoon during the experiments in July 2022, we decided to estimate the tide with the DVL.

Once the first scanning is completed, the current depth map is saved as a reference for tide estimate. Subsequently, the tide is estimated in real-time, with respect to the initial map, by comparing the current height of the water column, i.e., the sum of the altitude given by the barometer on the floating platform, and the smart-camera depth with the initially measured height (i.e. value stored in the initial depth map). This tide

value can then be used to compute the tide-compensation values of the depth map.

4.5 DVL workaround and bathymetry implementation

Unfortunately, the DVL broke down the next day without letting any possibility of quick reparation. As we did not start the cleaning campaign yet, we had to quickly find a workaround.

First, we chose to test the accuracy of the floating barge GPS position with respect to the bathymetry map: We used the GPS coordinates of one of the blocks visible in the bathymetry (as illustrated in Figure 3) and tried to place the end-effector of the robot at the same location. As the result appeared to be accurate enough (approximatively less than 5 cm error), we decided to implement the bathymetry directly into the GUI as shown in Figure 4. As shown by red arrows in Figure 4, we could see numerous objects, similar to tires, lying on the seabed on the bathymetry image. Because of the very low visibility and the DVL breakage, the operator used the bathymetry maps to pilot the robot (see video: <https://seafire.lirmm.fr/f/41f9c230bc864aa2add5/>).

In the bathymetry, the orange rectangle represents the inner pool of the barge, i.e., the rectangular hole allowing the mobile platform of the cable robot to dive in the water. The red rectangle represents the safe working zone of the robot where no cable collision with the barge can happen, and the scaled cross at the center of the image represents the robot position.

We decided to rotate the image with respect to the barge course but also the robot coordinate system to help the operator, so that the operator joysticks and the image are oriented in the same way.

Moreover, although we could not use the visual servoing because of high water turbidity, we added the possibility for the operator to enter desired GPS coordinates and send the cable robot mobile platform to these coordinates automatically.

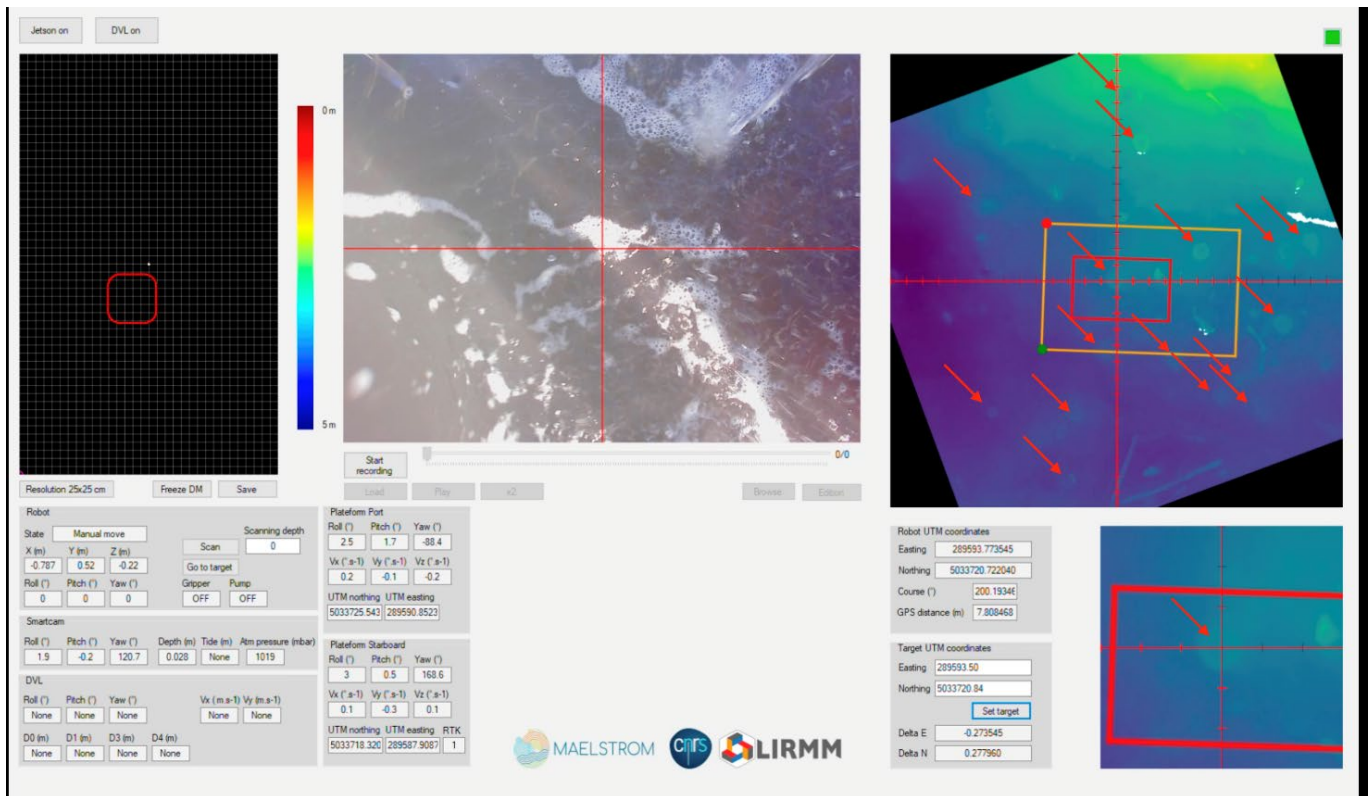


Figure 4 – The full GUI of the underwater perception system used in Venice during the first cleaning campaign in 2022. The red arrows show objects on the bathymetry image that could be tires.

5 Underwater Litter Datasets

As further improvements in helping the operator to efficiently collect underwater litter with the Robotic Seabed Cleaning Platform, a system able to provide suggestions on which regions or objects on the seafloor should be marked for removal was implemented. The suggestions take the form of areas/objects highlighted in the interactive image seen by the operator.

Thanks to the emergence of deep learning, particularly convolutional neural networks, models now enable us to detect objects in images and categorize them almost systematically. However, these models require a large amount of annotated data to learn different tasks, such as detecting, localizing objects in an image, and categorizing them. Despite the availability of a wide variety of databases, especially for autonomous driving or tracking animal species, few of them focus on the subject of underwater waste.

Moreover, a well-known problem in the deep learning community is the so-called "domain mismatch". This problem arises when the neural network is trained on a dataset in the source domain but, once the model is learned, used on another similar dataset in the target domain. The disparity between the two databases leads to a significant drop in performance, greatly affecting the confidence we can attribute to the evaluation of an object detection and classification model. To reduce the gap between source and target databases, domain adaptation methods can be used.

Given the limited quantity of underwater waste detection databases and the complete lack of diverse domains to assess model performance, we have created three different databases to detect waste in underwater images.

The three datasets introduced here are MORGANE, VENICE, and UNO. They all comprise images of underwater waste captured in natural environments. These datasets were collected from various locations and times, resulting in significant variations in data acquisition conditions and, consequently, in the characteristics of the datasets themselves. Despite their differences, they share the common goal of providing useful datasets to train and validate the machine learning tool for waste/litter detection and classification.

Considering this shared objective and the distinctions between them, these datasets could serve as practical cases for domain adaptation.

The two datasets MORGANE and VENICE have been specifically produced for the purpose of MAELSTROM. They will be available as open data later on during the project, and at the latest at the end of task T3.6. Indeed, they are still the subject of on-going work.

The dataset UNO is already available at <https://www.lirmm.fr/uno/> and https://github.com/CBarrelet/balanced_kfold.

5.1 UNO: Underwater Non-natural Object dataset

5.1.1 Acquisition condition

The UNO dataset is a derivation of the TrashCAN dataset (Hong, Fulton, & Sattar, 2020). It was introduced in (Barrelet, Marc, Gérard, Vincent, & Marc, 2022) in a version without classes. Here, its fully annotated version is introduced.

Videos were recorded by the Japan Agency of Marine Earth Science and Technology (JAMSTEC) since 1982 using cameras mounted on Remotely Operated Vehicles (ROVs). The JAMSTEC Deep Sea Debris database, from where the videos originate, contains footage of underwater waste from many locations in the Sea of Japan and Pacific Ocean. Due to the high depth, between 50 m and 9000 m, the seabed is mostly free of litter (objects) on the images and sea condition is constant, with very clear water and constant luminosity provided by the ROV.

During the TrashCAN to UNO transformation, frames were trimmed to prevent metadata from being overlaid on them. After this trimming process, the frames that did not contain anymore waste were deleted. This process resulted in 5,985 frames of resolution from 156x480 to 345x480 pixels, extracted from 279 videos. In average, 21.45 frames were extracted from each video. Some samples can be seen in Figure 5.

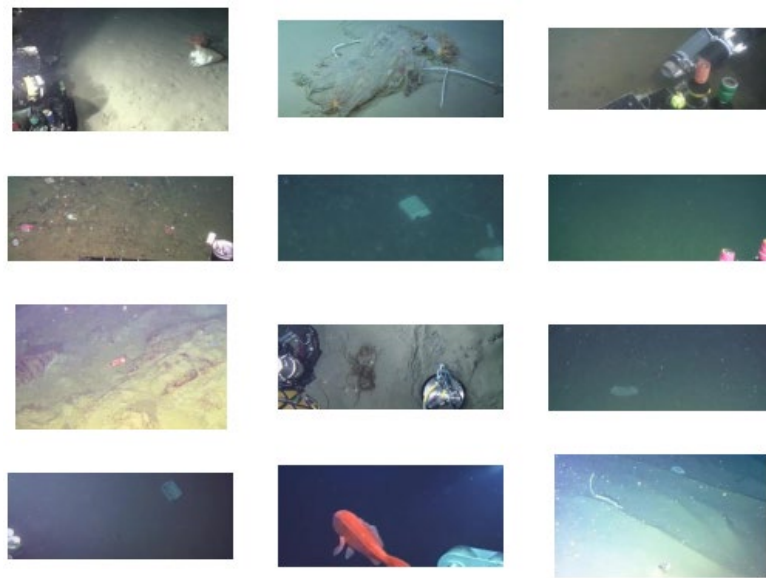


Figure 5 – Random samples of images from the UNO dataset

In order to conduct experiments, this dataset can be divided into training, validation and test splits. However, one needs to be careful since the dataset primarily consists of video frames rather than independent images. Consequently, it is crucial to ensure that video sequences remain intact within a single split to avoid biased performance estimation.

UNO contains 38 different classes, detailed in the Figure 6, along with their distribution over the dataset. One can see that the classes are not balanced, especially the class robot, which describes the visible ROV parts on the frames and is overrepresented since the robot appears in most of the frames.

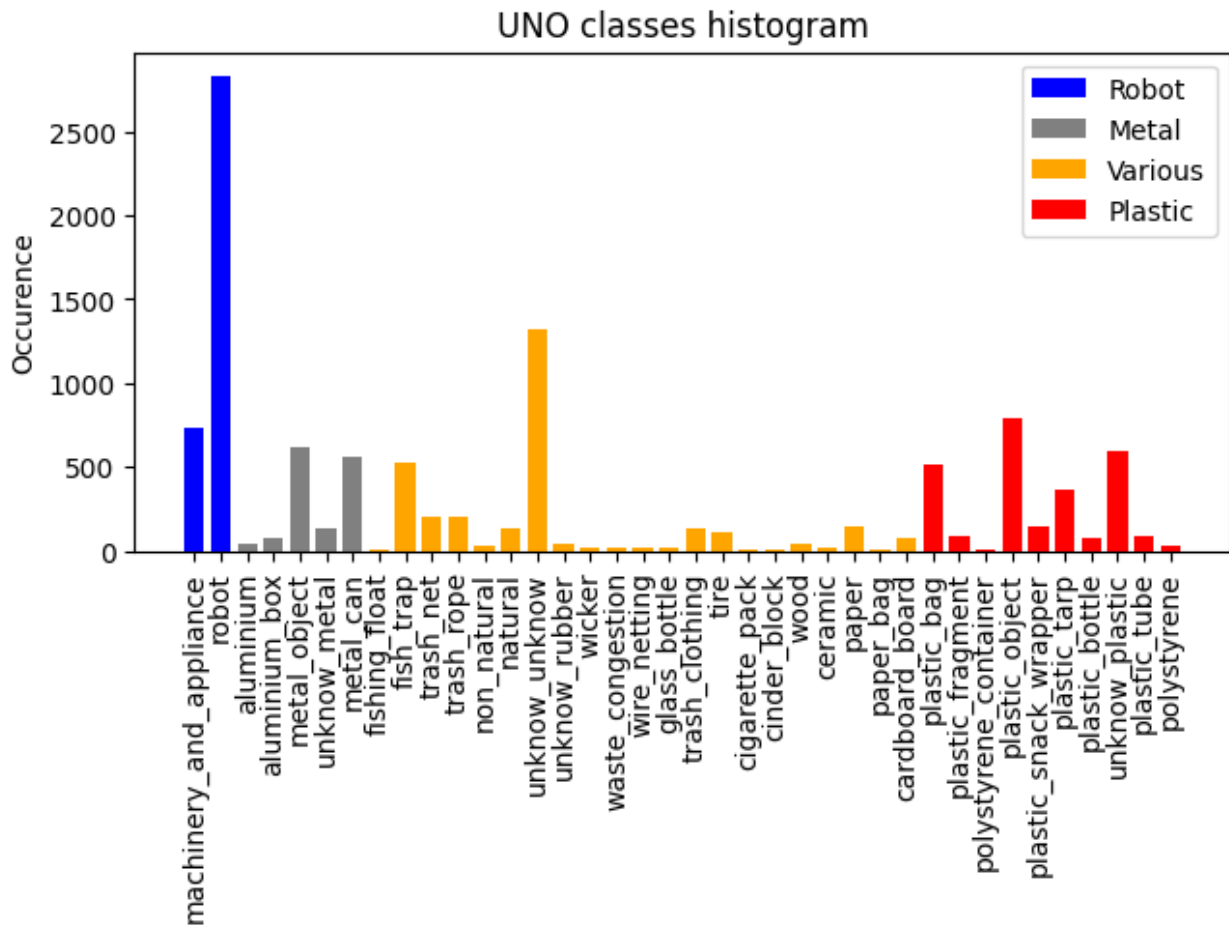


Figure 6 – Class histogram on the UNO dataset

5.1.2 Annotation Process

Three annotators worked on the UNO dataset. A first annotator corrected the position of the bounding boxes from the TrashCAN dataset, focusing on localisation rather than classification. It resulted in the annotations of 10,858 bounding boxes, constituting the previous version of UNO.

Then, two other annotators gave each bounding box a class, using the Roboflow web application². Since the classification work was split in two, it introduced some variability in the labels as annotation relies a lot on the annotator judgement.

² <https://roboflow.com/>

The annotation task was difficult as the pieces of waste are often degraded, partially occluded, and the low luminosity in the images allows the annotators to identify items only when they are relatively close to the robot. Human annotators are often only able to identify the item on a few frames from the video, and then propagate the ground truth class to the other frames, even if they are unable to correctly identify the item on those precise frames. It results in difficult individual frames where the item can be barely seen and yet is annotated precisely.

5.1.3 Variability Considerations

UNO counts many images compared to the other considered datasets. Yet, it also has more frames extracted per video, so the same piece of waste is seen more often in the dataset.

Moreover, the condition of the environment where the items are located is quite constant: Most of the images show an empty area of sand, illuminated by the robot light around the item on the foreground and dark waters in the background.

For those two reasons, according to the annotators, UNO images seem to have quite repetitive patterns and probably less variability than the ones from other datasets.

5.2 MORGANE: Marine Observation and Recycling for Garbage Assessment in Nearshore Environments dataset

MORGANE (Marine Observation and Recycling for Garbage Assessment in Nearshore Environments) is a collection of 1,901 underwater images captured at shallow depths (20cm – 3.5m) in sea harbours and a river. All the waste items have been exhaustively located and classified. Some samples can be seen in Figure 7).

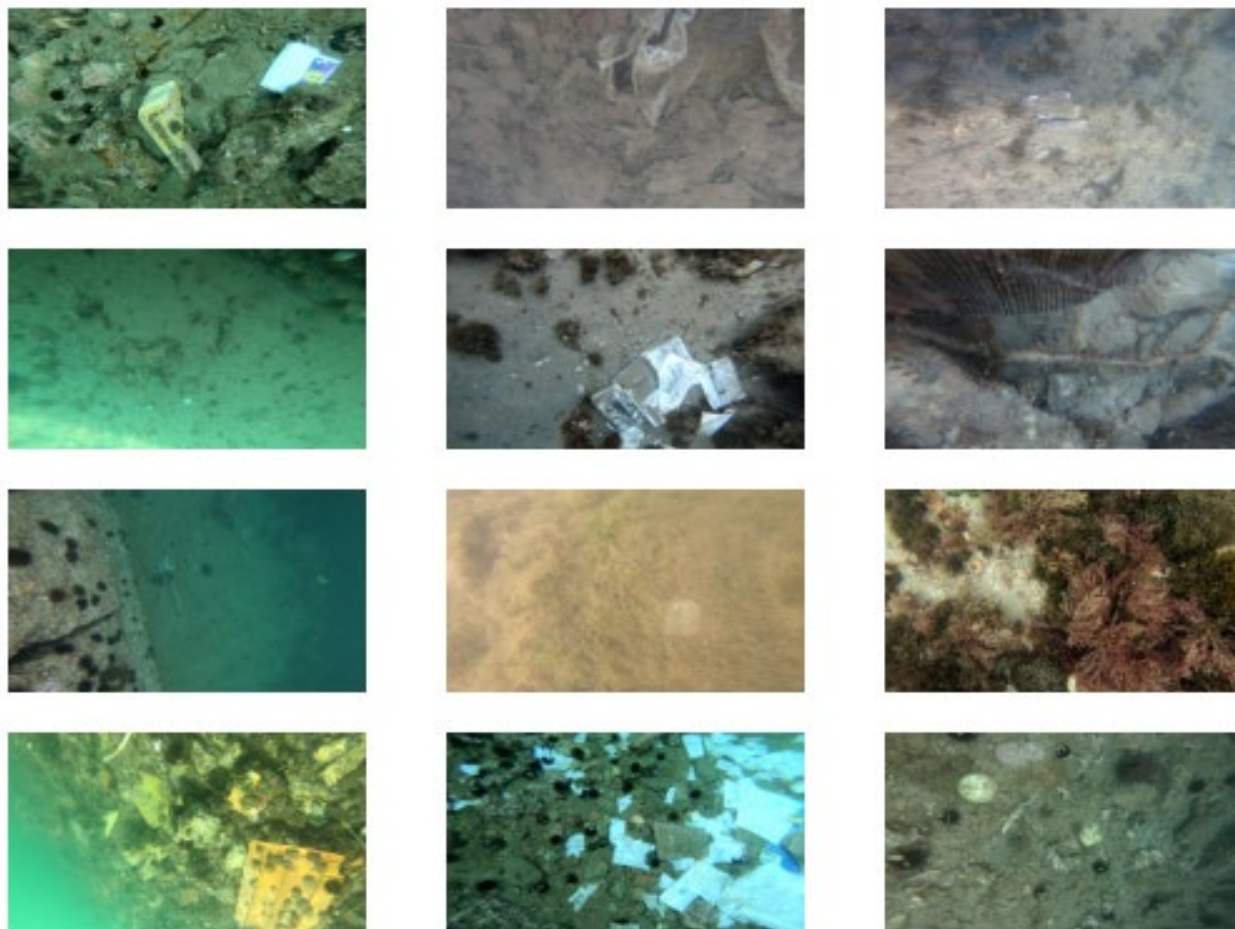


Figure 7 – Random samples of images from the MORGANE dataset

5.2.1 Acquisition condition

Videos were captured using a Go Pro 7 camera attached to a 1-meter-long pole with a rope, which was manually submerged. 575 full HD (1920x1080) video clips (duration between 0.3s and 2mn40s, 100fps) were acquired in various locations around Montpellier (France): sea harbours of Carnon, la Grande Motte, Port-Ariane, Port-Camargue, Palavas and Sète, and in the Lez-river in Montpellier, between December 2022 and February 2023. The operator did not have visual feedback from the camera and was unable to see what was filmed. Video clips were extracted from the SD card and archived in the MP4 format. Meta-data such as the date, GPS position, luminosity conditions, depth and more generally any comment on the sea conditions were archived in an associated file.

As the camera was hanging from the pole, it was not stabilized and oscillated at the end of the rope. In fact, only the camera position was controlled and not its orientation. Since acquisitions were performed on several days, with diverse weather conditions, the luminosity changes a lot between videos. Most of the time, the water was clear and allowed one to see the bottom from the surface. Submerged parts of the docks or land are often visible in the videos.

Given that the portions of the videos containing waste were limited, frames containing waste were manually extracted from the video, with a maximum of one frame per second. In average, 3.31 images were extracted from each video. These frames were labelled following the convention "videoName_minutes_seconds", where "minutes" and "seconds" refer to their temporal position in the video.

As for UNO, the dataset can be separated into splits for training, but its video origin needs to be taken into account.

We defined 29 distinct classes of underwater waste items in the MORGANE dataset. Yet, one can see in Figure 8 that the classes are not uniformly distributed with some very underrepresented classes (notably, the "plastic_lid" class appears only once).

It should be noted that few classes from the dataset have limited practical use. For examples, some classes appear less than 10 times and could be integrated in broader classes, like "plastic_lid" and "plastic_tube" being subclasses of "plastic_object". Despite considering such correction, we chose to retain these classes within the dataset because it is easy to remove them if necessary and they still contain valuable information.

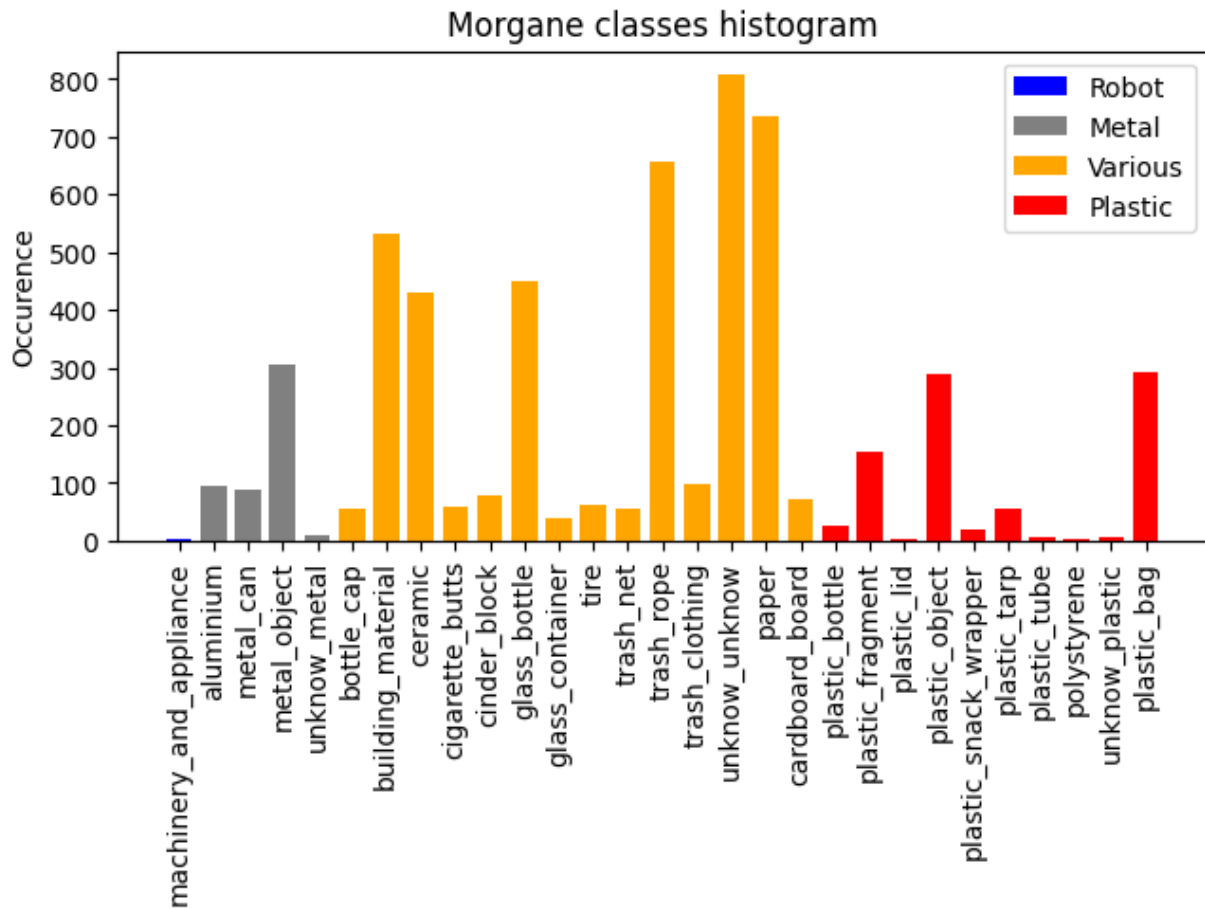


Figure 8 – Class histogram on the MORGANE dataset

5.2.2 Annotation Process

A single operator annotated the MORGANE dataset. It resulted in 5,478 bounding boxes.

This annotation process was challenging due to the inherent difficulty in discerning the nature of underwater waste, which often proved to be a complex task for human annotators. In many cases, a single frame was not enough to conclude on the nature of a waste item, and it required to examine the original video clip around the frame to make a decision. For instance, paper or plastic items often appeared as small, white, sheet-like items and were difficult to distinguish. Similarly, it was very difficult to classify painted items as “metal” or “plastic”. When the annotator judged that one could not draw a conclusion on the item type (class), the resulting annotation was labelled as “unknown_unknown”. On

overall, 22% of the waste items are labelled as “unknow_unknow” in the MORGANE dataset.

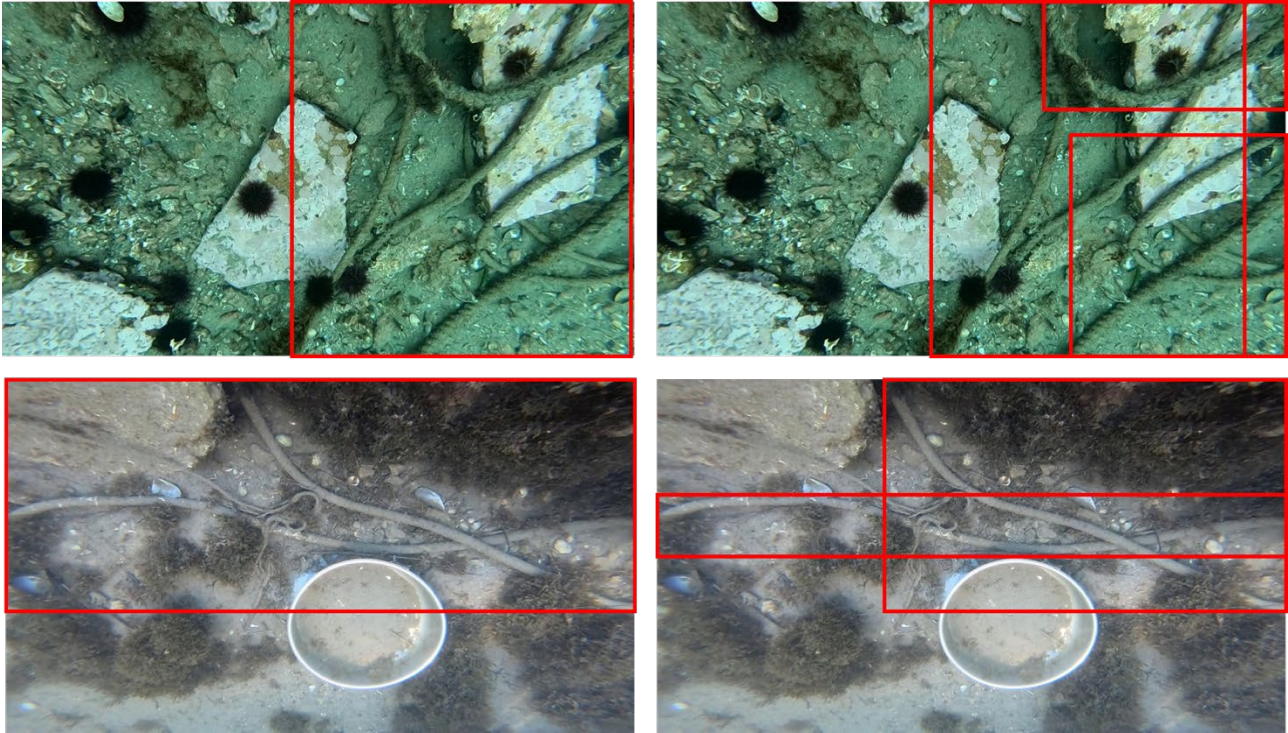


Figure 9 – Example of annotations of a rope. On the left, only one bounding box is used to capture the ropes. On the right, each main part of the ropes has its bounding box. Only the ropes are annotated. Note that the definition of “main part” of a rope rely on the annotator judgement.

As another difficulty, mud, sand or waters often partially occluded the items. In this case, establishing the precise boundaries between the item and its background could be particularly challenging as they were often not well defined, and we did not know if only one or several bounding boxes need to be defined.

Finally, annotating narrow items that still extends a lot in space, like ropes, was challenging. It also led to questions about how many bounding boxes to use in order to annotate a single item. If only one bounding box is used to depict a bunch of ropes, it would often result in an excessively large one compared to the item surface. Hence, we chose to use several bounding boxes to depict the main parts of the ropes. Nevertheless, determining the appropriate number of bounding boxes to use in such cases

remains a complex task and is likely to depend on the operator, as shown in Figure 9.

5.2.3 Variability Considerations

MORGANE is a dataset with a relatively high diversity compared to the two others. Its low number of images extracted per video implies that the same item does not appear too often in the dataset. The diversity is further accentuated by variations in turbidity, depth, luminosity and background mainly due to the various locations where the videos were recorded.

However, the manual selection of frames for inclusion in the dataset may introduce a bias toward cleaner frames, where the items are fully visible and exhibit less blurriness or distortion.

It should be noted that since MORGANE's images were collected in France, the pieces of waste appearing on the frames are different from the ones appearing in the UNO dataset, collected in the Sea of Japan. Similarly, since the videos of MORGANE were recorded in harbours, the nature of the pieces of waste differs: More everyday garbage, like cans and papers, and less big pieces like boat parts and machine parts are found.

5.3 VENICE dataset

The third dataset introduced here is VENICE. It is a small dataset of 683 images captured from mid-depth seabed (7-14m), using cameras mounted on a cable robot of the Robotic Seabed Cleaning Platform in the Venice Laguna. Although these pictures fit perfectly the purpose of the Maelstrom project, they lack diversity to properly represent underwater wastes. Some samples can be seen in Figure 10.

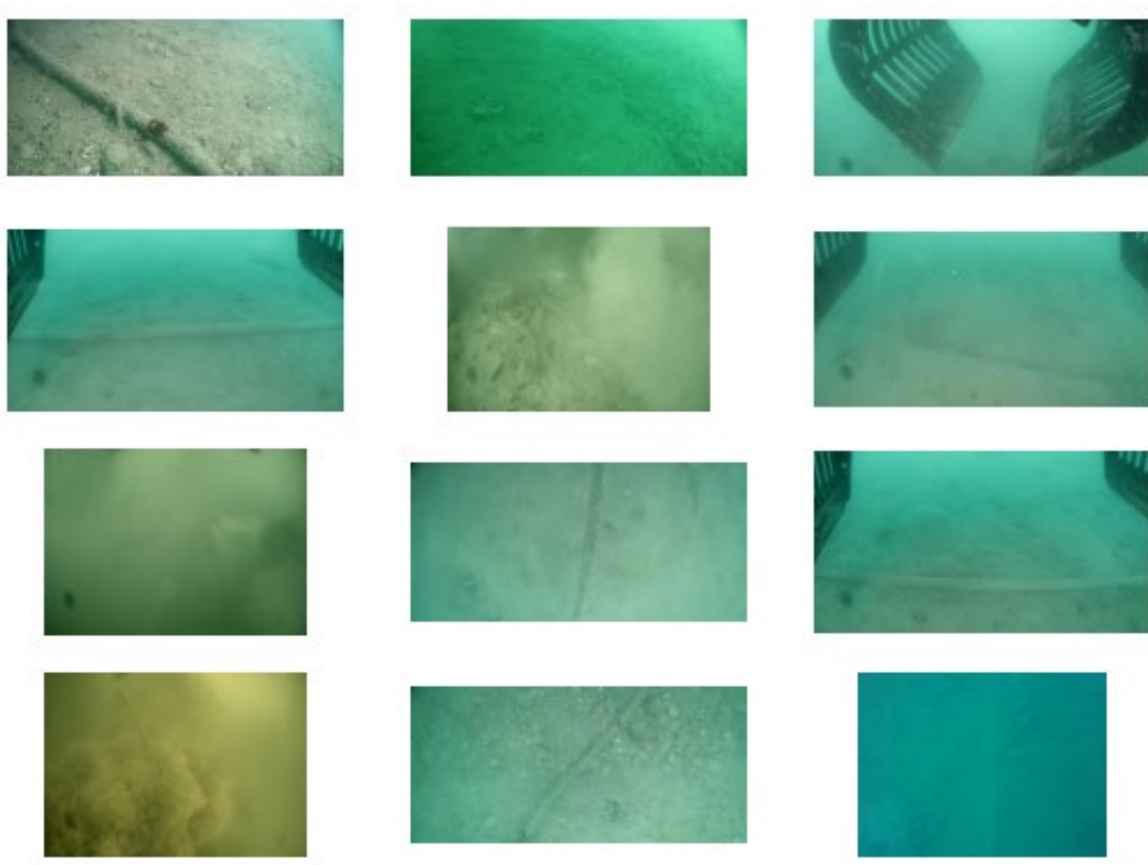


Figure 10 – Random samples of images from the VENICE dataset

5.3.1 Acquisition condition

About 40 videos were acquired by CNRS and Tecnalia during the last Maelstrom cleaning campaign with Robotic Seabed Cleaning Platform, conducted over a few days in June 2023 using a set of five cameras (four HB5C-30 cameras and a low Light HD camera) mounted on the underwater cable robot. The robot was equipped with a gripper for waste retrieval which often appears on the frames, along with plastic tubes, creating non-natural yet non-waste items on the frames.

The frames originate from only two specific locations: The surroundings of the Venice landfill and the site of a former mussel farm, this one located outside the lagoon in a coastal area close to Venice. The surroundings of the landfill were characterized by low luminosity, greenish murky waters, a seafloor covered with algae and a depth of approximately 7 meters.

Conversely, the mussel farm surroundings were characterized by clear, blue waters, and an empty seabed. The average depth on site was 14 meters. Only a limited number of pieces of waste were observed during the campaign.

The videos were extracted and saved in the MP4 format, then manually cut to remove segments where no waste items were visible on the frame and saved in the AVI format. They were trimmed to prevent metadata from being overlaid on the frames and then sampled to reduce the number of frames from each video. The sampling rate varies in between videos, from 2 s to 10 s. On average, about 17.1 images were extracted from each video. Their final resolution goes from 480x640 to 910x1280 pixels.

The VENICE Dataset counts eight distinct classes. Their distribution is shown in Figure 11.

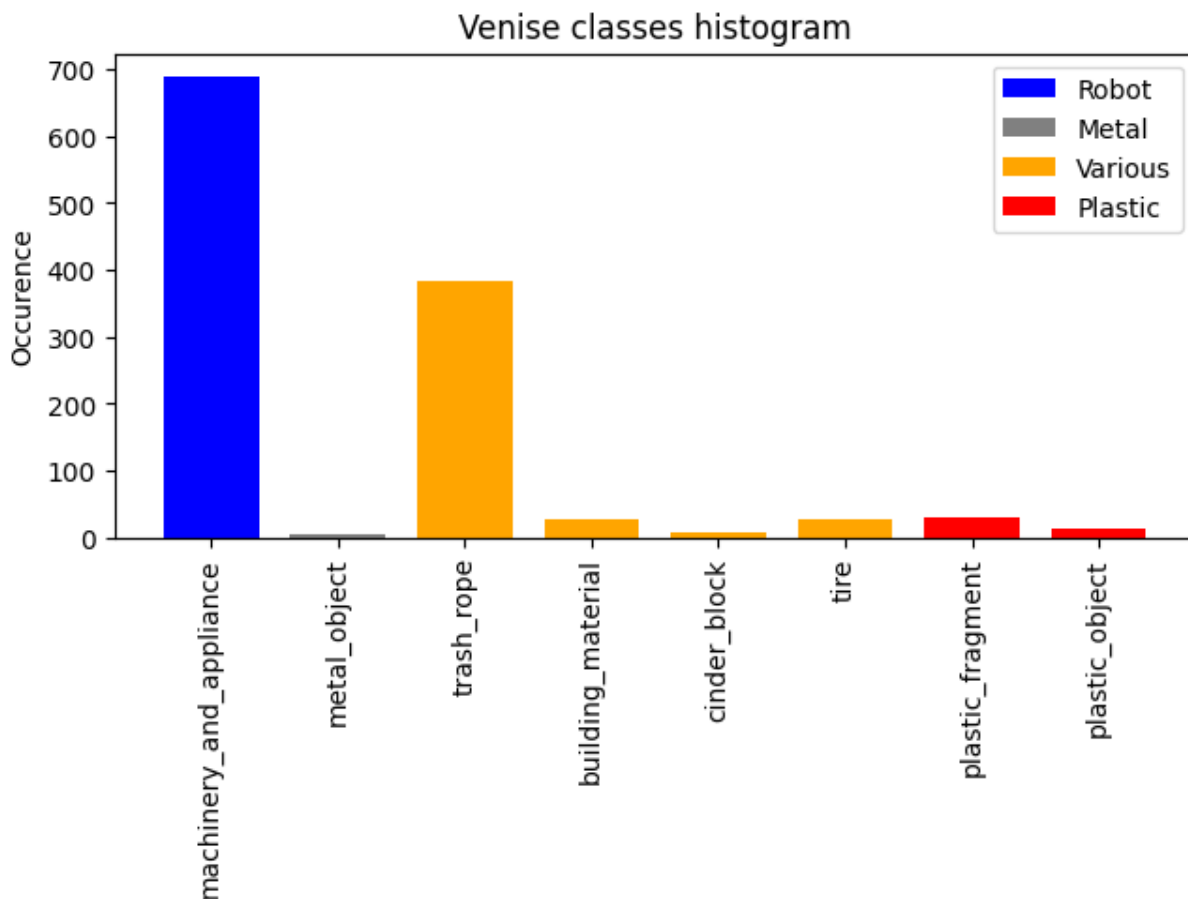


Figure 11 – Class histogram on the VENICE dataset

The classes are not well balanced on the dataset. For instance, the class “Machinery and appliances”, which is mainly composed of the robot parts, is overrepresented as the robot appear on most of the images.

Even if the classes in the VENICE dataset are mostly shared with the other datasets, it should be noted that the waste observed in Venice was substantially different from the one in MORGANE since the robot did not find small pieces of waste like everyday trash, but focused on bigger items like mooring ropes, buoys and tires. This focus can be explained by the fact the robot gripper could hardly pick small pieces, so it was deployed where bigger pieces of waste were encountered. It resulted on 1,183 bounding boxes being annotated.

5.3.2 Annotation Process

The VENICE dataset was annotated by a single annotator, who participated in the campaign in Venice. This significantly facilitated the annotation process compared to the work on MORGANE, as most of the items on the video were retrieved from the seabed and observed at the surface. Thus, the identification of the nature of the items was much easier.

Yet, with mainly ropes to annotate, the problem of the definition of the border between background and widespread items in murky water aroused one more time.

5.3.3 Variability Considerations

A significant limitation of the VENICE dataset lies in its lack of diversity. Given that the majority of the images originate from the same scene and that only a limited number of items were observed, the dataset is predominantly comprised of images depicting the robotic gripper and waste ropes. Up to 90% of the bounding boxes correspond to those two classes, meaning that any improvement on the detection or classification of those two classes may results in greatly increased results and thus lead to the choice of solution with poor generalization.

Given the limited number of images within the VENICE dataset, it was only used as a test set during experiments, i.e., used as a test set for the

evaluation of the models, and has never been divided into traditional train/validation/test splits.

5.4 Comparison between datasets

We used the same classes defined in UNO for MORGANE as most of these classes were present in both datasets. However, a few additional classes, such as "building materials", were introduced because the classes from the UNO dataset were not fully adapted to MORGANE.

This implies that some classes are never shared in between the datasets: MORGANE contains classes that do not exist in UNO, UNO contains classes that are not shared with MORGANE, and VENICE contains classes only from either UNO or MORGANE.

This led to the creation of superclasses to be able to investigate the mismatch between the datasets. To design the superclasses, we tried to meet two requirements: to have equilibrated classes on UNO, and to give superclasses a meaning from a detection point of view.

Following these two principles, we chose to reduce the number of classes to four superclasses: "robot", which contains all the items related to the ROV, the Maelstrom robot, or wasted parts of machinery, "plastic", "metal" which do not need further explanations and "various" which gathers all the pieces of waste of which the core material cannot be decided or is not metal or plastic. The "robot" superclass is mostly not constituted of pieces of waste. Even if most of its occurrences could fit in either the plastic or the metal superclass, we chose to create a different superclass to encompass these non-waste, non-natural objects. However, according to our second principle, we included wasted parts of machinery in the class "robot" as they have a high similarity with the robot parts.

The precise composition of the four superclasses is described in Figure 12.

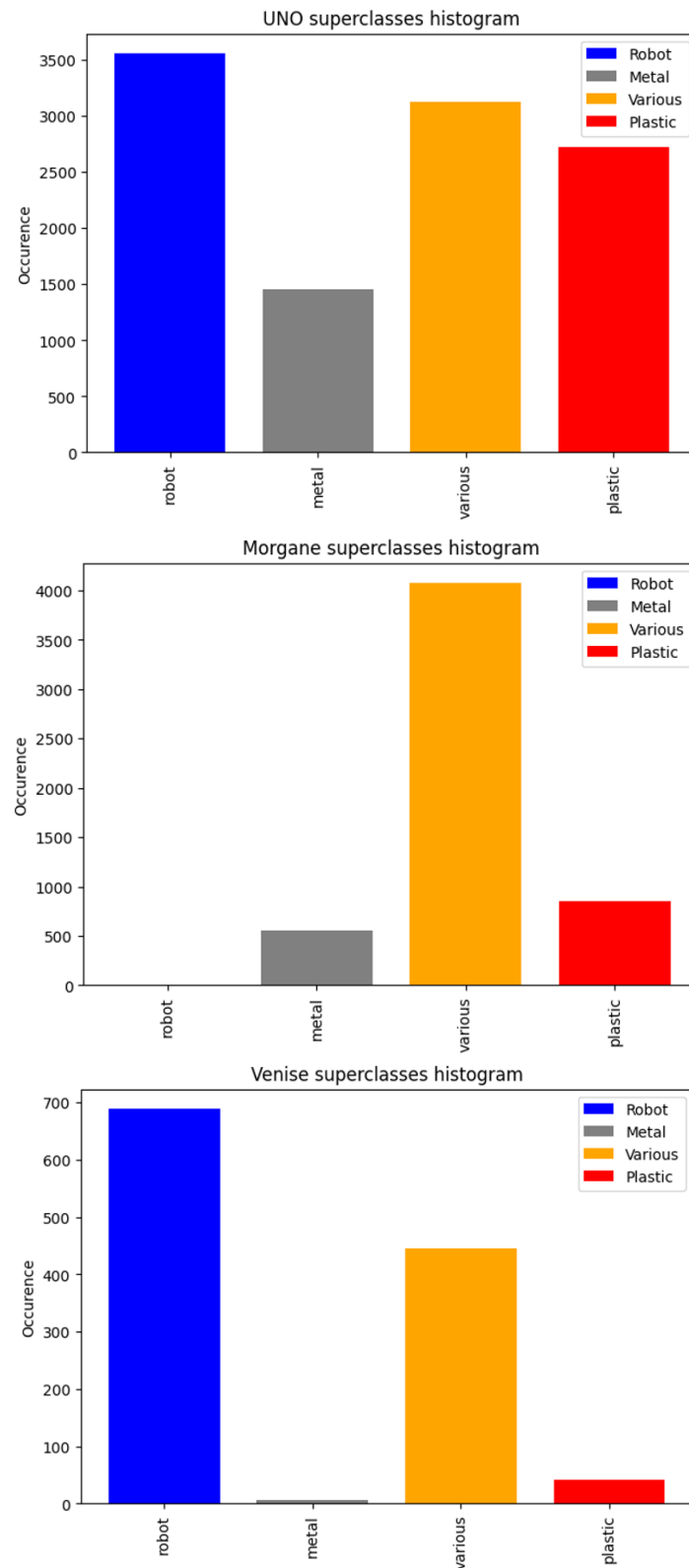


Figure 12 – Superclass histogram on UNO, MORGANE, and VENICE datasets

As shown in Figure 12, the superclasses are not balanced at all on MORGANE and VENICE, which will yield difficulties during the training of the model. Yet, from a real case of domain adaptation perspective, this situation is likely to happen, so we decided not to try to achieve balanced superclasses on every dataset.

Table 1 is a summary table of the main characteristics of the three datasets. It allows us to point out their differences. Note that a class is counted as "shared" if it has at least 50 occurrences in all the datasets to avoid considering classes that are practically absent from a dataset.

Table 1 – Resume table of the datasets characteristics

	UNO	MORGANE	VENICE
Image number	5,985	1,901	683
Images per video	21.45	3.31	17.1
Boxes per videos	1.81	2.88	1.73
Video number	279	575	40
Locations	Sea of Japan	Harbours and river South of France	Venice Laguna
Depth	50 – 9,000m	20cm – 3.5m	7 – 14m
Date	1982 – 2019	2022 – 2023	2023
Number of classes	38	29	8
Shared classes	With MORGANE: 13 With VENICE: 2	With UNO: 13 With VENICE: 1	With MORGANE: 1 With UNO: 2

In Table 1, one can see that even if UNO counts more images than MORGANE, they originated from less videos, leading the same item to be seen more often in the dataset. From this observation, one can wonder which dataset contains more information, and is the most suitable for training. Indeed, a balance point must be found between the diversity of the samples and their number, considering the video origin of the images.

The extreme case is VENICE with 40 videos only, and recording of the same subject using five cameras, which make the number of distinct spatial scenes captured notably lower.

6 Analysis of the datasets

This section is an experimental and theoretical analysis of the performances of detection and classification model on the three datasets presented in Section 5. Its purpose is to compare their links from a domain adaptation perspective. We studied three different domain adaptation setups for the detection task:

- The "raw" setup, where all items are considered.
- The "shared" setup, where only items from shared classes are considered.
- The "included" setup, where the classes that are not shared with the train set are removed from the test sets.

We study one more dataset setup: The one where we do not consider the class "robot". This one is not a classical domain adaptation scheme, yet the nature of our dataset made it interesting.

The underlying idea is to provide different use cases to foster further domain adaptation research on real datasets.

We used the YOLOv8 model in the detection and classification experiments, and a study of dimension was done for the theoretical approach.

To measure the mismatch between the datasets, we used the regret metric. It is defined as the difference between the performance of a deep neural network on a dataset and the performance on the test set from the same dataset when the network has been trained on the train set from another dataset. It takes into account the intrinsic difficulty of the dataset to compute how much performance is lost by switching the set on which the network was trained.

More formally, we note $\text{mAP}_b(c)$ the $\text{mAP}@.5$ of the network³ trained on the train set from dataset b and tested on the test set from dataset c .

We then define the intrinsic difficulty D , as shown in Equation (1), of a dataset b as the $\text{mAP}@.5$ obtained when the network is trained and tested on the train and test splits of the same dataset. This metric is relative to the network n . Note that "having a higher intrinsic difficulty" means having an easier dataset.

$$D_n(b) = \text{mAP}_{b,n}(b) \quad (1)$$

With these notations, the regret between a source dataset s and a target dataset t , relatively to the network n is defined as:

$$R_n(s, t) = D_n(t) - \text{mAP}_{s,n}(t) \quad (2)$$

6.1 Mismatch study

6.1.1 Mismatch on raw data

We first evaluate the performance of YOLOv8 on the datasets for the detection task only. We used YOLOv8 nano version, with pre-trained weights on COCO for the training. We trained for a maximum of 100 epochs, with a patience of 50. We kept the YOLOv8 default settings at its 0.147 version⁴, except we set "single_cls" to True, so the model would train without consideration for the classes of the bounding boxes. This way, YOLOv8 only performed the detection task on every object in the datasets.

We then evaluate the performance of the models on datasets that were not trained on to evaluate mismatch between them.

We call such experiment "detection on raw data" since no modification is made on the labels. A lot of classes are not shared in between datasets and their distribution is never identical. The domain adaptation case that

³See : https://papers.nips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

⁴ <https://github.com/ultralytics/ultralytics>

is considered here is a very complex one, as the datasets only share the same purpose and a few classes.

Table 2 – mAP@.5 of YOLOv8 for different train/test datasets on raw data

Train dataset	Test dataset			
	UNO	MORGANE	VENICE	MIX
UNO	0.643	0.204	0.445	0.545
MORGANE	0.593	0.479	0.262	0.570
MIX	0.626	0.486	0.476	0.601

In Table 2, we see that the network performs better on UNO than on MORGANE leading us to think that the UNO dataset is easier than the MORGANE one. Yet, the good results on UNO seem not to generalize well to MORGANE. The contrary is not true: The model trained on MORGANE performed well on UNO too. While the regret from UNO to MORGANE was $R_{raw}(u,m) = 0.275$, the regret from MORGANE to UNO is $R_{raw}(u,m) = 0.050$, which shows that almost no performance was lost when training on MORGANE and testing on UNO. It could be explained by the fact that the greater diversity of MORGANE may lead to better generalization with respect to other simpler datasets.

About the VENICE dataset, the model trained on UNO did better than the one trained on MORGANE. This may be due to the high similarity in the acquisition of UNO and VENICE data: Both datasets are composed of images recorded by camera mounted on underwater robots, which implies parts of the robots appearing in the frames and the orientation of the camera relatively to the ground being constant. Since VENICE dataset is not big enough to train on, no regret can be computed.

Because VENICE mainly contains items from the "robot" class, which is well represented on UNO, and the "trash_rope" class, which is well represented on MORGANE, we decided to create a fourth dataset composed of the union of the MORGANE and the UNO datasets. The idea was that combining the datasets would allow us to improve results by using all the data we had. We call

this new dataset MIX. Note that as UNO is much bigger than MORGANE, the MIX dataset tends to be more similar to UNO than to MORGANE.

Table 2 gives results for MIX which are indeed slightly increased when training on MIX and testing on MORGANE and VENICE, but not when testing on UNO. As a test set, MIX's performances are in between those of MORGANE and UNO, as is its intrinsic difficulty.

6.1.2 Mismatch without the robot class

When creating the superclasses, we wondered whether to include the robot superclass or not, as it does not really depict waste but non-natural objects, and if its inclusion had any impact on the training.

We annotated the robot and included it in the labels at first because we thought it made more sense for the network to detect all non-natural objects, rather than having to consider as background objects very similar to the ones it had to detect. Yet, to include the robot does not make sense for the task of waste detection, as the robot is not a piece of waste.

In order to check if this choice had any impact, we trained models using datasets from which we removed any occurrence of the robot superclass. Note that the superclass robot contained a few occurrences of wasted machinery, that were removed along with the ROV and robot occurrences, leading to false background annotations on some frames. We considered those negligible. The robot superclass is also overrepresented on VENICE and UNO and almost not represented on MORGANE.

Table 3 – mAP@.5 of YOLOv8 for different test/train datasets, data without the robot superclass

Train dataset	Test dataset		
	UNO	MORGANE	VENICE
UNO	0.524	0.146	0.091
MORGANE	0.432	0.465	0.051

As can be seen in Table 3, the mAP@.5 is lower when training on data without the robot class (see Table 2), especially considering the UNO and

VENICE datasets. On the contrary, it does not affect much the results on the MORGANE dataset.

It affects the quality of the training on UNO: When testing on MORGANE the model trained on UNO, the results decreased, despite the MORGANE test set having practically not changed since.

In Table 3, we observed that without the superclass robot, the model trained on MORGANE performed worse on the UNO dataset, which means that some of the robot bounding boxes were previously correctly detected by the MORGANE's model, despite the robot superclass being practically absent from the training set.

Computing regret, we see that it increased compared to the case with the robot class. Indeed, we have $R_{\text{raw}}(u, m) = 0.275$ which becomes $R_{\text{norobot}}(u, m) = 0.319$ without the robot class and $R_{\text{raw}}(m, u) = 0.05$ which becomes $R_{\text{norobot}}(m, u) = 0.092$. The fact the regret increased in between MORGANE and UNO despite removing a non-shared class show that it affected negatively the results.

Since the results decreased when removing the robot superclass, we conclude that the network cannot properly differentiate wastes items from robot parts, and that the detection of non-natural objects makes more sense than the detection of waste for the network.

6.1.3 Mismatch on shared classes

Conversely, we studied detection experiments on data with shared classes. The goal of this section is to simulate the common cases of domain adaptation studied in literature, where all classes are shared. In the present case, we could not remove the items directly from the images, so the detector had to consider them as background.

The experimental setup was the same as for the raw data case, except that we removed the bounding boxes that are not shared in between datasets from the labels. A class was considered as shared if it has more than 50 occurrences in both UNO and MORGANE datasets. We then trained YOLOv8 on the new labels and performed test across the datasets. The crossed test

was only performed between MORGANE and UNO, since very few classes were shared with VENICE.

Table 4 – $mAP@.5$ of YOLOv8 for different test/train datasets, shared classes

Train dataset	Test dataset	
	UNO	MORGANE
UNO	0.363	0.090
MORGANE	0.327	0.324

As shown in Table 4, results with shared classes are significantly lower than with raw data, highlighted in Table 2.

Computing regrets, we found $R_{\text{shared}}(m,u) = 0.036$ and $R_{\text{shared}}(u,m) = 0.234$ which are slightly lower than for the raw data case in absolute ($R_{\text{raw}}(u,m) = 0.275$ and $R_{\text{raw}}(m,u) = 0.05$). Thus, even if aligning classes between datasets could lead to reduced mismatch, we could not conclude it does.

Multiple factors can explain the decreased results on the shared-class datasets. Indeed, the fact that lots of bounding boxes were removed may lead to a scarcity of example of items. However, the major factor for decreased results is probably that items from classes that are not shared still exist in the images and are labelled as background.

We see that this case does not make much sense. However, it is common to assume that the classes are shared in between datasets when studying data adaptation. For our application, this case does not happen.

6.1.4 Mismatch in included classes

Compared to the previous section, addressing the class imbalance by detecting items only of classes that were in the train dataset would make more sense.

We created new labels for our datasets, where we removed from the test sets the classes that are not shared with the train set. This way, the model is not tested on classes it has never seen before. This modification corresponds to a change in performance metrics: The training is not

impacted, and the network itself does not change from the case where all labels are left in the test set.

We limited the study to UNO and MORGANE since VENICE shares all its significant classes with UNO, and the only class it does not share with MORGANE is the robot one, which leads the inclusion case (Section 6.1.4) and the non-robot case (Section 6.1.2) to be the same in this situation.

Table 5 – $mAP@.5$ of YOLOv8 for different test/train datasets, included classes

Train dataset	Test dataset	
	UNO	MORGANE
UNO	0.670	0.200
MORGANE	0.313	0.482

Compared to the results in Table 2, Table 5 shows that the results decrease when testing on a dataset on which the network was not trained. This lead us to think that the network can detect items from classes it was not trained on.

To investigate this hypothesis, we defined the mean recall mR to measure the performance of the detector for each class. Mean recall consists of the integral of the recall as a function of the confidence threshold used to sort the results of the network. It is inspired from the mAP measure, which could not be used here since the false positive measure per class does not make sense for a detector. To calculate the mean recall, we computed the true positives as the correctly detected bounding boxes of a class and the false negatives as all the bounding boxes of the same class that were not detected. This allowed us to compute the recall for each class and confidence threshold. To get rid of the confidence threshold dependency, we integrated over it.

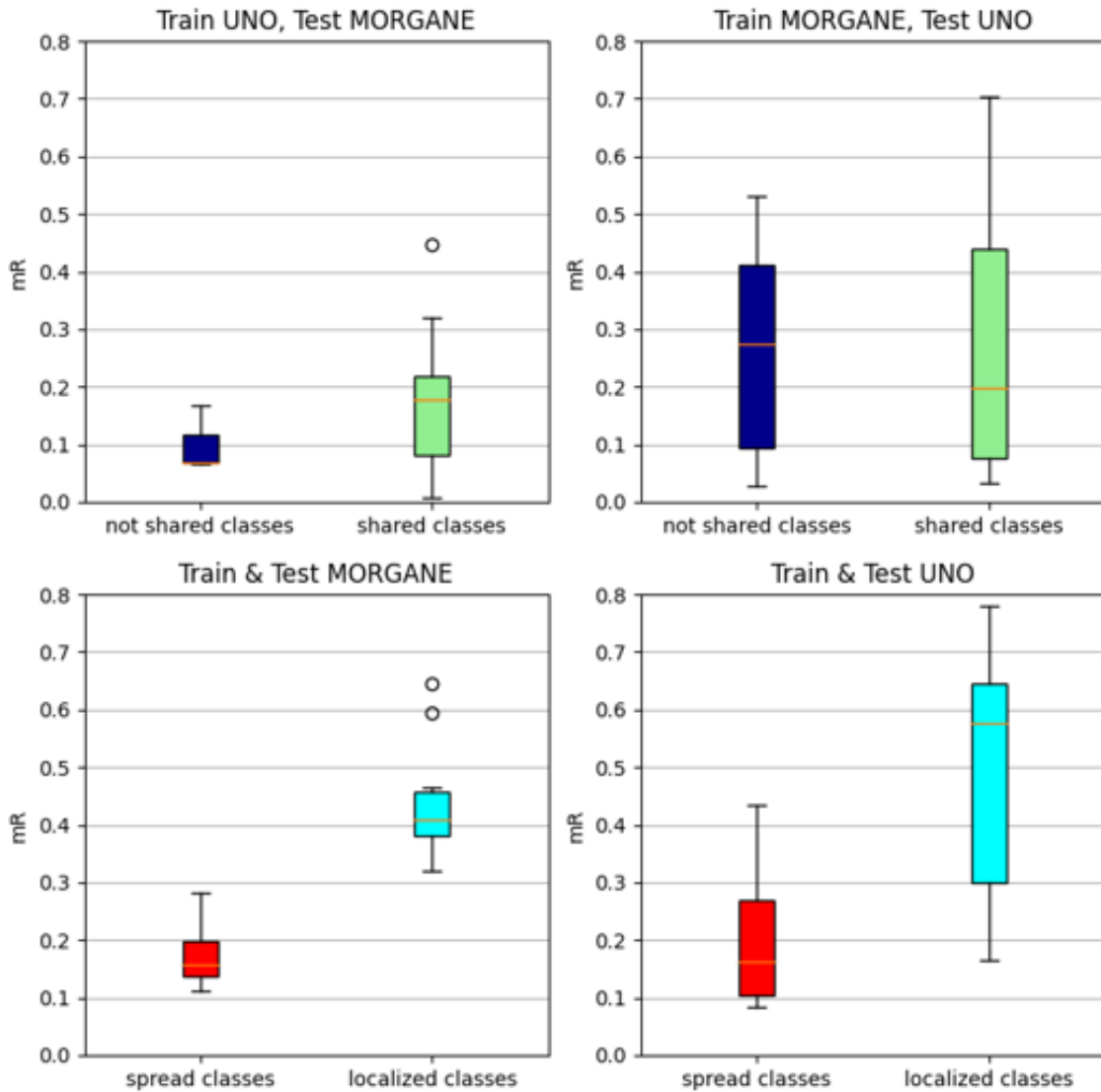


Figure 13 – Boxplot of the mR in different situations

We computed mR on all the classes from UNO and MORGANE for both the networks trained on UNO and MORGANE. Figure 13 shows box-and-whisker diagrams representing the distribution of mR over the dataset classes, depending on if they are shared with the training set or not, or if they are constituted of spread items or not. We only considered classes that have more than 10 occurrences on the test set. As in previous sections, classes were labelled as shared if they have more than 50 occurrences on both train and test datasets (without considering splits). As illustrated in Figure 13, shared classes are also detected as non-shared classes. Other

criteria have much more impact on mR than the fact a class is shared or not, like the shape of the bounding box, for example. Indeed, classes that have, by definition, long and spread bounding boxes have in average lower mR than other classes. For instance, classes such as "trash_rope", "trash_net", "trash_clothing", and "plastic_tarp" can be considered as "spread". This can probably be explained by the fact the Intersection over Union metric is stricter for elongated bounding boxes and because the items in these bounding boxes tend to occupy a small area of the bounding box.

6.2 Classification task

In this section, the argument "single_cls" is removed from YOLOv8 training function, so that the model is trained on both the detection and the classification tasks simultaneously. We call "classification", the task of both detecting and classifying the objects.

In order to compare models on the classification task, we trained YOLOv8 with the superclasses and not the natural classes. It must be noted that UNO and MORGANE are not equal regarding the superclasses, as the superclasses were designed to fit UNO's distribution and happened not to fit MORGANE's distribution well (see Figure 12). Since the superclasses may not fit an unknown target dataset in a real case application, we decided to keep those superclasses despite the fact they are not adapted to MORGANE.

Table 6 – $mAP@.5$ of YOLOv8 for different test/train datasets

Train dataset	Test dataset			
	UNO	MORGANE	VENICE	MIX
UNO	0.386	0.054	0.128	0.314
MORGANE	0.153	0.211	0.017	0.171
MIX	0.444	0.185	0.130	0.417

Classification results on our datasets are gathered in Table 6. As it was expected considering the fact the classification task is more complex than the detection one, results are poor. The incapacity to leave apart some

classes during detection may be related to the poor results on the classification task.

As for the detection task, the intrinsic difficulty of UNO is higher than the one of MORGANE, meaning the UNO dataset is easier to train on. However, the regrets are not the same than for the detection task. We do not observe the same asymmetry between UNO and MORGANE: here $R_{classif}(m,u) = 0.233$ and $R_{classif}(u,m) = 0.157$, so performance decreases when training on MORGANE and testing on UNO rather than when training on UNO and testing on MORGANE. It was the opposite way around for the detection task. The fact that the superclasses robot and metal are underrepresented in MORGANE probably explains why testing on UNO gives such bad results, even though the dataset is not too hard to train on. Nevertheless, this phenomenon affected less the detection task. We think that the detection task is easier to generalize to unknown classes than the classification task, since detection relies on differentiating the background from the general notion of waste, when the classification task needs precise information about the class in order to identify it properly.

On VENICE, results were similar to the detection ones. The model trained on UNO did better than the one trained on MORGANE. Training on the mixed dataset improved a little the results obtained when training on UNO.

Table 7 – mAP@.5 of YOLOv8 for different superclasses and test sets; Training on UNO

Superclass	Test dataset			
	UNO	MORGANE	VENICE	MIX
Robot	0.612	X	0.431	0.599
Metal	0.454	0.049	0.000	0.349
Plastic	0.333	0.064	0.056	0.213
Various	0.144	0.061	0.020	0.094

Table 8 – mAP@.5 of YOLOv8 for different superclasses and test sets; Training on MORGANE

Superclass	Test dataset			
	UNO	MORGANE	VENICE	MIX

Robot	0.104	X	0.034	0.092
Metal	0.354	0.200	0.000	0.297
Plastic	0.097	0.252	0.003	0.103
Various	0.149	0.490	0.030	0.191

Table 7 and Table 8 take a closer look on how well the networks performed on each superclass.

Training on UNO, the network could detect and classify all superclasses. It performed at best on the robot superclass since it is overrepresented on the dataset and worst on the various superclasses. It may be because this superclass is the one that makes less sense as it is a gathering of random other classes. As previously stated, the network could not generalize these results on MORGANE's superclasses. On VENICE, the network trained on UNO could detect the Robot superclass very well. However, the other classes were not correctly classified.

Training on MORGANE, the superclass Various was well classified since it is overrepresented on the dataset. The network avoided the option to classify all objects as "various", which gives a 75% accuracy on the dataset if we do not consider detection errors. It was able to generalize a bit to UNO, especially for the superclasses Metal and Various. However, no superclass was correctly detected on VENICE.

Considering the difficulty for human to annotate underwater waste, the poor classification results are not surprising. Indeed, a lot of items are impossible to classify for human. This is due to different phenomena.

The first one is that the videos from which the images originate follow a pattern of approaching an item and then move away from it. Hence, on the first and last frames of the video, it is sometimes impossible to identify the object since it is too far from the camera. In these cases, the ground truth label was propagated from the frames where the item was possible to identify to the ones where it was not. We stopped propagating when even the detection task was impossible for the annotator.

Another phenomenon is that sometimes it is the movement of the object on the video that allowed the annotator to decide to which class this object belongs. The class "paper" is a good example: Paper items often sway in water currents and this fact can help to distinguish them from harder plastic items. In this case, it is the succession of frames that allows the identification, and the class cannot be decided from a single frame. Still, the ground truth was propagated as before.

Finally, the fact that we are classifying pieces of waste according to their main components, which can be degraded or covered with biological substances, lead to a substantial difficulty in the classification task. Whereas the detection task may have relied on shape, differentiating natural shapes from artificial shapes, the model needs to learn texture to fulfil the classification task. However, texture is not always visible because of the lack of luminosity or the presence of algae or sediments on the item, which may thus complicate the classification task.

This led us to wonder whether the choice to sort superclasses by the core materials is a good idea. It could be replaced by other sorting systems, such as size for example. Sorting pieces of waste by size comes with other difficulties, but may make more sense for the network, and for the process of picking up the piece of waste using a robot since object of too small or too large sizes cannot be picked by the gripper. We did not investigate this lead any further, for the moment at least.

7 Conclusion

MAESLSTROM WP3 aims at developing, implementing and integrating the core technologies for the automated system for marine litter removal: the Robotic Seabed Cleaning Platform. The present deliverable dealt with the cable robot autonomous control and machine learning for litter identification.

Section 4 summarized the control system of the cable robot of the Robotic Seabed Cleaning Platform, where tools and capabilities enabling to reduce the complexity of the robot piloting and associated operator fatigue were

introduced. Then, in Section 5, three datasets were presented: UNO, MORGANE, and VENICE. They all comprise images of underwater objects captured in natural environments. These datasets were collected from various locations and times, resulting in significant variations in data acquisition conditions and, consequently, in the characteristics of the datasets themselves. Despite their differences, they share the common goal of providing means to detect and classify underwater litter.

The initial version of the UNO dataset was introduced during the 5th Workshop on Computer Vision for Analysis of Underwater Imagery (CVAUI), held in conjunction with the International Conference on Pattern Recognition (ICPR) in Montreal in 2022. Both the dataset and the accompanying research paper are available for download at <https://lirmm.fr/uno>.

Thanks to the hiring of a 6-month fixed-term employee, we were able to update the final version of the UNO dataset (where all classes and superclasses have been created), acquire and annotate images for the MORGANE dataset, and annotate the VENICE dataset. A paper is currently being written to introduce the UNO updated version, MORGANE, and VENICE datasets.

Finally, Section 6 presented experimental and theoretical analyses of the performances of detection and classification models on the three datasets presented in Section 5. The main goal was to compare them from a domain adaptation perspective. The underlying idea is to provide different use cases to foster further domain adaptation research on real datasets. From a research standpoint, leveraging domain adaptation methods on these datasets could enhance the detection/classification model. Additionally, we could utilize them to develop novel methods for domain adaptation, hence improving suggestions for the operator to detect and classify marine litter.

8 References

- Barrelet, C., Marc, C., Gérard, S., Vincent, C., & Marc, G. (2022). From TrashCan to UNO: Deriving an Underwater Image Dataset To Get a More Consistent and Balanced Version. *CVAUI 2022-5th Workshop on Computer Vision for Analysis of Underwater Imagery@ ICPR*.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*.
- Hong, J., Fulton, M., & Sattar, J. (2020). TrashCan: A Semantically-Segmented Dataset towards Visual Detection of Marine Debris. *arXiv preprint arXiv:2007.08097*.