university of groningen

faculty of science and engineering

# Solid waste detection in context: Merging Datasets and using Open-Set Vision Models

Alexandros Sokratis Charisis

university of groningen / faculty of science and engineering

**University of Groningen**

# Solid waste detection in context: Merging Datasets and using Open-Set Vision Models

**Master's Thesis**

To fulfill the requirements for the degree of
Master of Science in Artificial Intelligence
at University of Groningen under the supervision of
Dr. M.A. (Matias) Valdenegro Toro (Artificial Intelligence, University of Groningen)
and
Dr. M. (Matthia) Sabatelli (Artificial Intelligence, University of Groningen)

**Alexandros Sokratis Charisis (s3992853)**

July 17, 2024

# Contents

**Page**

**Acknowledgements** **5**

**Abstract** **6**

**1 Introduction** **7**
1.1 Impact of solid waste on aquatic life and ecosystems . . . 7
1.2 Economic impact of inefficient solid waste management . . . 8
1.3 Degradation of solid plastic waste into microplastics and impact on living organisms 8
1.4 100% Bio-degradable plastic alternatives . . . 10
1.5 Segregation facilities and working conditions . . . 10
1.6 Waste management stages . . . 11
1.7 Artificial Intelligence applications for waste management . . . 11
1.8 Challenges of Artificial Intelligence applications in waste management . . . 13
1.9 Purpose of the Thesis . . . 13
1.10 Research Questions . . . 15

**2 Background Literature** **16**

**3 Methods** **17**
3.1 Data . . . 17
3.1.1 Trash Annotations in Context Dataset (TACO) . . . 17
3.1.2 UAVVaste . . . 19
3.1.3 Cigarette Butts . . . 19
3.1.4 PlastOPol . . . 19
3.1.5 Other Datasets . . . 20
3.2 Models . . . 21
3.2.1 Closed-Set vs. Open-Set Object Detection Paradigms . . . 21
3.2.2 Closed-Set Object Detectors . . . 21
3.2.3 Pre-trained Open-Set Object Detectors . . . 23
3.2.4 Instance Segmentation . . . 25

**4 Experimental Setup** **27**
4.1 Tools and Technologies . . . 27
4.2 Combined Dataset . . . 27
4.3 Label space taxonomy . . . 27
4.3.1 Label Mapping . . . 28
4.4 Datumaro framework . . . 31
4.5 Custom harmonisation script . . . 31
4.6 Custom merging script . . . 31
4.7 Hugging Face platform . . . 32
4.8 Meta's Segment Anything Model (SAM) - demo . . . 34
4.9 Roboflow platform . . . 35
4.9.1 Chosen License . . . 35
4.9.2 Training YOLO-NAS . . . 35

4.9.3 GroundingDINO - Auto-label . . . . . 35

**5 Results** **40**
5.1 What are the challenges of integrating multiple public datasets for solid waste detection? 40
5.2 How can we impose a label hierarchy to each dataset ensuring the label space is harmonized and perform the merging? . . . . . 40
5.2.1 Datumaro . . . . . 40
5.2.2 Custom scripts . . . . . 40
5.3 How can we automate solid waste detection with auto-labelling techniques using language-vision models? . . . . . 40
5.3.1 Grounding DINO . . . . . 40
5.3.2 YOLO-World . . . . . 46
5.4 How closed-set perform compared to open-set computer vision models on the combined solid waste dataset? . . . . . 51
5.4.1 YOLO-NAS . . . . . 51
5.4.2 Segment Anything Model (SAM) . . . . . 56
5.5 Discussion of results . . . . . 60
5.5.1 Segment anything Model (SAM) . . . . . 61
5.5.2 YOLO-NAS . . . . . 62
5.5.3 YOLO-World . . . . . 62
5.5.4 GroundingDINO . . . . . 62
5.5.5 Auto-labelling capabilities with open-set detectors . . . . . 62

**6 Conclusion** **63**

**References** **64**

**Appendices** **68**

# Acknowledgments

Given this opportunity, I would like to thank my parents Costas and Felicia, my sister Anna, and my wife Natalia for their loving support every step of the way, their patience and for believing in me. Without their unwavering support, this journey would not be feasible to embark, let alone complete.

I would also like to thank my supervisor, Dr. Matias Valdenegro Toro (Artificial Intelligence, University of Groningen), for his valuable support and guidance along the way. It is an absolute joy collaborating with him and learning from him the scientific ways of working. It is also amazing witnessing myself how a tiny idea can take up a solid form and turn into this present work, with consistency and supportive supervision.
Moreover, I would like to thank my second supervisor Dr. M. (Matthia) Sabatelli for reviewing my final project proposal and my final project thesis.

Last but not least, I would like to thank the FSE's AI and CCS Academic Advisor Rachel van der Kaaij for her support along the way.

# Abstract

Waste management gets harder the more cities grow and plastic waste degradation leads to unwanted results like ending up in the top of the food chain, with tremendous impact on living organisms and aquatic ecosystems. Efficient waste detection, collection, classification, segregation, and recycling are all important for reducing the amount of waste ending up in landfills and tackle the issue of degradation of plastic waste ending up in our ecosystems.

Existing publicly available datasets of 2D solid waste images are small in size and not diverse enough to cover all natural settings. This thesis proposes the harmonious merging of three public solid waste datasets to increase the quantity and variability of the training data. It compares various computer vision models like YOLO-NAS, Grounding DINO, YOLO-World and Segment Anything Model on solid waste detection through 2D image data.

Qualitative analysis showed a robust combined multipurpose solid waste dataset, poor performance for YOLO-NAS, medium to high performance for YOLO-World/Grounding DINO respectively and high performance for Segment Anything model. Findings showed that open-set vision language models can accelerate image annotation and as a result the automation of solid waste detection.

# 1 Introduction

In this day and age, it is becoming more obvious how important efficient waste management really is, as the total global population increases, and the cities' inhabitants also grow in numbers. To understand the magnitude of the problem a World Bank Group report estimated that by year 2050 municipal solid waste will increase approximately 69% or 1.39 billion tones compared to year 2016 (Kaza et al., 2018). Solid waste is a human-generated type of waste in solid state, which includes household, industrial, construction, electronic, hazardous, agricultural, and medical waste amongst others (Kaza et al., 2018).

The findings of a waste generation and recycling report from (Eurostat, 2023), noted a *7.6kg/capita* increase in plastic packaging waste generation and a *3.9kg/capita* increase in plastic packaging waste recycling from year 2011 to 2021. This means that besides Europe's significant efforts to increase recycling rate (*38% increase*), we are still far behind the rate of plastic packaging generation.

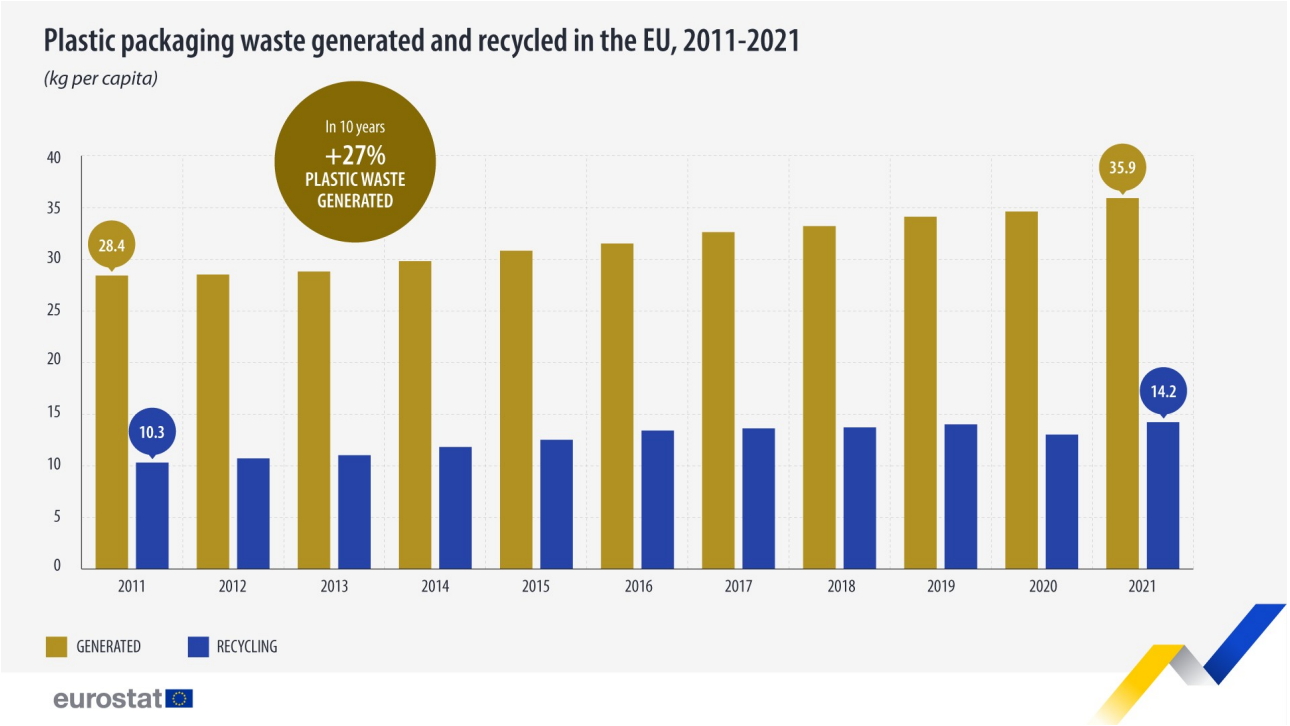



Figure 1: Plastic packaging waste generation versus recycling (per capita)

Source: (Eurostat, 2023)

## 1.1 Impact of solid waste on aquatic life and ecosystems

In an article (National-Geographic-Society, 2019), the authors reported an incident in which a young whale died because its stomach was full of roughly *40kg* of solid plastic waste, which eventually killed the animal, as the stomach could not fit enough food to process and starved to death. According to the authors of the factsheet at the *UNESCO, Ocean Conference in 2017* (UNESCO, 2017), roughly *100.000* marine mammals, 1 million sea birds, countless fish and marine turtles amongst others, die

every year, with the culprit being plastic pollution. *80%* of all marine pollution comes from human activities in the land. This is a problem (also depicted in figure 2), that we need to address more efficiently and take action to minimize solid waste generation as well as collect existing solid waste.

Figure 2: Marine life threatened by human generated solid waste

Source: (National-Geographic-Society, 2019)

## 1.2 Economic impact of inefficient solid waste management

Both adequately collecting and properly disposing waste is significant for tackling the above-mentioned problems (Majchrowska et al., 2022). Moreover, the economic impact of not efficiently collecting and disposing waste becomes multiple time more costly, compared to investing funds on constructing or improving waste management plants (Kaza et al., 2018).

## 1.3 Degradation of solid plastic waste into microplastics and impact on living organisms

Efficient waste detection, collection, classification, segregation, and recycling are all important processes for reducing the amount of waste that ends up in landfills (Pawaskar & Dhanya, 2022; Vierah Hulley, 2020), but also prevents waste disintegration/degradation as a result of the minimization of

waste spillage. Especially in the case of plastic waste degradation, with micro-plastics *(< 5mm)* accumulating in the top of the food chain, *including humans* (Osman et al., 2023)), having a devastating impact on living organisms and aquatic ecosystems (Wayman & Niemann, 2021), (Issac & Kandasubramanian, 2021) (see figures 3 and 4). It is of imperative importance to minimize waste spillage as much as possible.

We have already mentioned above that microplastics have been found in humans, within biological samples such as blood and saliva (Osman et al., 2023), but also in every part of the body that researchers have focused on, even reproductive organs (Stone, 2024). There are findings linking microplastics with inflammatory, intestinal, cardiovascular, cancer and infectious diseases amongst others (Osman et al., 2023).

Figure 3: Solid waste in natural settings containing degrading plastics

Source: *left image*(Unsplash, 2019) *right image*(Unsplash, 2021)

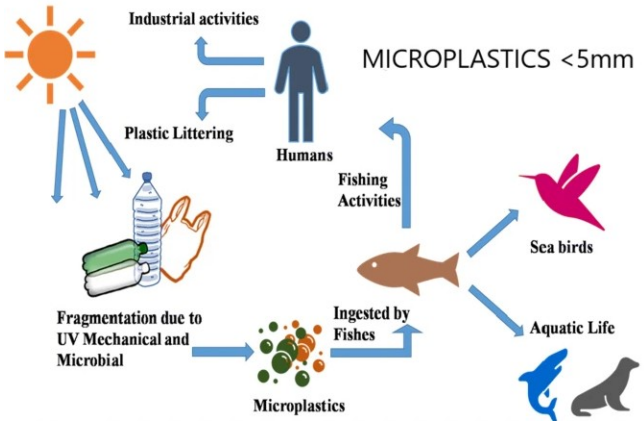



Figure 4: Effects of microplastics in water bodies

Source: (Issac & Kandasubramanian, 2021)

## 1.4 100% Bio-degradable plastic alternatives

Even if we succeed in mass producing 100% biodegradable plastic-like substitutes like composite material *barley plastic* (Xu et al., 2021), (UCPH, 2024) that can substitute the current plastic material used for packaging (see figure 5), we still need to address the issue of recovering the spilled plastic that has escaped our waste management efforts and degrades into microplastics thanks to the nature's biological plastic degrading enzymes and Ultraviolet (UV) radiation.


Figure 5: Barley-starch 100% bio-degradable plastic alternative: *Top - A to F images*: Top middle and top right objects are made from barley-starch, which over two months degrade completely, compared to top left traditional plastic packaging and three bottom ones that are current bio-degradable packaging solutions. *Bottom - A & B images*: Barley-starch plastic packaging alternatives can be seen.

Source: (Xu et al., 2021), (UCPH, 2024)

But what if we managed to create a viable bio-engineered PET eating enzyme (FAST-PETase) (H. Lu et al., 2022) using machine learning, that can be used to fully degrade a PET plastic bottle in 24 hours and do so at an industrial scale? Wouldn't that accelerate our recycling efforts? Yes! Would that solve the ecological challenge posed by plastic waste? Not entirely, because not all consumer plastics end up in landfills or are disposed properly (Kaza et al., 2018) and also we still need to collect the plastic waste that is already out there in the nature as mentioned above. What about the illegally disposed waste (Kaza et al., 2018) that ends up in rivers and then in the ocean? Shouldn't we be able to detect the contamination early on to impose fines and inverse the damage while it's still irreversible?

## 1.5 Segregation facilities and working conditions

Another big problem is that the waste segregation facilities like in figure 6 (*next page*), lack personnel, and the nature of the work involved is risky because of the health hazards associated with manual segregation (Pawaskar & Dhanya, 2022).

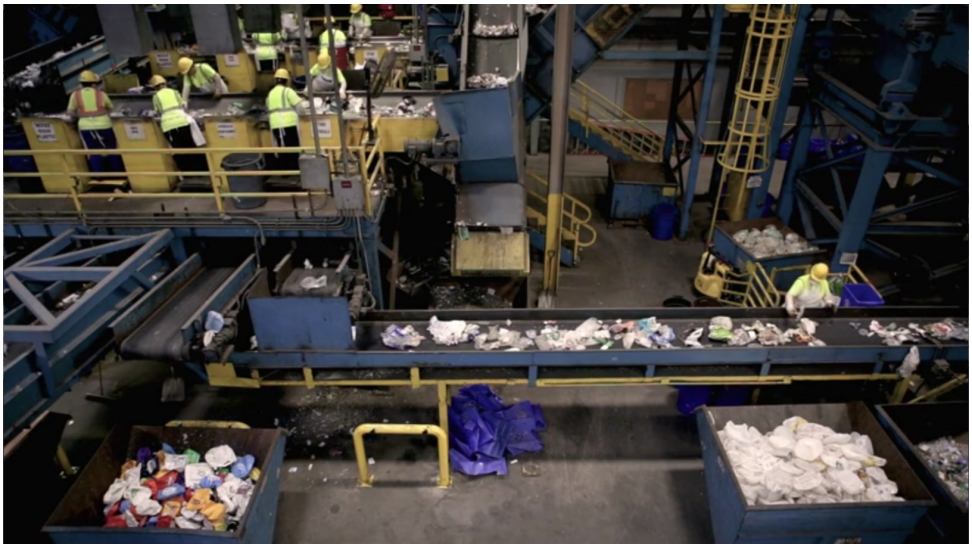


Figure 6: Waste segregation facility in USA

Source: (NPR, 2024)

Amongst others, automated solid waste detection is an important step towards automating solid waste segregation process with the hope of alleviating the health risks associated with it, as mentioned above, and help segregate waste quicker at least as fast as the waste arrives at those facilities with the use of dedicated robotic arms (W. Lu & Chen, 2022).

Common sense says that the sooner we can detect and collect the waste, the better we can tackle the issue of waste getting disintegrated and further harming the environment like in the case of waste ending up in water bodies.

## 1.6 Waste management stages

With the ever-increasing amount of waste generated worldwide, waste management is quite important to make sure urban life remains sustainable, healthy, efficient with a scaling number of inhabitants and that the environmental impact is kept to minimum. Figure 7 depicts the important stages of waste management and even the slightest disruption in one of them, can cause devastating results as seen in the case of covid-19 and is described in (Fang et al., 2023).

## 1.7 Artificial Intelligence applications for waste management

In figure 8, an overview of the various applications for waste management can be seen (Fang et al., 2023). While this is by no means covering all possible applications, it helps to have an idea of the possibilities and prepare the basis for understanding the overarching purpose of this thesis.
Artificial Intelligence (AI) and more specifically computer vision methods like object detection (A. Zhang et al., 2021) can be of considerable assistance to detect waste in various contexts (Majchrowska et al., 2021). Especially as more and more people are using smart devices and web cameras, the big data generated by those devices coupled with the low powered but powerful IOT/ edge computing devices (P. Zhang et al., 2019; White et al., 2020) can potentially make it possible to run computationally


Figure 7: Overview of waste management stages

Source: (Fang et al., 2023)

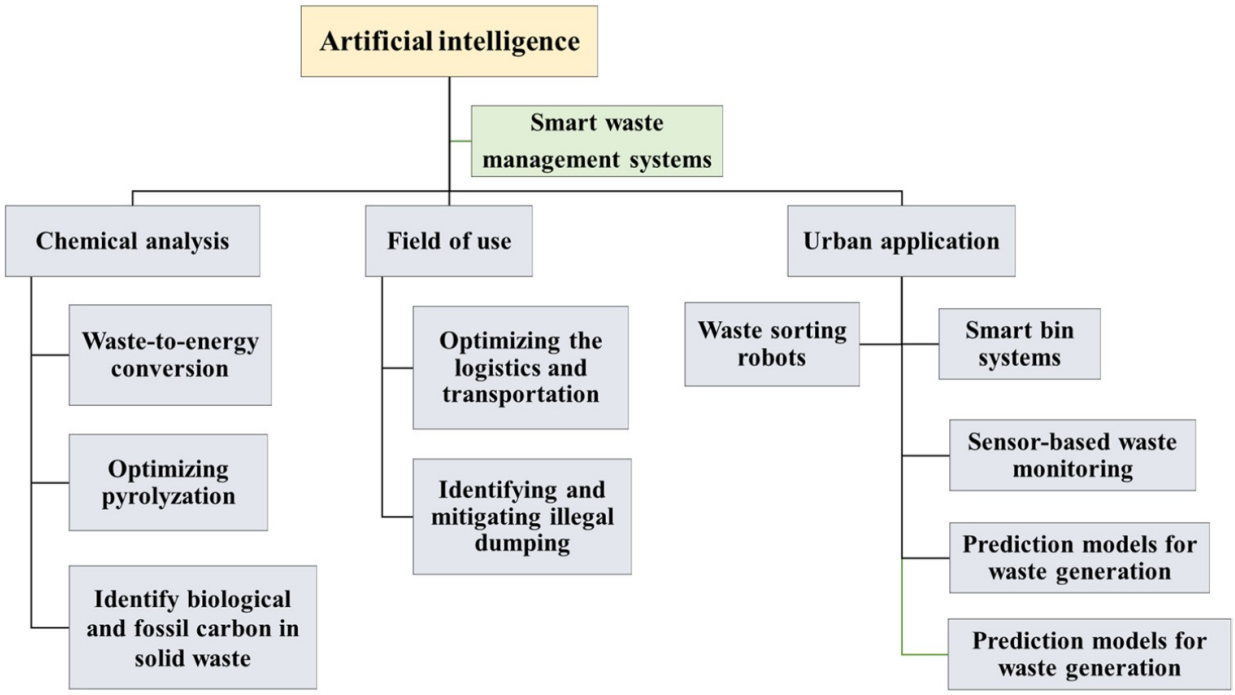



Figure 8: Overview of the AI applications in Waste management

Source: (Fang et al., 2023)

demanding deep learning models on the field and accelerate waste detection process.

Other applications of automated solid waste detection include swarms of robots that can forage for waste (Alfeo et al., 2019) and/or Unmanned Aerial Vehicles (UAVs) that patrol areas to detect solid waste with potentially hazardous (e.g. flammable) or harmful effects for the environment (Kraft et al., 2021).

It is quite important to note that AI has been tangibly helpful so far in waste sorting, waste identifi-

cation, optimizing the travelled distance for vehicles that transport waste, saving costs, time, and thus alleviating some of the biggest pain points of waste logistics (Fang et al., 2023).

Smart garbage bins that perform waste fill-level monitoring with the help of sensors, can tell completely remotely when a bin is about to overflow (Sensoneo, 2024) or the *TrashBot* (CleanRobotics, 2024) which can sort solid waste on the time of disposal and thus eliminating human error and waste inter-contamination or even waste sorting robot *Max-AI* which can perform real-time video detection of e.g. PET Plastic bottles and make sorting facilities safer (Sadako-Technologies, 2024). Representations of these systems can be seen in figure 9 (*next page*).

Last but not least, automated solid waste detection can help in the case of already polluted water bodies that need cleaning and especially shallow waters where solid waste is aggregated more densely and can be more efficiently collected using Autonomous Underwater Vehicles (AUVs) (Valdenegro-Toro, 2019).

## 1.8 Challenges of Artificial Intelligence applications in waste management

The following are challenges that Artificial Intelligence (AI) faces with applications in waste management:

- Lack of specialized models: The AI models that are available have not been custom made for the application at hand and most applications use a pre-trained (on generic data) foundation model which is later fine-tuned to be fitted in the custom dataset of a specific use case.
- Black Box: The models most of the times are seen as black boxes to the non-AI experts or researchers that use them, because explaining how the model arrives at certain outputs or decisions cannot be inferred with certainty. This lack of information leads to the necessity to estimate uncertainty and quantify it by design in computer vision models (Valdenegro-Toro, 2021).
- Lack of Data: There is not sufficient data to train the models as needed, especially in the case of Deep Learning (LeCun et al., 2015). This is a particular challenge, as most public datasets related to waste management which are used for research, are small in size and not diverse enough to cover all use cases. The models cannot generalize easily without enough input data during training.

The above-mentioned challenges are depicted in figure 10.

## 1.9 Purpose of the Thesis

This thesis explores ways to combine multiple public waste datasets into a bigger combined multi-purpose public waste dataset that has images from various background scenes in natural settings. The idea is that the bigger the combined dataset, the more data variability will increase, and the better the results will be overall for multi-class detection of solid waste in various contexts. The combined dataset should also follow a specific label space taxonomy. The labels coming from the individual datasets should be harmonized, such that no two semantically similar labels will exist under different naming or as duplicates.

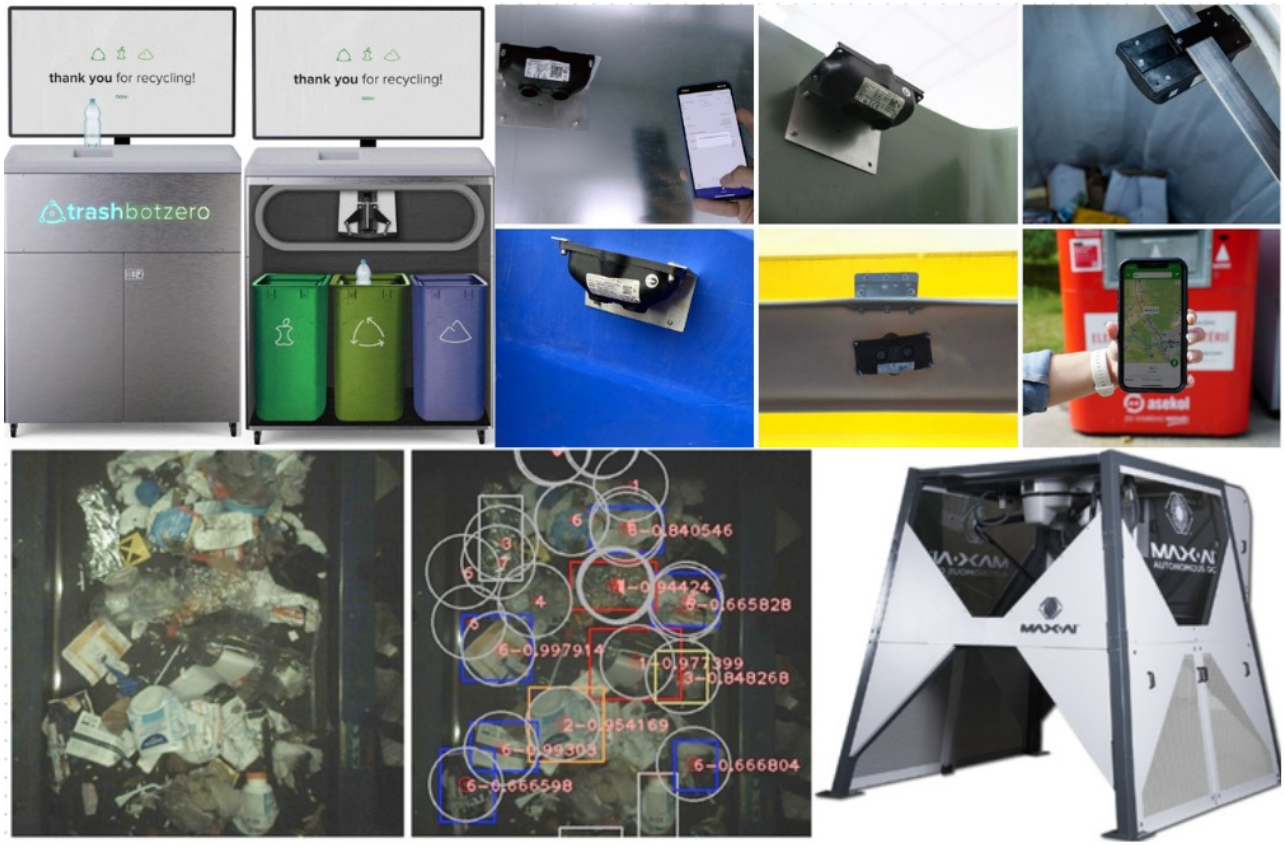



Figure 9: Smart waste recycling bin (upper left corner), smart garbage bin with fill-level sensor (upper right corner), waste sorting robot for segregation facilities (down)

Source: *upper left corner*(CleanRobotics, 2024), *upper right corner*(Sensoneo, 2024), *bottom*(Sadako-Technologies, 2024)

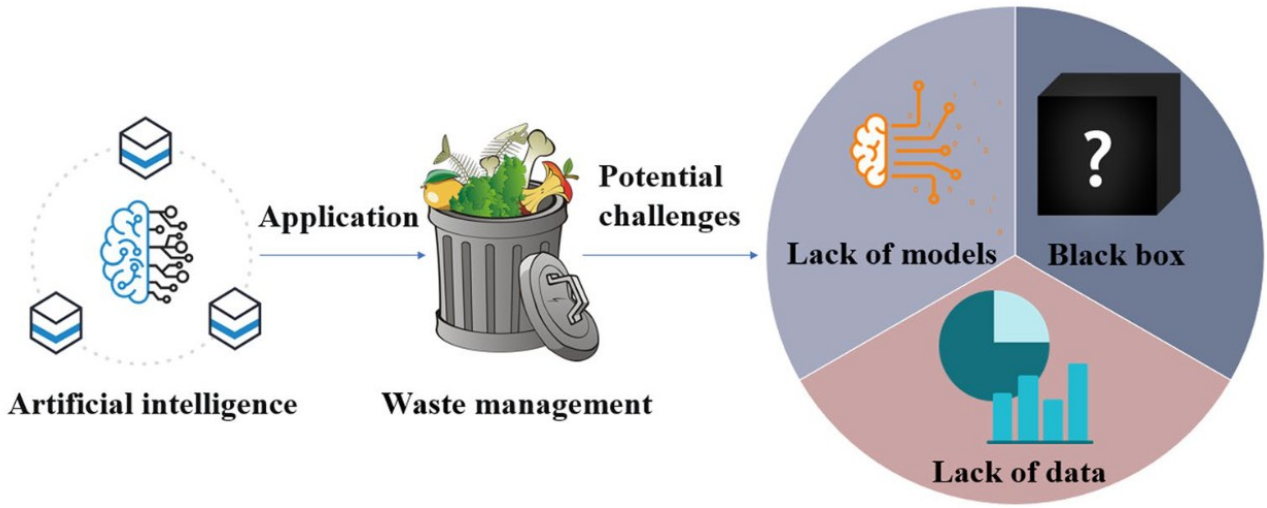



Figure 10: Challenges of Artificial Intelligence applications in waste management

Source: (Fang et al., 2023)

This work will also explore different model architectures from closed-set to open-set computer vision models and generate predictions on the combined multi-purpose public waste dataset to determine how those models qualitatively compare to each other, and whether vision language models can automate parts of the solid waste detection pipeline.

Finally, this work is inspired by the overarching goal of accelerating the improvement of current automated solid waste detection systems using deep learning methods, with the aim to help accelerate waste detection in various contexts and minimizing the need for human involvement.

## 1.10 Research Questions

To summarize, this thesis focuses on problems that can be formulated as research questions as follows:

Q1 What are the challenges of integrating multiple public datasets for solid waste detection?

Q2 How can we impose a label hierarchy to each dataset ensuring the label space is harmonized and perform the merging?

Q3 How can we automate solid waste detection with auto-labelling techniques using vision language models (VLMs)?

Q4. How closed-set perform compared to open-set computer vision models on the combined solid waste dataset?

# 2 Background Literature

In a work done by (Proença & Simões, 2020), TACO public dataset was introduced which has 1500 images and 4784 annotations of solid waste in context (meaning in different scenes/backgrounds), suitable for solid waste object detection in real life use cases.

Another work on aquatic solid waste detection (Hong et al., 2020) introduced TrashCan, a dataset which contains 7212 annotated underwater images of solid waste as well as ROVs, flora & fauna and is suitable for solid waste detection in water bodies.

772 low altitude aerial images have been collected using an Unmanned Aerial Vehicle (UAV) to prepare UAVVaste solid waste dataset in context as described in (Kraft et al., 2021), which can definitely serve as a use case in remote areas where access is very difficult.

An interesting idea came up during the exploratory phase of the literature. If one could somehow combine the different public datasets in context, and manage to carefully synchronise the labels/annotations, the resulted combined dataset would be larger, with enough data to cover all the possible backgrounds/scenes (e.g. oceans, rivers, sand, dirt, pavement, vegetation etc.) found in real life.

The combined multipurpose public waste dataset can then help train a computer vision model, able to detect the objects in question with better precision. Inspiration to achieve this feat was extracted from a work done by (Redmon & Farhadi, 2017), where the authors used *WordTree* a hierarchical model to combine COCO and ImageNet together for joint training.

Different modalities have been used to detect solid waste in context, like using 2D image pictures, 2D aerial images, videos, sonar images, sensor input amongst others, with 2D images being the most common modality, as seen in literature review works like (Lin et al., 2022) and (Shahab et al., 2022).

Once the present work manages to combine multiple datasets, it will explore different computer vision models in order to to find the suitable candidates for the combined dataset. Possible models that this work will explore are: *a)* open-set object detector *GroundingDINO* (Liu et al., 2023), *b)* open-vocabulary object detector *YOLO-World* (Cheng et al., 2024), *c)* image segmentation model *Segment Anything (SAM)* (Kirillov et al., 2023) and *d) YOLO-NAS* object detector (Skalski, 2023) as our baseline.

# 3 Methods

Hereby the methods used for this work are mentioned. The following section describes the data of this work and where it was sourced from. As mentioned before in section 1.9, combining multiple datasets together is the way this work used to obtain a multipurpose combined dataset in context.

## 3.1 Data

The datasets used for this work are all publicly available and suitable for non-commercial use or academic research use. Most datasets contain images and annotation data with bounding boxes and sometimes also segmentation masks. All datasets used for this work are in natural settings (also referred to as *in context* according to the bibliography) and in *COCO* or *.json* format. More information on the datasets that were considered for this work can be found on the table below 1.

| Dataset name | Training Images | Classes | Sub-classes | License Type |
|---|---|---|---|---|
| TACO | 1500 | 28 | 60 | Free license (citation required) |
| UAVVaste | 772 | 1 | N/A | Publicly available (Apache License 2.0) |
| Cigarette Butts | 2000 | 1 | N/A | Non-commercial research license |
| PlastOPol | 2418 | 1 | N/A | Open Access (CC) |

Table 1: Datasets used in this work.

### 3.1.1 Trash Annotations in Context Dataset (TACO)

TACO, is a public dataset of 1500 images in various backgrounds/contexts and 4784 annotations. The annotations contain 28 main categories 60 individual sub-categories (Proença & Simões, 2020). The dataset also contains segmentation points and bounding boxes for each image as seen in figure 11.

In figure 12 some samples of the original TACO dataset can be seen alongside the respective image from the combined dataset.


Figure 11: Original TACO dataset samples

**Original TACO Dataset**


**GT labels:** Cigarette, Other plastic


**GT labels:** Other plastic, Styrofoam piece


**GT labels:** Other plastic, Plastic bottle cap

**Combined Dataset**


**GT labels:** Entity-Non-Living-Waste-Cigarette Butt, Entity-Non-Living-Waste-Other plastic


**GT labels:** Entity-Non-Living-Waste-Other plastic, Entity-Non-Living-Waste-Styrofoam piece


**GT labels:** Entity-Non-Living-Waste-Other plastic, Entity-Non-Living-Waste-Bottle cap-Plastic bottle cap

Figure 12: Original Ground Truth TACO Dataset Image with the respective combined dataset image

### 3.1.2 UAVVaste

UAVVaste is a public dataset of 772 low altitude aerial images taken from an unmanned aerial vehicle (UAV) (Kraft et al., 2021) (as seen in figure 13). It contains 3716 annotations with segmentation and bounding boxes information. This is a one-class dataset with each image containing objects labeled as *rubbish*.

Figure 13: Original UAVVaste dataset samples

### 3.1.3 Cigarette Butts

This non-commercial, annotated dataset consists of 2000 high quality images in different backgrounds or context (*Cigarette Butt Dataset*, n.d.), a sample of which can be seen in figure 14. A part of the dataset is synthetically composed to achieve greater variability of contexts. It is a one-class dataset, and each image contains objects labeled as *cig_butt* under super-category *litter*, with a total number of 2000 annotations for the training set.

### 3.1.4 PlastOPol

PlastOPol is a public dataset of 2418 annotated images in context (samples can be seen in figure 15). Initially It was not possible to find the annotations file. After tracing the annotations later in the thesis pipeline, I decided to keep this set of images as a testing set. Given its significant number of annotated images in context, PlastOPol is definitely a great option to consider as well, for creating a larger combined dataset.

Figure 14: Original Cigarette Butts dataset samples

Figure 15: Original PlastOPol dataset samples

### 3.1.5 Other Datasets

Initially, more datasets were considered for this work. However, some of them did not have annotations' information available to use in a known format (COCO or Pascal VOC) *(e.g. TrashNet, Drinking waste, OpenLitterMap)*, others did not contain images in natural settings which was crucial to achieve real-world object detections in the wild *(e.g. TrashNet, MJU-Waste)*. Moreover, some datasets were not publicly available and needed special permission to use for research purposes or were closed behind a pay wall *(e.g. WaBaDa, Domestic Garbage)*. For all these reasons this work focused on the above four datasets. For any future work that would like to reference them, this work also includes a table with information about the datasets that were considered.

From table 3, TrashCan 1.0 has underwater images which are not suitable for the purposes of this work (which is focused on above water natural settings), but would be suitable for waste detection using Autonomous Underwater Vehicles (AUVs). More image datasets related to solid waste or waste

| Dataset name | Training Images | Classes | Sub-classes | License Type |
|---|---|---|---|---|
| TrashCan 1.0 | 7212 | 3 | N/A | Free for academic research use |
| MJU-Waste v1.0 | 2475 | 1 | N/A | Publicly available (Apache License 2.0) |
| TrashNet | 2527 | 6 | N/A | Publicly available (MIT license 2.0) |
| Drinking waste | 4810 | 4 | N/A | Public domain (CCO 1.0) |
| WaBaDa | 4000 | 7 | N/A | Permission needed to obtain annotations |
| OpenLitterMap | >100k | 11 | 187 | Open Access (CC) |
| Domestic Garbage | >9000 | 4 | N/A | Paid license for images >250 |

Table 3: Datasets not used in this work but were considered.

in general can be found in a comprehensive review compiled from (Mikołajczyk, 2024).

## 3.2 Models

As described in section 2, and for the purpose of this thesis we used various models to generate results over the combined multipurpose solid-waste dataset.

### 3.2.1 Closed-Set vs. Open-Set Object Detection Paradigms

When the categories/labels are pre-defined, like in the case of COCO object detection benchmark, where there are 80 object categories, then this is called a *closed-set* object detection, also known as *fixed-vocabulary* object detection. There is also the so-called *open-set* object detection where apart from existing categories, the model can also detect arbitrary classes using text prompts, also known as *open-vocabulary* object detection (OVD). E.g. An *open-set* object detector pre-trained on COCO (80 classes) and prompted with 10 novel classes, will attempt to detect objects within a class space of $10 + 80 = 90$ classes in total.

### 3.2.2 Closed-Set Object Detectors

**YOLO-NAS** According to (Skalski, 2023) YOLO-NAS outperformed models like YOLOv7 & YOLOv8 being around 0.5 mAP more accurate and 10-20% faster with latency of just 2.36 milliseconds. The architecture of YOLO-NAS with the quantization blocks and selective quantization converts activations, biases, and weights from floats to integers with small precision loss with the added value of enhanced efficiency. With each neural architecture search (NAS) it enhances object detection performance, efficiency and makes it more robust. In figure 16, the architecture of the model can be seen.
YOLO-NAS has the following key architecture components that make it able to stand out from the YOLO family:

- AutoNAC which is an optimization algorithm that helps determine the most suitable architecture for the task at hand. AutoNAC uses a hybrid quantization method that selectively quantizes specific layers of the neural network to optimize accuracy and latency trade-offs while also maintaining the overall performance.

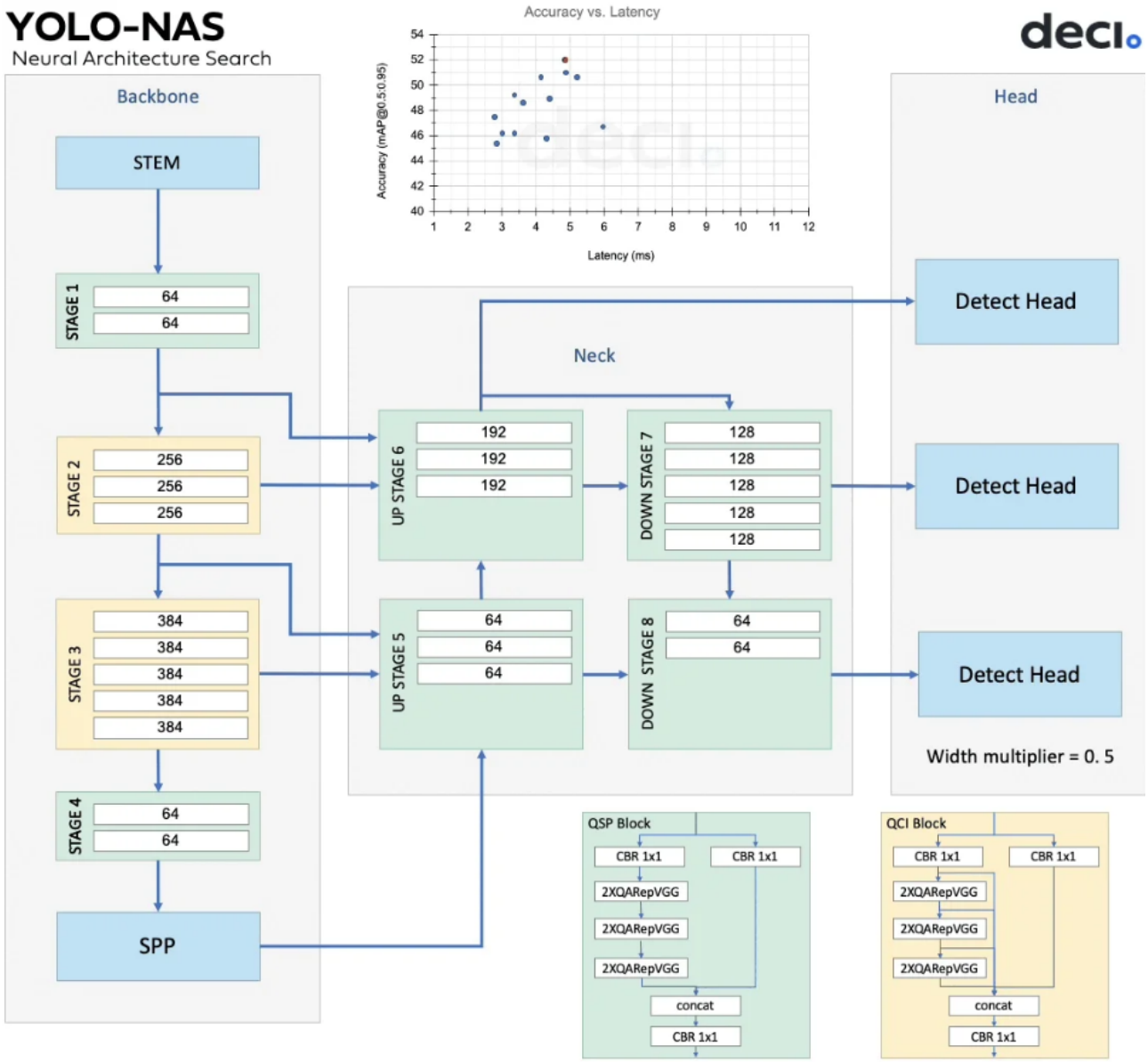



Figure 16: YOLO-NAS model architecture.

Source: (Deci.ai, 2023)

- QSP (*Quantization Specific Parameters*) and QCI *Quantization Centric Initialization*) blocks are the building blocks that allow the model to remain modular, flexible and be able to adjust its parameters by shrinking the model without losing information. QSP is the one that performs the quantization and thus making the model smaller and faster to train, while QCI is making sure the necessary information is kept intact. AutoNAC orchestrates the process of which layers to quantize and how to arrange the building blocks (QSP & QCI) together to build the final model.

YOLO-NAS is a foundation model that utilizes pre-training on Roboflow100, COCO, Objects 365 datasets, which means that is faster, cheaper and less time consuming to train overall.

### 3.2.3 Pre-trained Open-Set Object Detectors

**GroundingDINO** The authors of (Liu et al., 2023) introduced the zero-shot & open-set detector groundingDINO which achieved high performance on COCO dataset, gives flexibility for researchers to integrate it with other models (*e.g. Stable Diffusion*), it's readily accessible to researchers and did bring a lot of progress in the field of open-vocabulary detection (OVD). In figure 17, the architecture of the model can be seen.

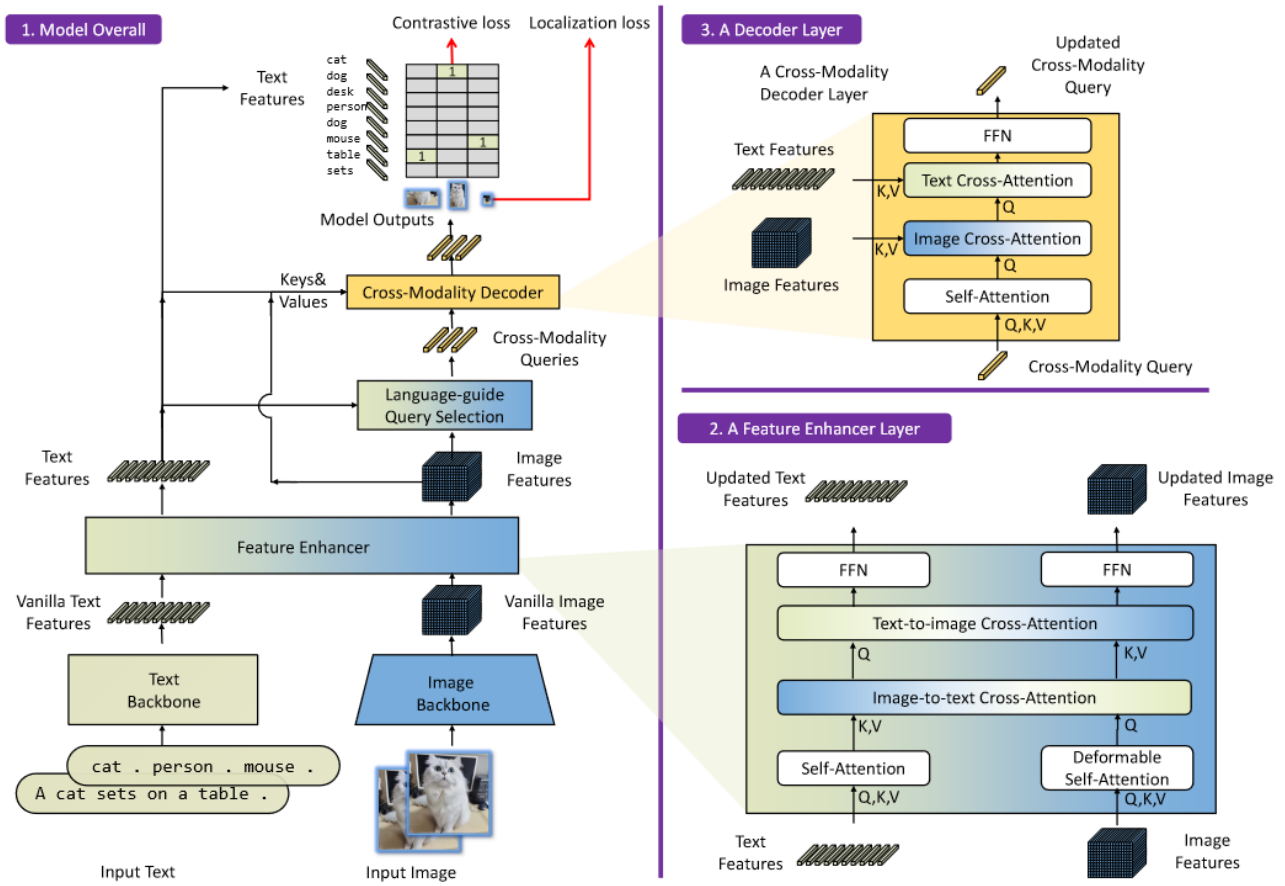



Figure 17: Grounding DINO model architecture.

Source: (Liu et al., 2023)

GroundingDINO architecture consists of the following key components:

- Image backbone (e.g. Swin-Transformer), which extracts visual information features from images.
- Text backbone (e.g. BERT), which extracts textual information features from text.
- Feature enhancer layer, which performs the cross-modality feature fusion of the two previous components using the self-attention mechanism in two steps:
  - Image-to-text-Cross-Attention: Which looks at which parts of the textual input are relevant for each specific visual feature.
  - Text-to-image-Cross-Attention: Which looks at which parts of the visual input are relevant for each textual feature.

- Language-guided Query selection: This component selects the most relevant visual features for a given textual input/prompt/query.
- Cross-modality Decoder: This component's role is to select the relevant features from both visual and textual features using the self-attention, which results in an updated cross-modality query.
- Loss computation: This last part, follows DETR-like works and computes the loss by combining the L1 loss or Absolute Error Loss (which contrasts the predicted bounding boxes with ground truth bounding boxes and compares their coordinates/location) with GIoU loss (Generalized Intersection over Union-which measures the overlap between the predicted and ground truth bounding boxes and completely ignores non-overlapping bounding boxes).

**YOLO-World** In the work of (Cheng et al., 2024), the authors introduced a new version of the YOLO series which improves upon the previous works by adding the possibility of using open-vocabulary as textual input and be able to detect novel categories not included in the closed-set siblings of the YOLO series. Moreover, zero-shot object detector YOLO-World enables the fusion of text embeddings with vision image embeddings. The novel component that the authors introduced called *Re-parameterizable Vision-Language Path Aggregation Network (RepVL-PAN)* can match the input textual information with the regions of interest that are obtained from the input image, enabling a multi-modal approach to object detection. This approach can open-up opportunities to use more diverse datasets and expanded datasets to help accelerate object detection development. In figure 18, the architecture of the model can be seen.

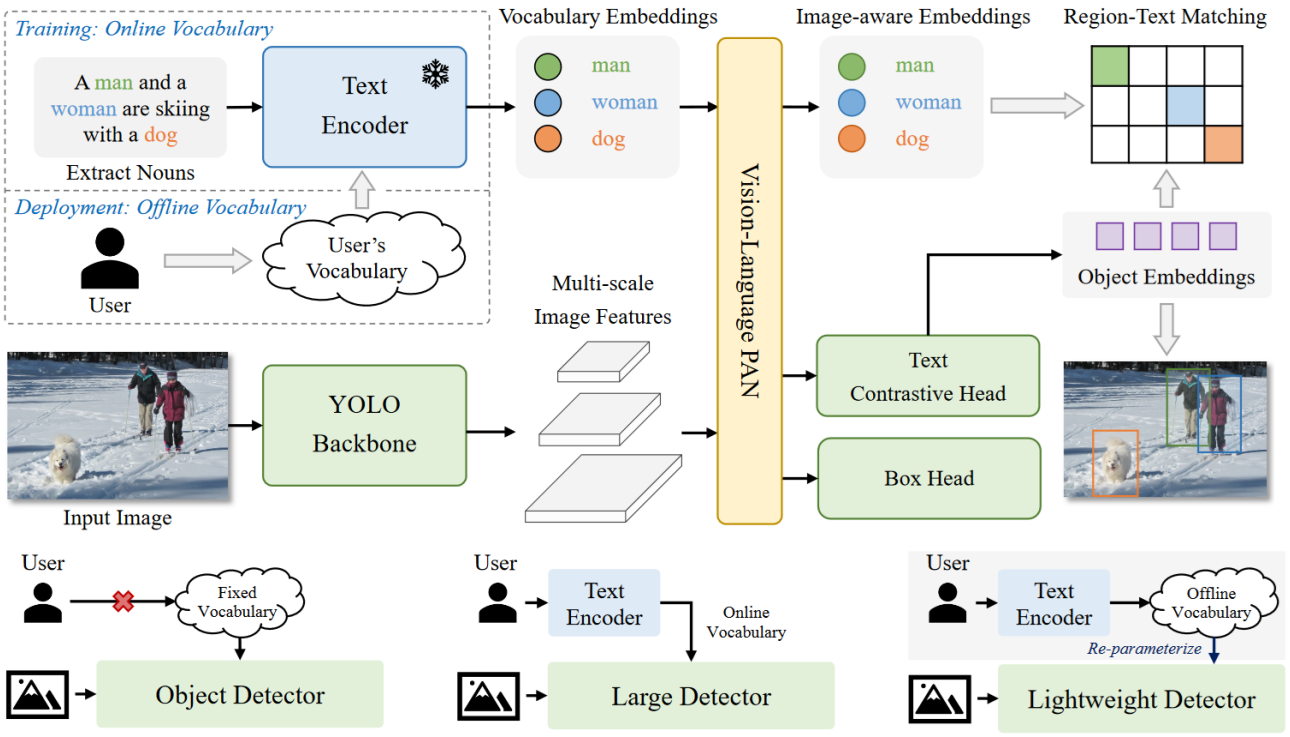



Figure 18: YOLO-World model architecture.

Source: (Cheng et al., 2024)

Regarding the architecture of the YOLO-World model, it has three key parts:

- YOLO detector that extracts multi-scale features from the input image.
- CLIP text encoder that encodes the text into text-embeddings.
- and RepVL-PAN custom network mentioned also above that performs multi-level cross-modality fusion between image features and text embeddings.

Reasons for YOLO-World's real-time detection speed:

- The backbone that this model uses compared to other open-vocabulary object detectors (e.g. GroundingDINO which uses computationally heavier DINO backbone) is a faster and lighter Convolutional Neural Network (CNN) based on Darknet architecture, which highly accounts for its inference speed increase.
- Another reason for the inference speed increase is the prompt then detect paradigm. Instead of encoding the user's prompt real-time during inference, YOLO-World uses CLIP to convert text input (generated while prompting the model) into offline vocabulary embeddings which are basically cached and re-used, thus circumvent the need for real-time text encoding.

Overall, YOLO-World can make the open vocabulary object detection faster, cheaper, and widely available. While maintaining roughly the same accuracy compared to its predecessors, it is 20 times faster than other architectures in the same category. Last but not least, YOLO-World makes it easier to deploy and can be integrated with other architectures for further enhancing its object detection capabilities (e.g. EfficientSAM).

### 3.2.4 Instance Segmentation

**Segment Anything Model (SAM)** In this work (Kirillov et al., 2023), the authors introduced a zero-shot segmentation model that enables the segmentation of unseen objects from an image and is based on a transformer architecture. Moreover, this work introduced the largest (at the time of publication) segmentation dataset *SA-1B* and a prompt-able segmentation task that enables the input of textual information alongside the image to produce the segmentation masks. In figure 19, the architecture of the model can be seen.

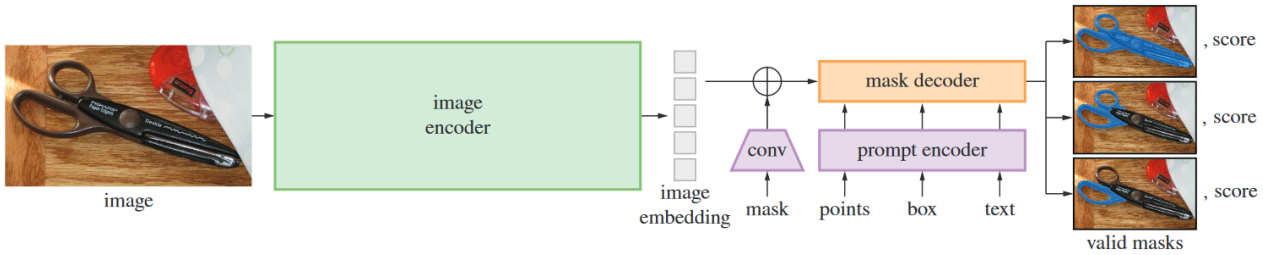



Figure 19: Segment Anything model architecture.

Source: (Kirillov et al., 2023)

The Segment Anything Model architecture consists of three main components:

- Image encoder that takes an image as input and then outputs its embeddings. This is the most computationally heavy of the three.
- Prompt encoder which can take a set of points, a bounding box, another mask or simply input text and outputs a prompt embedding.
- Mask decoder combines the outputs of the two above-mentioned encoders and predicts the segmentation masks.

What is standing out with this foundation model release, is the opportunities it opens in terms of generating accurate segmentation masks in a zero-shot fashion, the efficiency in terms of how fast it can segment input data even when running on a CPU, and the huge SA-1B dataset with over 11 million licensed images and more than 1 billion masks.

# 4 Experimental Setup

## 4.1 Tools and Technologies

The students own laptop was used for writing the thesis and when it comes to training the models on the combined public waste dataset. Remote access was granted by the supervisor (16 core processor, Core i9-12900K, 64 GB RAM, 2x RTX 3090 24GB each) to test inference models, play around with different architectures, and write small Python scripts to call the inference API module of Roboflow. Support in the form of supervision took place once a week, every two weeks or even once a month depending on the complexity of the work parts (WPs) at hand.

## 4.2 Combined Dataset

The combined dataset consists of TACO, UAVVaste and Cigarette Butts individual datasets. The number of total images amounts to 4272 solid waste images in context and the number of total annotations amounts to 10500.

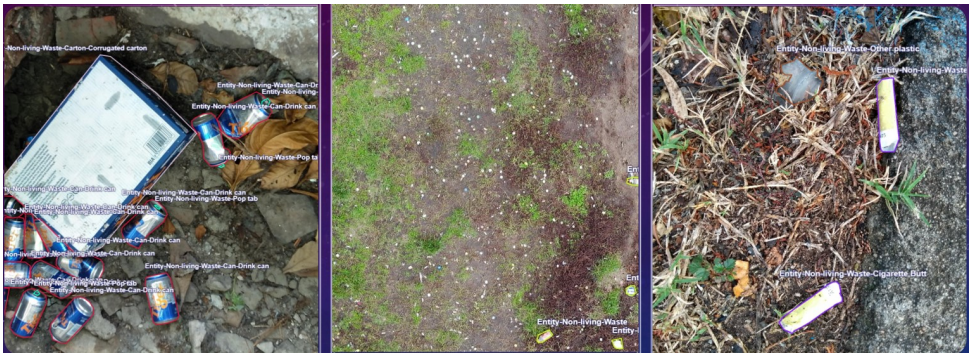
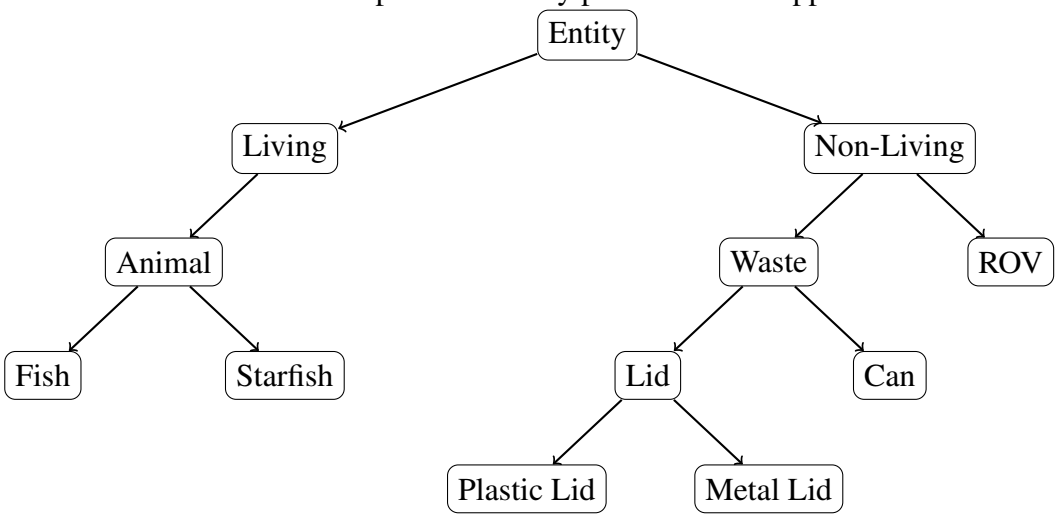


Figure 20: Combined multipurpose public waste dataset

## 4.3 Label space taxonomy

The label space is based on a logical taxonomy that goes from a high level to low levels of abstraction as shown below. For the complete taxonomy please refer to Appendices section.



### 4.3.1 Label Mapping

As a result of the heterogeneous nature of the label spaces of the different datasets, after importing the datasets, one needs to harmonize the labels so that they have a common label taxonomy, meaning that the label space used is aligned across all datasets. To achieve this, a *.json* file with key-value pairs was composed. This file manually maps each label item for a given original dataset label with a new semantically similar label that adheres to the desired label taxonomy (mentioned in 4.3) and to make sure there is no duplication across all used datasets as shown below. The depicted listing below is the entire mapping file used for both datumaro framework and the custom harmonisation script.

```
"Trash": "Entity/Non-living/Waste",
"Rubbish": "Entity/Non-living/Waste",
"rubbish": "Entity/Non-living/Waste",
"litter":  "Entity/Non-living/Waste",
"Waste": "Entity/Non-living/Waste",
"cig_butt": "Entity/Non-living/Waste/Cigarette Butt",
"Cigarette": "Entity/Non-living/Waste/Cigarette Butt",
"trash_can": "Entity/Non-living/Waste/Can",
"trash_rope": "Entity/Non-living/Waste/Rope & strings",
"trash_cup": "Entity/Non-living/Waste/Cup",
"Unlabeled litter": "Entity/Non-living/Waste/Unknown",
"trash_clothing": "Entity/Non-living/Waste/Clothing",
"trash_pipe": "Entity/Non-living/Waste/Pipe",
"trash_bottle" : "Entity/Non-living/Waste/Bottle",
"trash_bag": "Entity/Non-living/Waste/Bag",
"trash_container": "Entity/Non-living/Waste/Plastic container",
"trash_unknown_instance": "Entity/Non-living/Waste/Unknown",
"trash_branch": "Entity/Non-living/Waste/Branch",
"trash_wreckage": "Entity/Non-living/Waste/Wreckage",
"trash_tarp": "Entity/Non-living/Waste/Tarp",
"trash_net": "Entity/Non-living/Waste/Net",
"Non-waste": "Entity/Non-living/Non-waste",
"Cigarette Butt": "Entity/Non-living/Waste/Cigarette Butt",
"Can": "Entity/Non-living/Waste/Can",
"Rope & strings": "Entity/Non-living/Waste/Rope & strings",
"rov": "Entity/Non-living/Non-waste/ROV",
"plant": "Entity/Living/Plant",
"animal_fish": "Entity/Living/Animal/Fish",
"animal_starfish": "Entity/Living/Animal/Starfish",
"animal_shells": "Entity/Living/Animal/Shells",
"animal_crab": "Entity/Living/Animal/Crab",
"animal_eel": "Entity/Living/Animal/Eel",
"animal_etc": "Entity/Living/Animal/Etc",
"Clothing": "Entity/Non-living/Waste/Clothing",
"Pipe": "Entity/Non-living/Waste/Pipe",
"Bottle": "Entity/Non-living/Waste/Bottle",
"Bag": "Entity/Non-living/Waste/Bag",
```

```
"Paper bag": "Entity/Non-living/Waste/Bag/Paper bag",
"Plastic bag": "Entity/Non-living/Waste/Bag/Plastic bag",
"Plastic wrapper": "Entity/Non-living/Waste/Plastic wrapper",
"trash_snack_wrapper": "Entity/Non-living/Waste/Plastic wrapper/
  Other plastic wrapper",
"Cup": "Entity/Non-living/Waste/Cup",
"Plastic container": "Entity/Non-living/Waste/Plastic container",
"Unknown": "Entity/Non-living/Waste/Unknown",
"Other plastic": "Entity/Non-living/Waste/Other plastic",
"Branch": "Entity/Non-living/Waste/Branch",
"Wreckage": "Entity/Non-living/Waste/Wreckage",
"Tarp": "Entity/Non-living/Waste/Tarp",
"Net": "Entity/Non-living/Waste/Net",
"Styrofoam piece": "Entity/Non-living/Waste/Styrofoam piece",
"Straw": "Entity/Non-living/Waste/Straw",
"Squeezable tube": "Entity/Non-living/Waste/Squeezable tube",
"Shoe": "Entity/Non-living/Waste/Shoe",
"Scrap metal": "Entity/Non-living/Waste/Scrap metal",
"Pop tab": "Entity/Non-living/Waste/Pop tab",
"Plastic utensils": "Entity/Non-living/Waste/Plastic utensils",
"Plastic glooves": "Entity/Non-living/Waste/Plastic glooves",
"Paper": "Entity/Non-living/Waste/Paper",
"Lid": "Entity/Non-living/Waste/Lid",
"Glass jar": "Entity/Non-living/Waste/Glass jar",
"Food waste": "Entity/Non-living/Waste/Food waste",
"Carton": "Entity/Non-living/Waste/Carton",
"Broken glass": "Entity/Non-living/Waste/Broken glass",
"Bottle cap": "Entity/Non-living/Waste/Bottle cap",
"Blister pack": "Entity/Non-living/Waste/Blister pack",
"Battery": "Entity/Non-living/Waste/Battery",
"Aluminium foil": "Entity/Non-living/Waste/Aluminium foil",
"Paper straw": "Entity/Non-living/Waste/Straw/Paper straw",
"Plastic straw": "Entity/Non-living/Waste/Straw/Plastic straw",
"Other plastic container": "Entity/Non-living/Waste/Plastic
  container/Other plastic container",
"Foam food container": "Entity/Non-living/Waste/Plastic container/
  Foam food container",
"Disposable food container": "Entity/Non-living/Waste/Plastic
  container/Disposable food container",
"Tupperware": "Entity/Non-living/Waste/Plastic container/Tupperware
  ",
"Spread tub": "Entity/Non-living/Waste/Plastic container/Spread tub
  ",
"Other plastic wrapper": "Entity/Non-living/Waste/Plastic wrapper/
  Other plastic wrapper",
"Crisp packet": "Entity/Non-living/Waste/Plastic wrapper/Crisp
  packet",
```

```
"Six pack rings": "Entity/Non-living/Waste/Plastic wrapper/Six pack rings",
"Plastic film": "Entity/Non-living/Waste/Plastic wrapper/Plastic film",
"Polypropylene bag": "Entity/Non-living/Waste/Bag/Plastic bag/Polypropylene bag",
"Single-use carrier bag": "Entity/Non-living/Waste/Bag/Plastic bag/Single-use carrier bag",
"Garbage bag": "Entity/Non-living/Waste/Bag/Plastic bag/Garbage bag",
"Plastified paper bag": "Entity/Non-living/Waste/Bag/Paper bag/Plastified paper bag",
"Magazine paper": "Entity/Non-living/Waste/Paper/Magazine paper",
"Wrapping paper": "Entity/Non-living/Waste/Paper/Wrapping paper",
"Tissues": "Entity/Non-living/Waste/Paper/Tissues",
"Normal paper": "Entity/Non-living/Waste/Paper/Normal paper",
"Metal lid": "Entity/Non-living/Waste/Lid/Metal lid",
"Plastic lid": "Entity/Non-living/Waste/Lid/Plastic lid",
"Other plastic cup": "Entity/Non-living/Waste/Cup/Other plastic cup",
"Glass cup": "Entity/Non-living/Waste/Cup/Glass cup",
"Foam cup": "Entity/Non-living/Waste/Cup/Foam cup",
"Disposable plastic cup": "Entity/Non-living/Waste/Cup/Disposable plastic cup",
"Paper cup": "Entity/Non-living/Waste/Cup/Paper cup",
"Other carton": "Entity/Non-living/Waste/Carton/Other carton",
"Toilet tube": "Entity/Non-living/Waste/Carton/Toilet tube",
"Pizza box": "Entity/Non-living/Waste/Carton/Pizza box",
"Meal carton": "Entity/Non-living/Waste/Carton/Meal carton",
"Egg carton": "Entity/Non-living/Waste/Carton/Egg carton",
"Drink carton": "Entity/Non-living/Waste/Carton/Drink carton",
"Corrugated carton": "Entity/Non-living/Waste/Carton/Corrugated carton",
"Food Can": "Entity/Non-living/Waste/Can/Food Can",
"Aerosol": "Entity/Non-living/Waste/Can/Aerosol",
"Drink can": "Entity/Non-living/Waste/Can/Drink can",
"Metal bottle cap": "Entity/Non-living/Waste/Bottle cap/Metal bottle cap",
"Plastic bottle cap": "Entity/Non-living/Waste/Bottle cap/Plastic bottle cap",
"Other plastic bottle": "Entity/Non-living/Waste/Bottle/Other plastic bottle",
"Glass bottle": "Entity/Non-living/Waste/Bottle/Glass bottle",
"Clear plastic bottle": "Entity/Non-living/Waste/Bottle/Clear plastic bottle",
"Carded blister pack": "Entity/Non-living/Waste/Blister pack/Carded blister pack",
```

```
110 "Aluminium blister pack": "Entity/Non-living/Waste/Blister pack/
      Aluminium blister pack"
```

## 4.4 Datumaro framework

For harmonizing the labels of datasets, imposing a common label space taxonomy amongst the datasets, merging the datasets as well as for visualizing the resulting merged dataset's labels, datumaro was used (*openvinotoolkit/datumaro*, 2024) developed by OpenVINO. Datumaro framework, accommodates using mapping files like the one shown above, to impose the label taxonomy to any dataset and harmonize the labels. After imposing the same label taxonomy to every dataset and achieving harmonisation, the next step was to merge the datasets together using datumaro. The entire workflow can be seen in figure 21.

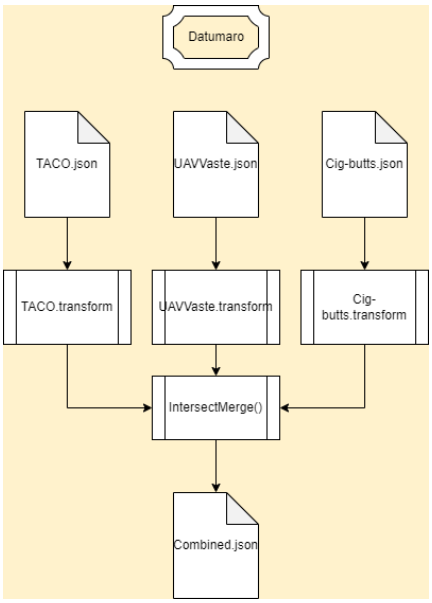



Figure 21: Datumaro framework workflow for merging datasets

## 4.5 Custom harmonisation script

Apart from datumaro harmonisation method described above, a custom script was developed to harmonise the datasets in a way that follows the label taxonomy and uses the mapping file. For each dataset, a separate *.json* file is created that adheres to the imposed label taxonomy as can be seen in the workflow of figure 22.

## 4.6 Custom merging script

Apart from datumaro merge discussed above, a custom merging script was developed to merge the datasets' *.json* files together. The script parses each *.json* file and generates a single, merged *.json* file that consists of all the info, licences, categories, images and annotations information from the harmonized datasets used as input, while ensuring that during merging unique identifiers are renewed and information stays intact, as can be seen from the workflow figure 23.

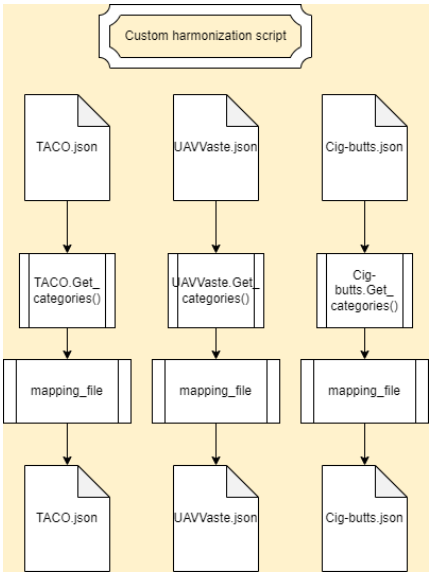



Figure 22: Custom script workflow for harmonizing datasets

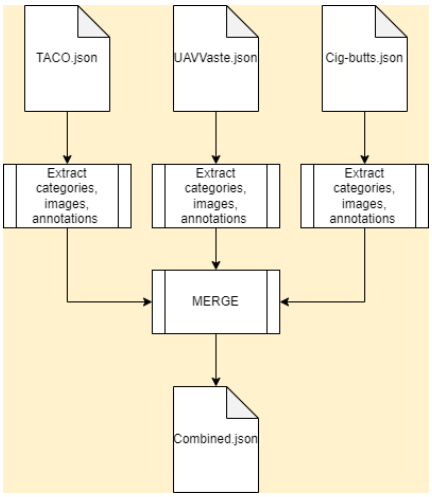



Figure 23: Custom script workflow for merging datasets

## 4.7 Hugging Face platform

Hugging face (Face, 2024) platform was used to perform parameter search and test predictions for YOLO-World model and get acceptable results faster for our test inferences with the samples from the combined dataset. Using the same tool, the final test results were created.
The landing page of the particular YOLO-World space (Grove, 2024), is easy to use, and the workflow is as follows:

1. Click on *Drop an image or Click to Upload* on the top right side of the page to input an image.
2. On the section below, enter the classes that you would like the detector to detect separated by a comma in the form of a prompt.
3. Input the parameter values:
   (a) Maximum number of boxes *(referring to the predicted bounding boxes)*.

(b) Score threshold *(referring to the confidence score threshold)*.

(c) NMS threshold *(referring to the threshold for Non-Maximum Suppression)*.

4. Submit.
5. Output image with predictions appears on the upper right side of the screen with the option to download.
6. Download the image.

This above process/workflow was repeated for each input image. Figure 24 shows the user interface of the space by *stevengrove* in Hugging Face platform.

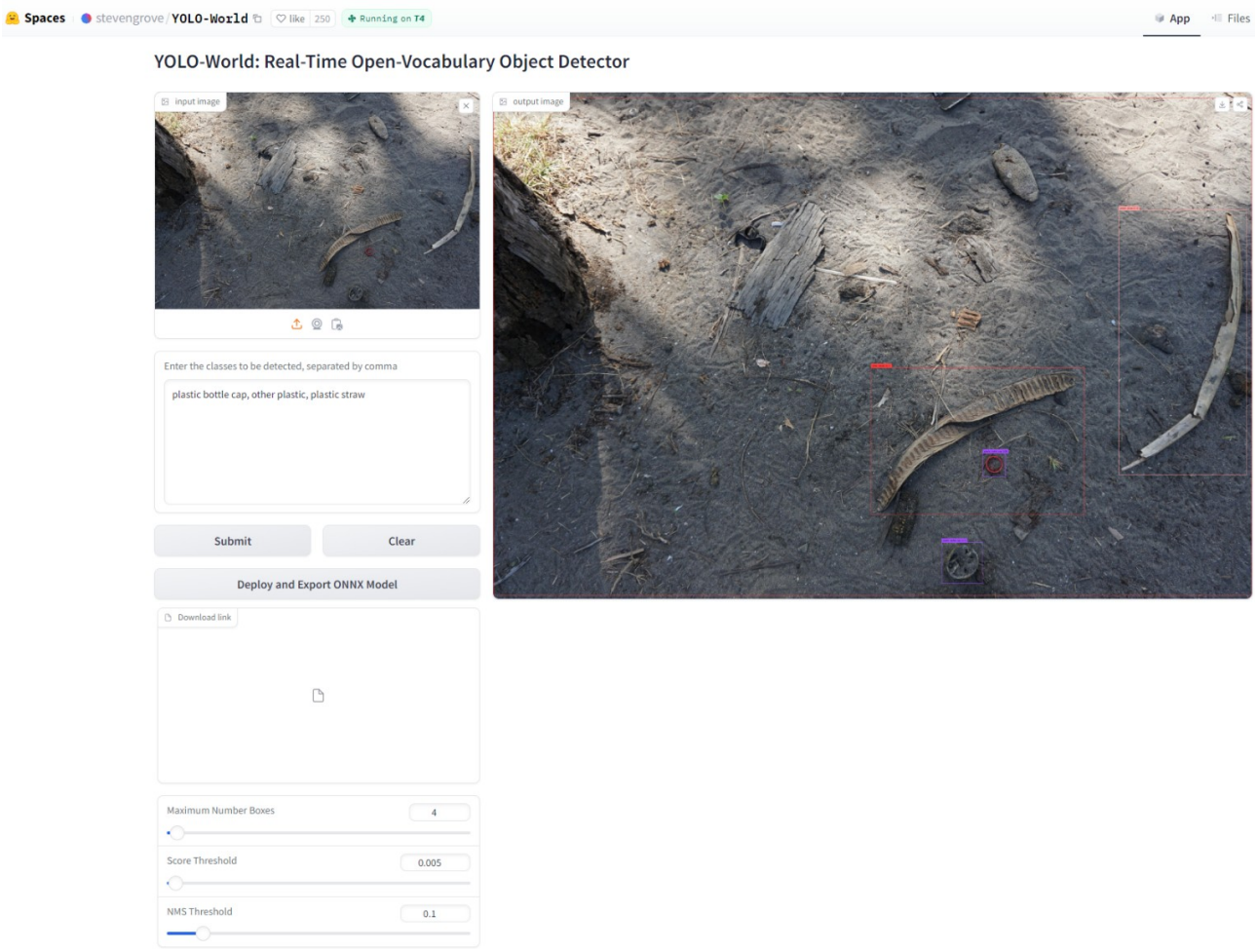



Figure 24: YOLO-World Hugging Face Space by Steven Grove

Source: (Grove, 2024)

To summarize, the inputs that YOLO-World demo application expects are the following:

- the image,
- the classes of the objects to be predicted,
- the input parameters values (Max. number of boxes, Conf. score, NMS).

## 4.8 Meta's Segment Anything Model (SAM) - demo

SAM demo (*Segment Anything | Meta AI*, 2023) was used to obtain the results for Segment Anything model, particularly the function that segments all objects in the input image. Using the official website of the demo, I clicked on *Try the demo* and accepted the terms and conditions. This opened the demo service in which I uploaded a test image. The demo service then automatically extracts an embedding from the uploaded image, after which, I used the *Everything* tool, which splits the image into grids, localizes all the objects that apply to each grid and outputs the predicted segmentation masks resulting in segmenting all the objects automatically. Having all the object masks for an image, the results can be contrasted with the ground truth image to make a qualitative analysis of the resulted segmented objects that can be identified as solid waste.

The above workflow was repeated for all the test images that were used for the qualitative analysis which can be seen in the section 5.4.2. Figure 25 depicts the user interface of Segment Anything Model demo.

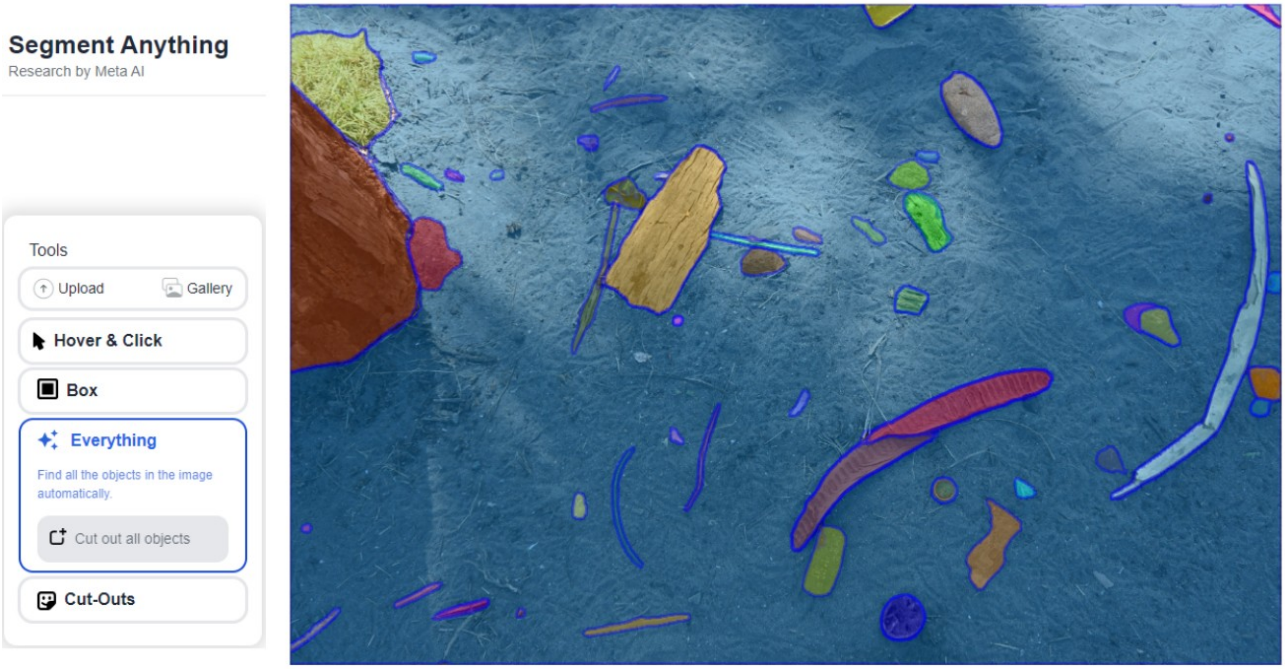



Figure 25: Segment Anything Model - demo page

Source: (*Segment Anything | Meta AI*, 2023)

Summarizing the workflow, the inputs that SAM demo application expects are:

- the input image,
- the selection and usage of any of the following available tools:
  - Hover & Click - Which allows to hover over and click any object and create multi-mask which can be further examined.
  - Box - To manually draw a bounding box around an object.

- Everything - Localizes and draws segmentation masks for all objects in the image automatically.
- Cut-outs - Shows the cut-out objects .

## 4.9 Roboflow platform

Roboflow platform (Roboflow, 2024) served the purpose of streamlining the computer vision pipeline such as a web-based user interface to store the combined dataset, evaluate the correctness of the labels, train the YOLO-NAS model, obtain graphs and quantitative data, and to test the auto-label functionality for GroundingDINO architecture.

### 4.9.1 Chosen License

In order to be able to use Roboflow, platform registration was required, in which I have opted for the *Research plan*, which compared to the *Public - Free version* gives 22 more training credits to train your own custom or provided Roboflow model, 15.000 more inference API calls and the same amount of Auto-label credits to label data using models like *GroundingDINO*.

### 4.9.2 Training YOLO-NAS

Within Roboflow platform, I created a new project. Inside that project I uploaded the combined dataset images alongside the *.json* file containing the annotations. The classes were updated according to the uploaded annotation file. To train a new model based on the combined dataset, I clicked *Create New Version* under Versions section. During the creation of the new version, it is possible to apply pre-processing and augmentation steps, determine how the dataset will be split between *Train, Valid, Test* sets and inspect the dataset's images once more. For the split I went for the following proportions *(Train: 80%, Valid: 10%, Test: 10%)*.
To start the training, I clicked on *Train with Roboflow* button and selected the desired model architecture. From the available models that could be selected with the academic license, I chose the *YOLO-NAS-S* architecture (where *S* stands for the small variant of the model). The training took a few hours, and a notification was sent to my academic email once the training job was done. More information on how to train YOLO-NAS model on Roboflow can be found in (*Launch: Train and Deploy YOLO-NAS Models on Roboflow*, 2024).
Once the training is done, within the platform we can see the obtained metrics for mAP, Precision and Recall. The training graphs can be visualized by clicking the *More metrics* and selecting the *Training Graphs* option. Under *More metrics* average precision per class can be seen as well for both validation and test sets. Moreover, through *Visualize* option, *Test* images can be inspected alongside their ground truth and predicted labels.

### 4.9.3 GroundingDINO - Auto-label

To obtain the predictions from GroundigDINO, the Auto-label functionality from Roboflow was used. In order to obtain a resulted zero-shot object detection a specific workflow was used that follows:

1. Create a new project *(See figure 26)*.
2. Upload Data:

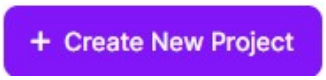



Figure 26: Roboflow - Creating a new project

Source: (Roboflow, 2024)

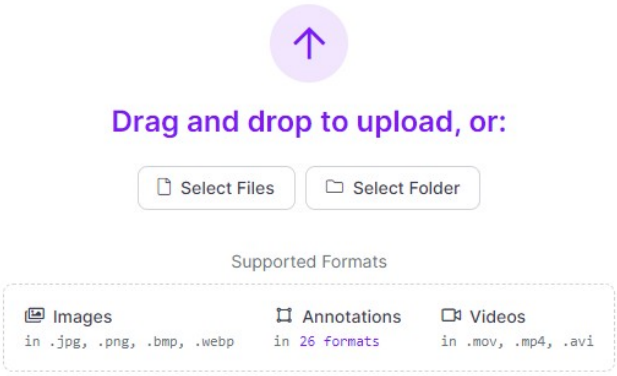



Figure 27: Roboflow - Upload image

Source: (Roboflow, 2024)

(a) Upload image on Roboflow *(See figure 27)*.

(b) Click on Save and Continue *(See figure 28)*.

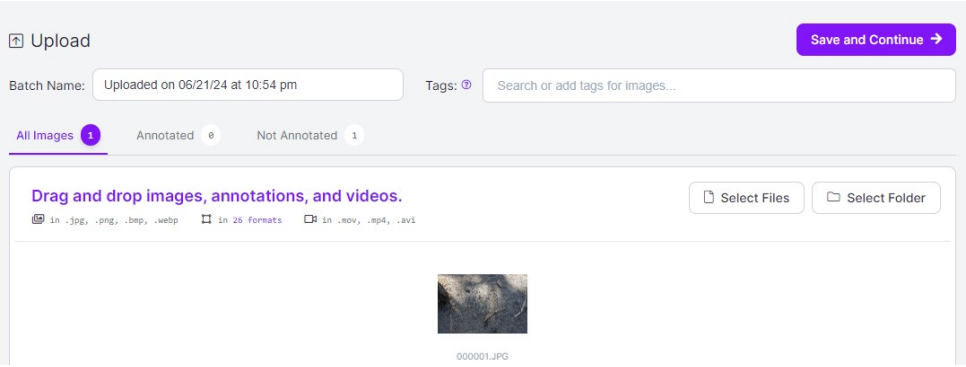



Figure 28: Roboflow - Save and Continue

Source: (Roboflow, 2024)

3. Annotate *(Using Auto-label functionality)*:

(a) Click on Start Auto-label *(See figure 29)*.

(b) The classes view opens, in which you can add classes/delete classes *(See figure 30)*.

(c) Add prompt for each class (optional) *(If no prompt is specified, class name is used instead)*.

(d) Make sure to choose the correct model for auto-label from the drop-down menu *(See figure 31)*.

(e) Click on Generate Test Results *(See figure 32)*.

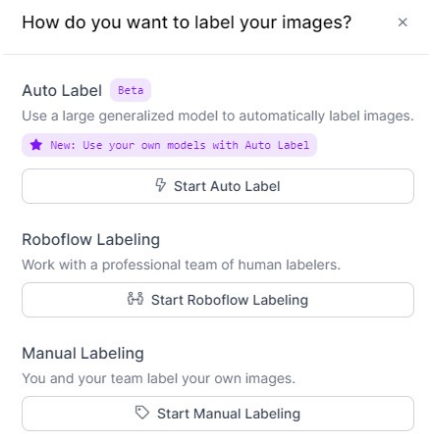



Figure 29: Roboflow - Start Auto-label

Source: (Roboflow, 2024)

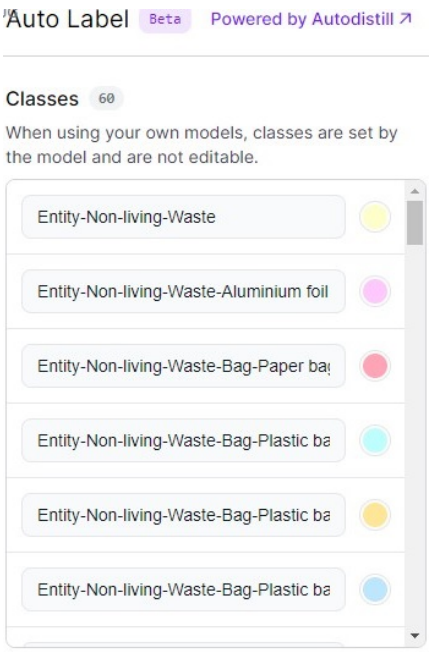



Figure 30: Roboflow - Edit classes in Auto Label tool.

Source: (Roboflow, 2024)

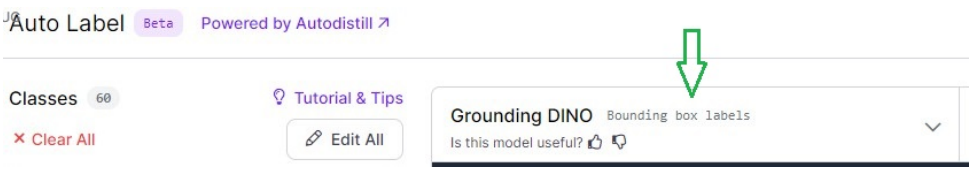



Figure 31: Roboflow - Choosing the model architecture for auto-labelling from drop-down list.

Source: (Roboflow, 2024)

(f) Determine the confidence threshold for each added class *(available range: 10 - 95%) (See*

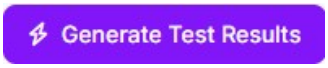



Figure 32: Roboflow - Generate Test Results - Auto Label.

Source: (Roboflow, 2024)

*figure 33).*

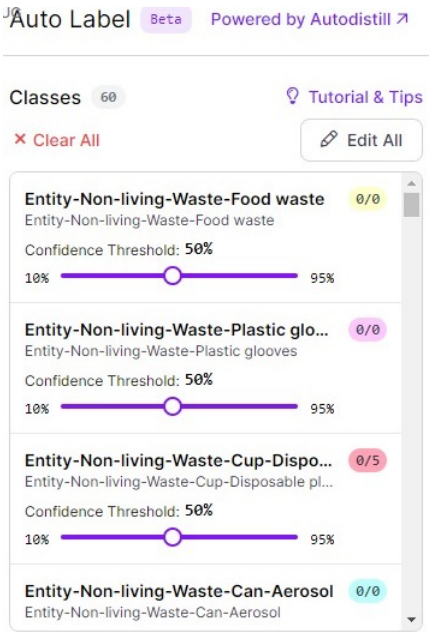



Figure 33: Roboflow - Determine the confidence threshold for each class - Auto Label.

Source: (Roboflow, 2024)

(g) Click on Auto-label with this model *(this creates an annotation job with predicted labels) (See figure 34).*

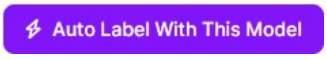



Figure 34: Roboflow - Auto-label with this model.

Source: (Roboflow, 2024)

(h) Review auto-annotated image by clicking on the created job *(See figure 35).*

(i) Edit labels where applicable with Annotation Editor by clicking on the object at hand *(See figure 36).*

(j) Approve or Reject annotated image *(See figure 37).*

(k) Click Add approved to Dataset *(See figure 38).*

To obtain the results seen in section 5.3.1, the above workflow was repeated for every image.

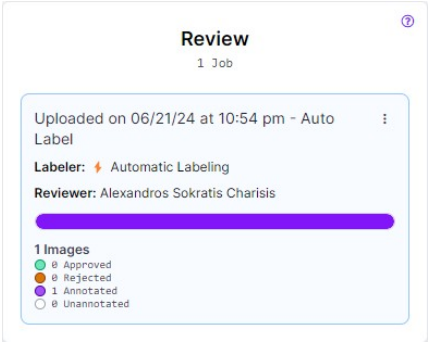



Figure 35: Roboflow - Review auto-annotated image.

Source: (Roboflow, 2024)

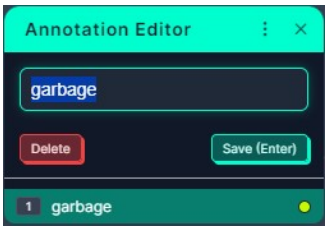



Figure 36: Roboflow - Edit labels where applicable with Annotation Editor.

Source: (Roboflow, 2024)

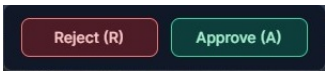



Figure 37: Roboflow - Approve or Reject annotated image.

Source: (Roboflow, 2024)

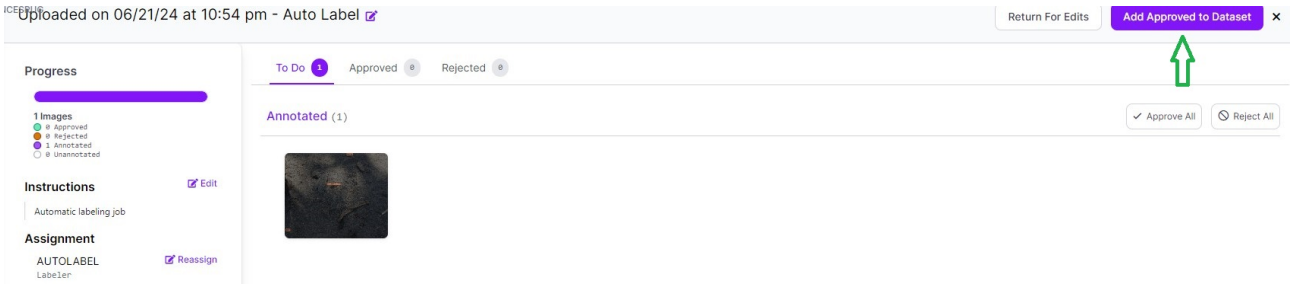



Figure 38: Roboflow - Add approved (annotated image) to the Dataset.

Source: (Roboflow, 2024)

# 5 Results

## 5.1 What are the challenges of integrating multiple public datasets for solid waste detection?

Merging multiple datasets together, has challenges:

- *Harmonizing the label space of each original dataset and imposing a unified label hierarchy to ensure uniformity of the target label space*.
  - Mapping the semantically similar labels for all used public datasets before merging them together is important to avoid duplicates. To achieve this, a mapping file was used, and not only the semantically similar labels were tackled, but also a label space taxonomy was imposed. The mapping file can be seen in section 4.3.1.
- *Merging the datasets without introducing issues and validating the results of the combined dataset*.
  - To make sure no issues were introduced in the combined dataset after merging, thorough visual inspection using *Roboflow platform* was done by looking into samples from the uploaded combined multipurpose public waste dataset, to make sure that the annotations remained intact and no issues were introduced.

## 5.2 How can we impose a label hierarchy to each dataset ensuring the label space is harmonized and perform the merging?

### 5.2.1 Datumaro

Imposing a label hierarchy to ensure a harmonized label space for the combined dataset using datumaro produced the desired results. However, the merging of the datasets using datumaro produced a problematic combined dataset. To be more precise, after visualizing the combined dataset using Roboflow dataset inspection tool, there was an introduction of erroneous annotations of objects not even part of the original set of images before merging. This erroneous introduction of non-existent objects to images led to the decision to abandon this method altogether.

### 5.2.2 Custom scripts

Imposing a label hierarchy and merging the datasets produced the desired results with the developed custom scripts. To validate the result the combined dataset was uploaded and inspected thoroughly using the Roboflow platform.

## 5.3 How can we automate solid waste detection with auto-labelling techniques using language-vision models?

### 5.3.1 Grounding DINO

Using this language-vision model to generate the predictions for the selected sample of test images, we obtain the following results which can be seen in figures 39, 40, 41 and 42. As input/detection prompt, the same classes as the ground truth labels were used. The confidence threshold is a range of

confidence threshold percentages set as an input parameter for the Grounding DINO model.

Diving into the results in more detail, in figure 39, we observe the following:

- Top image: In this image we can see that only the *plastic straw* was correctly predicted, all other objects were missed. There were no misclassifications.
- Middle image: In this case, the model managed to predict correctly *the drink cans, other carton, clear plastic bottle*, but missed the *plastic film, the single-use carrier bag* and the *plastic bottle cap*.
- Bottom image: This scene is a very difficult one, as there are many cluttered objects on the image, with a lot of them not solid waste objects, which can confuse even the human eye. In this case the model managed to localize two *cigarette butts* correctly but also three false positives that admittedly look a lot like cigarettes/cigarette butts to the human eye. However, it also missed the rest of the two classes compared to the ground truth.

For figure 40 we can observe the following:

- Top image: In this image we can see that all *drink cans and the corrugated carton* were predicted correctly. In this case we have obtained perfect results, as there are only two classes that the model is requested to predict.
- Middle image: In this case, the model managed to predict correctly one instance of *other plastic* out of the three instances seen in the ground truth image, as well as the *styrofoam piece*.
- Bottom image: In this case, the model predicted the two instances of *styrofoam piece* and missed the *unknown waste* instance compared to the ground truth image.

For figure 41 we can observe the following:

- Top image: All solid waste objects were predicted correctly in this instance.
- Middle image: Here, the model missed the *cigarette butt* instance which is a very small object to detect on this scale and correctly predicted the *other plastic* object.
- Bottom image: Almost all solid waste objects were predicted correctly in this instance apart from the top left object for which two bounding boxes were over-predicted/drawn.

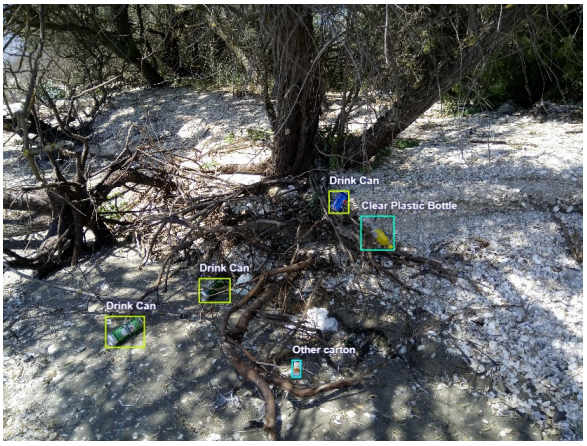
Lastly, for figure 42 all objects were predicted correctly.

**Ground Truth** | **Grounding DINO**


**GT labels:** Plastic bottle cap, Other plastic, Plastic straw


**Prompt:** Plastic bottle cap, Other plastic, Plastic straw — **Confidence Threshold:** 50%


**GT labels:** Single-use carrier bag, Plastic bottle cap, Clear plastic bottle, Drink Can, Other carton, Plastic film




**Prompt:** Single-use carrier bag, Plastic bottle cap, Clear plastic bottle, Drink Can, Other carton, Plastic film — **Confidence Threshold:** 10-60%


**GT labels:** Cigarette Butt, Other plastic, Waste/Unknown


**Prompt:** Cigarette Butt, Other plastic, Unknown waste — **Confidence Threshold:** 15-60%

Figure 39: Grounding DINO results - part 1 - Where **prompt** is the detection prompt in which distinct classes were used as input for GroundingDINO model, and **confidence threshold** is the range of confidence threshold percentages set across all distinct classes used in the detection prompt

**Ground Truth** | **Grounding DINO**


**GT labels:** Drink can, Corrugated carton, Pop tab


**Prompt:** Drink can, Corrugated carton, Pop tab — **Confidence Threshold:** 10-50%


**GT labels:** Other plastic, Styrofoam piece


**Prompt:** Other plastic, Styrofoam piece — **Confidence Threshold:** 30-45%


**GT labels:** Styrofoam piece, Unknown waste


**Prompt:** Styrofoam piece, Unknown waste — **Confidence Threshold:** 35-55%

Figure 40: Grounding DINO results - part 2 - Where **prompt** is the detection prompt in which distinct classes were used as input for GroundingDINO model, and **confidence threshold** is the range of confidence threshold percentages set across all distinct classes used in the detection prompt

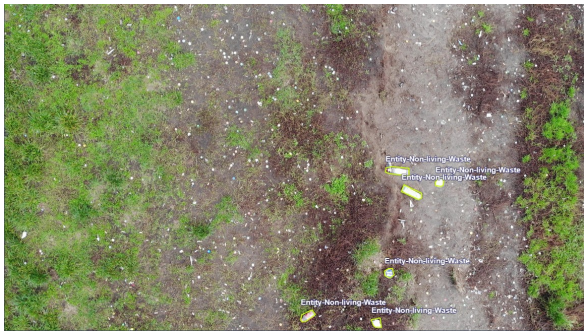
**Ground Truth** | **Grounding DINO**


**GT labels:** Other plastic, Plastic bottle cap

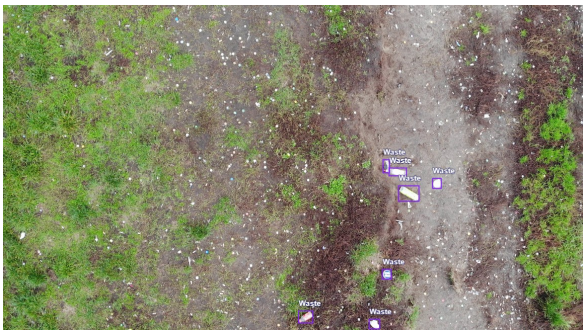

**Prompt:** Other plastic, Plastic bottle cap — **Confidence Threshold:** 40-50%


**GT labels:** Cigarette butt, Other plastic


**Prompt:** Cigarette butt, Other plastic — **Confidence Threshold:** 30-40%




**GT labels:** Waste



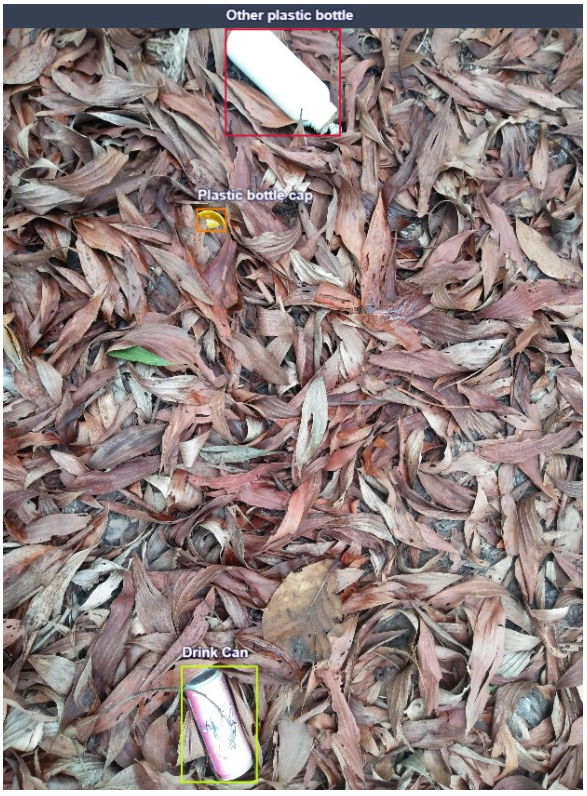

**Prompt:** Waste — **Confidence Threshold:** 20%

Figure 41: Grounding DINO results - part 3 - Where **prompt** is the detection prompt in which distinct classes were used as input for GroundingDINO model, and **confidence threshold** is the range of confidence threshold percentages set across all distinct classes used in the detection prompt

**Ground Truth**


**GT labels:** Plastic bottle cap, Other plastic bottle, Drink can

**Grounding DINO**




**Prompt:** Plastic bottle cap, Other plastic bottle, Drink can — **Confidence Threshold:** 10-50%

Figure 42: Grounding DINO results - part 4 - Where **prompt** is the detection prompt in which distinct classes were used as input for GroundingDINO model, and **confidence threshold** is the range of confidence threshold percentages set across all distinct classes used in the detection prompt

### 5.3.2 YOLO-World

Inference results on test set images from this language-vision model can be seen in figures 43, 44, 45 and 46.
For the first set of images as seen in figure 43, a few remarks:

- The model misclassified the *drink can* as *other plastic bottle* and missed the *plastic bottle cap* altogether. On a positive note, the upper *other plastic bottle* solid waste object was correctly predicted.

For the second set of images as seen in figure 44, a few observations:

- Top image: The model managed to correctly predict the small solid waste object *plastic bottle cap* while also produced three misclassifications, two of which belonged to non-solid waste objects present on the scene and one big bounding box of *plastic straw* which is an over-prediction of a set of objects as a single object. This is something we observed quite a lot happening with *YOLO-World* model and could only be mitigated partially by experimenting with *confidence score and Non-Maximum Suppression* parameters.
- Middle image: In this case, the model managed to predict correctly two out of three *drink can* instances, misclassify two objects as *plastic bottle cap* and missed all the other solid waste objects.
- Bottom image: Apart from a *cigarette butt* instance that the model predicted correctly (top left corner), all other predictions are either false positives or misclassified objects.

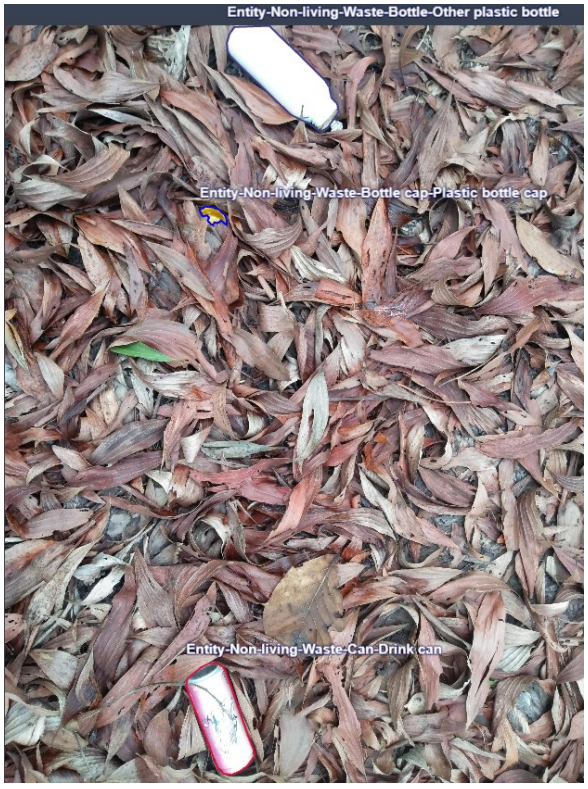
For the third set of images as seen in figure 45, the following can be observed:

- Top image: The model managed to correctly predict almost all *drink can* instances (apart from one), and the *corrugated carton*, but produced one false positive prediction for *pop tab* on the top right corner of the image.
- Middle image: Here, the model predicted correctly the *styrofoam piece*, misclassified one object as *styrofoam piece* and missed out the *other plastic* instances on the top of the image.
- Bottom image: In this case, we can see two objects correctly classified as *styrofoam piece*, but one of them has an over-drawn bounding box that includes both instances. The *unknown waste* instance was left out completely.

Lastly, for the fourth set of images as seen in figure 46, the following remarks can be made:

- Top image: The *plastic bottle cap* instance was predicted correctly, we have two false positives of tree leaves being predicted as solid waste and classified as *other plastic*, while the actual *other plastic* object has an overdrawn bounding box over it that includes the tree leaf.
- Middle image: Here, the model predicted correctly *other plastic* instance and has predicted incorrectly a big bounding box for almost all the scene as *other plastic*.
- Bottom image: In this case, three out of 6 solid waste objects present in the image were correctly predicted.

**Ground Truth**




**GT labels:** Plastic bottle cap, Other plastic bottle, Drink can

**YOLO-World**

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.3 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 3

Figure 43: YOLO-World results - part 1 - Where **Prompt** refers to the input classes that the YOLO-World model is prompted to detect, **Score threshold** is confidence score threshold and **NMS threshold** is the threshold for Non-Maximum Suppression

**Ground Truth**


**GT labels:** Plastic bottle cap, Other plastic, Plastic straw

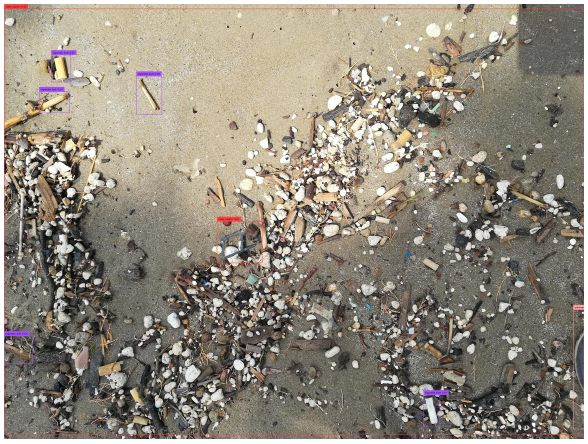

**GT labels:** Single-use carrier bag, Plastic bottle cap, Clear plastic bottle, Drink Can, Other carton, Plastic film


**GT labels:** Cigarette Butt, Other plastic, Waste/Unknown

**YOLO-World**

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 4

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.04 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 8



**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 17

Figure 44: YOLO-World results - part 2 - Where **Prompt** refers to the input classes that the YOLO-World model is prompted to detect, **Score threshold** is confidence score threshold and **NMS threshold** is the threshold for Non-Maximum Suppression

**Ground Truth**


**GT labels:** Drink can, Corrugated carton, Pop tab

**YOLO-World**

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 21


**GT labels:** Other plastic, Styrofoam piece

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.01 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 3


**GT labels:** Styrofoam piece, Unknown waste

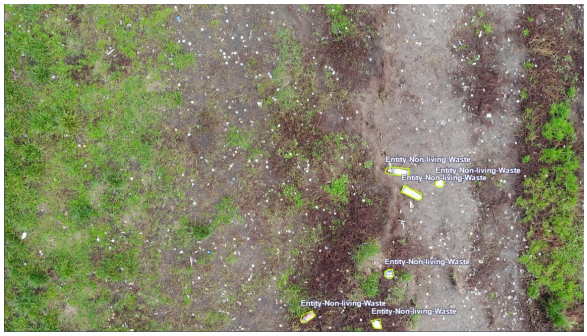
**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.3 — **Maximum number of boxes:** 3

Figure 45: YOLO-World results - part 3 - Where **Prompt** refers to the input classes that the YOLO-World model is prompted to detect, **Score threshold** is confidence score threshold and **NMS threshold** is the threshold for Non-Maximum Suppression

**Ground Truth**


**GT labels:** Other plastic, Plastic bottle cap


**GT labels:** Cigarette butt, Other plastic

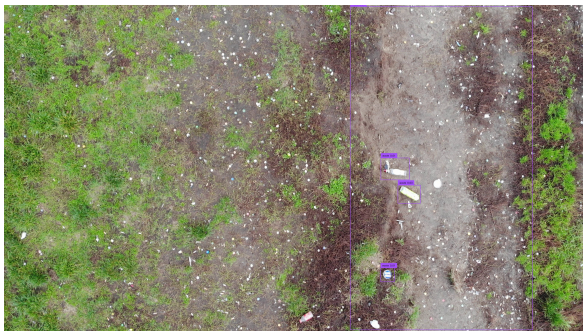



**GT labels:** Waste

**YOLO-World**

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.009 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 4

**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 2



**Prompt:** Same as GT labels — **Confidence Threshold:** 0.005 — **NMS threshold:** 0.1 — **Maximum number of boxes:** 5

Figure 46: YOLO-World results - part 4 - Where **Prompt** refers to the input classes that the YOLO-World model is prompted to detect, **Score threshold** is confidence score threshold and **NMS threshold** is the threshold for Non-Maximum Suppression

## 5.4 How closed-set perform compared to open-set computer vision models on the combined solid waste dataset?

### 5.4.1 YOLO-NAS

The quantitative data with details for YOLO-NAS closed-set detector trained for this work can be found in table 12:

| model | task | mAP | Precision | Recall | license |
|---|---|---|---|---|---|
| YOLO-NAS Small | object-detection | 11.5% | 18% | 6.7% | Apache License 2.0 |

Table 12: Quantitative model comparison of the closed-set object detectors for combined multipurpose solid-waste dataset

The training graphs for YOLO-NAS can be seen in figure 47.

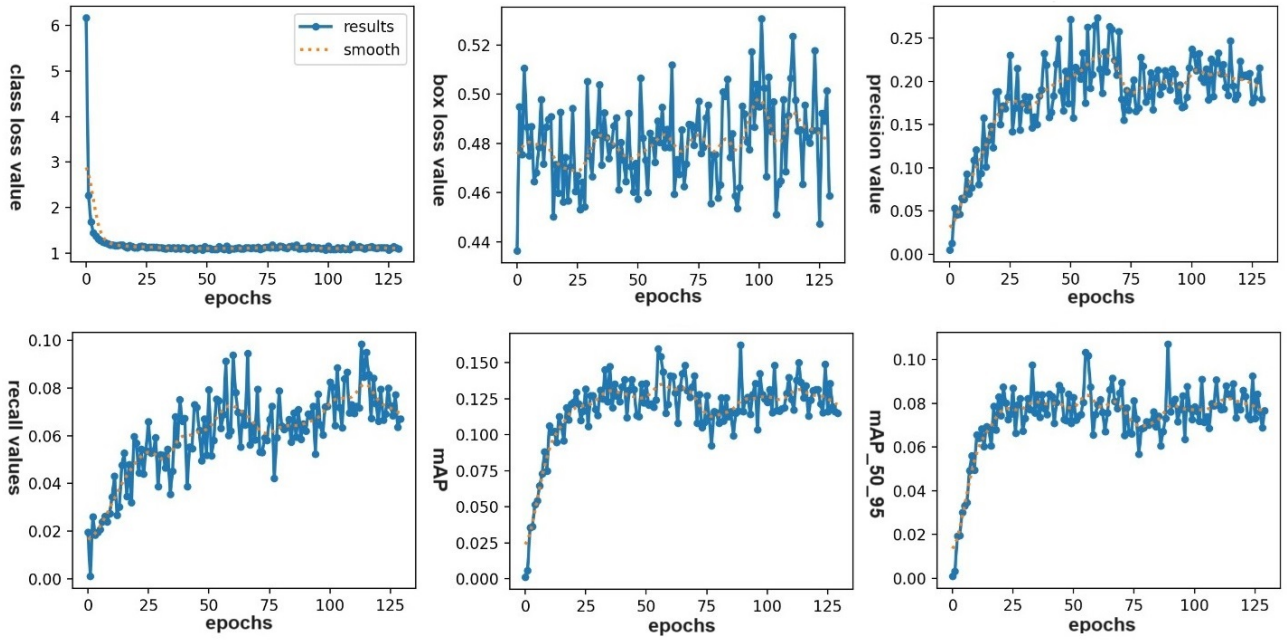



Figure 47: YOLO-NAS training graphs.

This model's results can be seen in figures 48, 49, 50 and 51.

A few remarks can be made for the predictions of YOLO-NAS results:
In the image of figure 48 only the *drink can* was correctly predicted. In figure 49 only the *plastic bottle cap* of the top image has been correctly predicted. Regarding 50, top image had three instances of *drink can* objects correctly classified and middle image had one instance of *styrofoam piece* correctly classified. Lastly, in figure 51, top image had *other plastic* instance correctly predicted and classified. All other predicted objects from the obtained results were either false positives or misclassifications.

**Ground Truth**


**GT labels:** Plastic bottle cap, Other plastic bottle, Drink can

**YOLO-NAS**

**Minimum confidence:** 26% — **Maximum overlap:** 10%

Figure 48: YOLO-NAS results - part 1 - Where **Minimum confidence** is the minimum accepted confidence score & **Maximum overlap** Maximum accepted overlap between two bounding boxes

**Ground Truth**


**GT labels:** Plastic bottle cap, Other plastic, Plastic straw

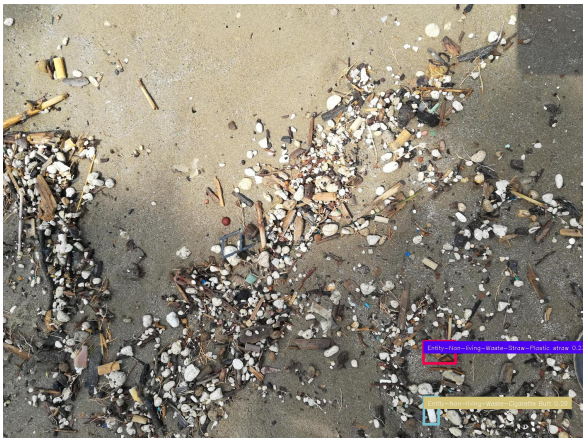

**GT labels:** Single-use carrier bag, Plastic bottle cap, Clear plastic bottle, Drink Can, Other carton, Plastic film


**GT labels:** Cigarette Butt, Other plastic, Waste/Unknown

**YOLO-NAS**

**Minimum confidence:** 30% — **Maximum overlap:** 10%

**Minimum confidence:** 38% — **Maximum overlap:** 10%



**Minimum confidence:** 23% — **Maximum overlap:** 10%

Figure 49: YOLO-NAS results - part 2 -Where **Minimum confidence** is the minimum accepted confidence score & **Maximum overlap** Maximum accepted overlap between two bounding boxes

**Ground Truth** **YOLO-NAS**

**GT labels:** Drink can, Corrugated carton, Pop tab

**Minimum confidence:** 40% — **Maximum overlap:** 30%

**GT labels:** Other plastic, Styrofoam piece

**Minimum confidence:** 35% — **Maximum overlap:** 30%

**GT labels:** Styrofoam piece, Unknown waste

**Minimum confidence:** 40% — **Maximum overlap:** 30%

Figure 50: YOLO-NAS results - part 3 - Where **Minimum confidence** is the minimum accepted confidence score & **Maximum overlap** Maximum accepted overlap between two bounding boxes

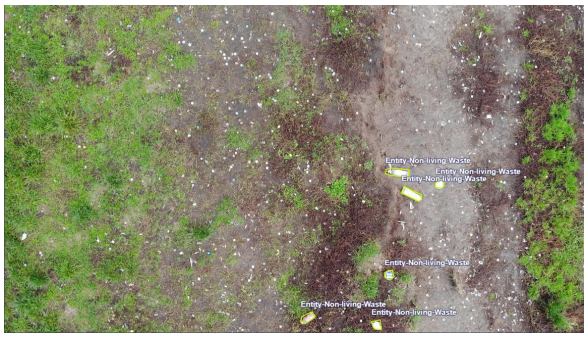
**Ground Truth** **YOLO-NAS**


**GT labels:** Other plastic, Plastic bottle cap

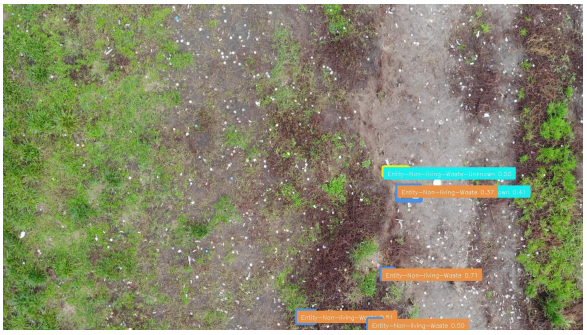
**Minimum confidence:** 30% — **Maximum overlap:** 10%


**GT labels:** Cigarette butt, Other plastic

**Minimum confidence:** 45% — **Maximum overlap:** 30%




**GT labels:** Waste



**Minimum confidence:** 32% — **Maximum overlap:** 10%

Figure 51: YOLO-NAS results - part 4 - Where **Minimum confidence** is the minimum accepted confidence score & **Maximum overlap** Maximum accepted overlap between two bounding boxes

### 5.4.2 Segment Anything Model (SAM)

This automated instance segmentation model's derived results can be seen in figures 52, 53 and 54.

Regarding the results seen in figure 52, almost all instances of solid waste were segmented correctly apart from:

- Second image from top: *Clear plastic bottle* has been over-segmented into two objects while it is clearly one object.
- Second image from bottom: Given the difficulty of the cluttered scene, normally the model has omitted segmenting some very small objects instances of label *other plastic* and three instances of *cigarette butt*.
- Bottom image: Two instances of *Drink can* have been over-segmented into two objects.

In figure 53 almost all instances of solid waste were segmented correctly apart from:

- Second image from top: One instance of *unknown waste* has been missed by the model and not segmented at all.

Lastly, in figure 54 the following remarks can be made:

- Top image: One instance of *Waste* has been over-segmented into two objects (top left instance).

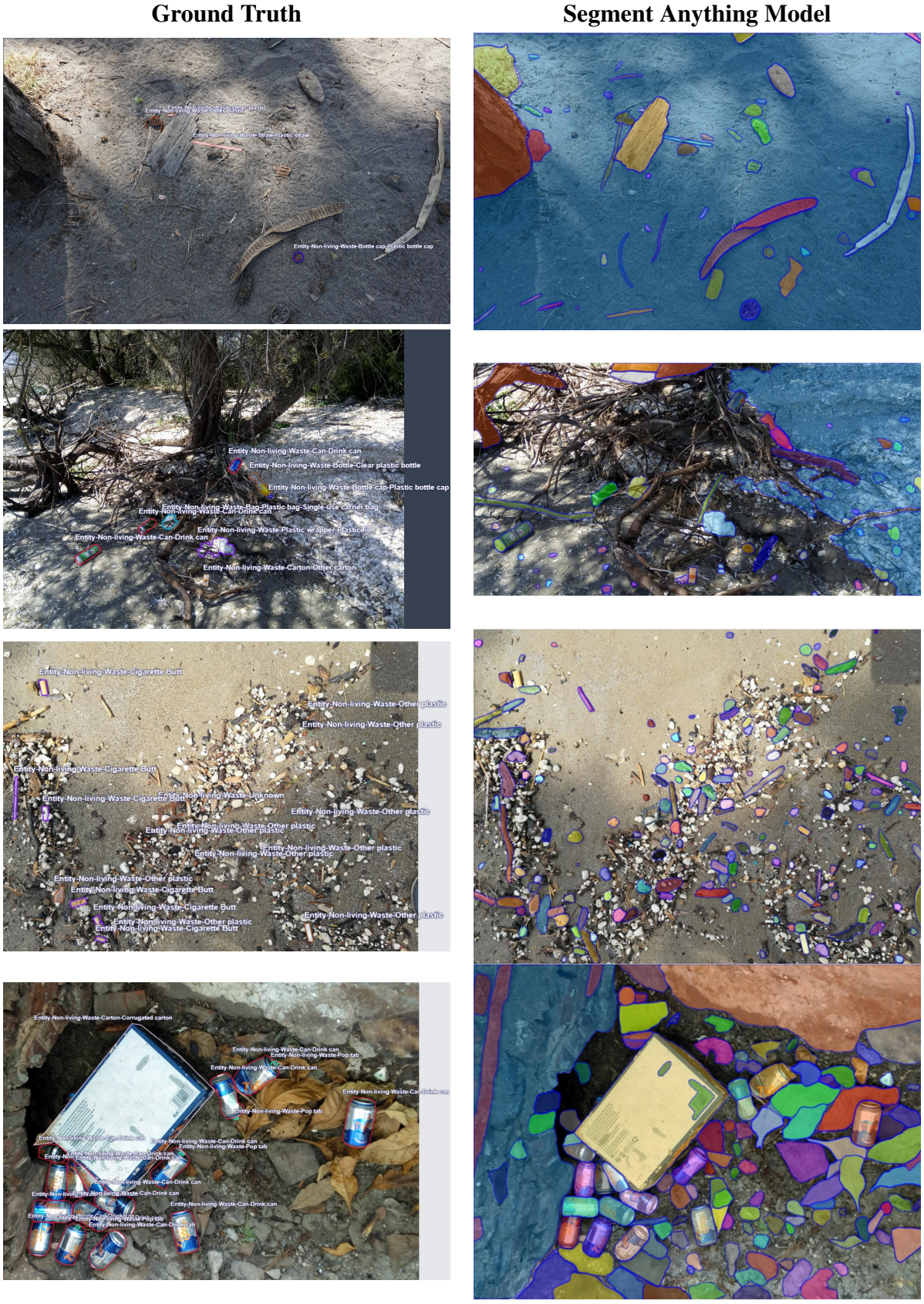



Figure 52: Segment Anything Model results - part 1 - To obtain SAM predictions the *Everything* functionality was used, which finds and segments all objects in the input image


Figure 53: Segment Anything Model results - part 2 - To obtain SAM predictions the *Everything* functionality was used, which finds and segments all objects in the input image

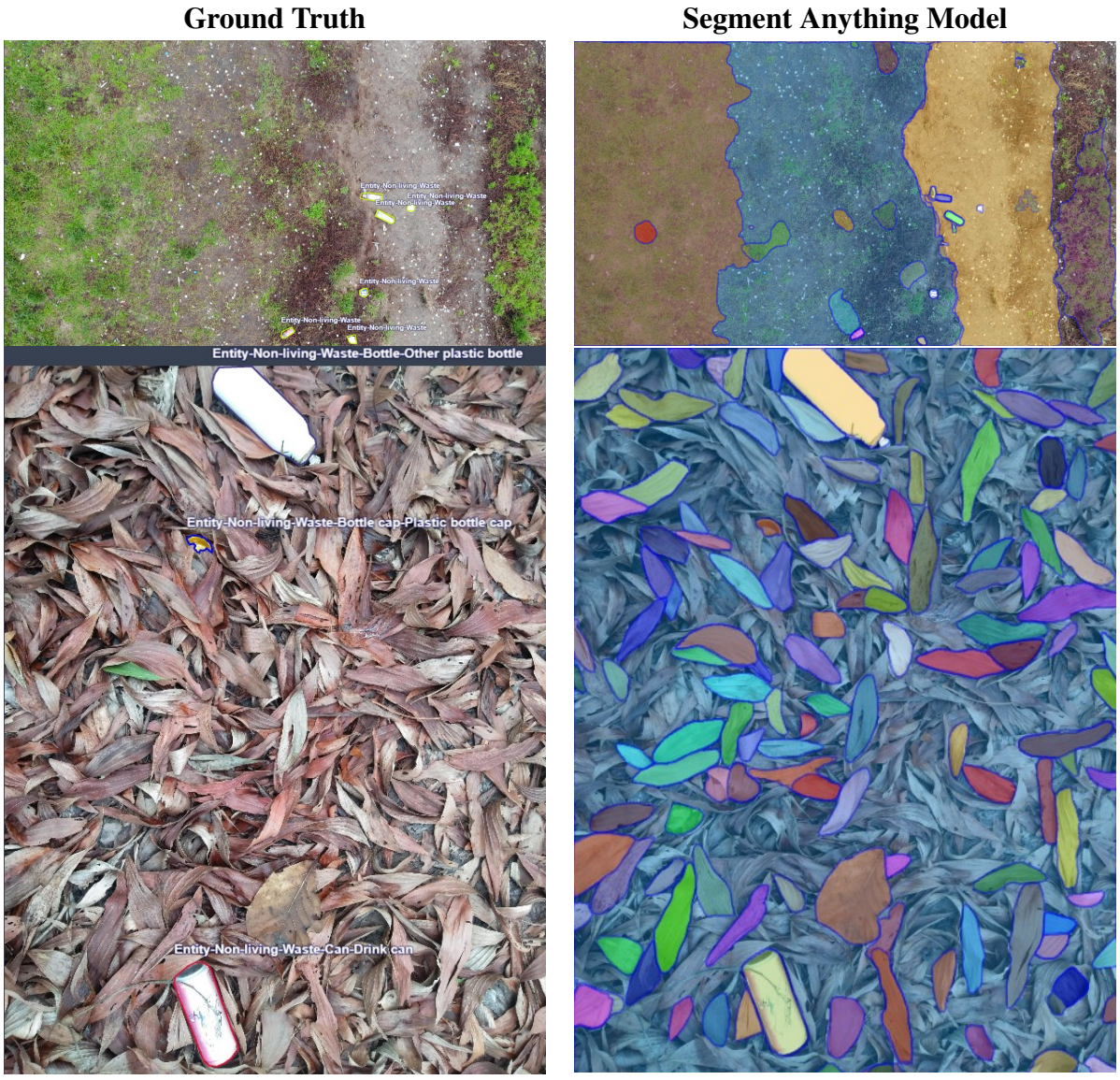



Figure 54: Segment Anything Model results - part 3 - To obtain SAM predictions the *Everything* functionality was used, which finds and segments all objects in the input image

## 5.5 Discussion of results

The following is a qualitative analysis of the generated results with general remarks on the performance of each computer vision model and a table *(table number 17)* that evaluates each model on a specific quality that corresponds to the task at hand. The scoring system is an effort to compare the obtained results of this work on a Likert scale and is not evaluating the models' overall performance numerically. Scoring is based on the author's interpretation regarding the perceived performance of the used computer vision models on the limited test sample that the obtained results represent. The scoring levels are the following:

- Very High:
  - Misclassifications: Model showcased a large number of incorrect classifications.
  - Multi-class classification accuracy: Almost perfect, seems like almost all cases are correctly classified.
  - Object Localization accuracy: Localization is highly accurate.
  - Segmentation masks accuracy: Highly accurate segmentation masks obtained.
  - Over-segmentation: Highly over-segmented objects, wrongly divided into multiple objects.
  - Under-segmentation: Highly under-segmented objects, wrongly merged into one object.
- High
  - Misclassifications: Model showcased a significant number of incorrect classifications.
  - Multi-class classification accuracy: Seems like most cases are correctly classified.
  - Object Localization accuracy: Localization is mostly accurate.
  - Segmentation masks accuracy: Mostly accurate segmentation masks obtained.
  - Over-segmentation: Significantly over-segmented objects, wrongly divided into multiple objects.
  - Under-segmentation: Significantly under-segmented objects, wrongly merged into one object.
- Moderate
  - Misclassifications: Model showcased a moderate number of incorrect classifications.
  - Multi-class classification accuracy: Moderately correct multi-class classification accuracy.
  - Object Localization accuracy: Localization is moderately accurate.
  - Segmentation masks accuracy: Moderately accurate segmentation masks obtained.
  - Over-segmentation: Moderately over-segmented objects, wrongly divided into multiple objects.
  - Under-segmentation: Moderately under-segmented objects, wrongly merged into one object.
- Low

| | Misclassifications (FP / FN) | Multiclass classification accuracy | Object Localization accuracy | Segmentation masks accuracy | Over-segmentation | Under-segmentation |
|---|---|---|---|---|---|---|
| Segment Anything Model | n/a | n/a | Very High | High | Low | Very Low |
| Grounding DINO | Low | Moderate | High | n/a | n/a | n/a |
| YOLO-NAS | High | Low | Low | n/a | n/a | n/a |
| YOLO-World | Moderate | Moderate | Moderate | n/a | n/a | n/a |

Table 17: Qualitative computer vision model comparison.

  - Misclassifications: Model showcased a low number of incorrect classifications.
  - Multi-class classification accuracy: Seems like a low number of instances are classified correctly on a multi-class basis. Poor performance.
  - Object Localization accuracy: Object localization was poor overall.
  - Segmentation masks accuracy: Mostly inaccurate segmentation masks obtained.
  - Over-segmentation: Some over-segmented objects, wrongly divided into multiple objects.
  - Under-segmentation: Some under-segmented objects, wrongly merged into one object.
- Very Low
  - Misclassifications: Model showcased almost no incorrect classifications.
  - Multi-class classification accuracy: Seems like almost no instance was classified correctly on a multi-class basis.
  - Object Localization accuracy: Object localization has some inaccuracies.
  - Segmentation masks accuracy: Highly inaccurate segmentation masks obtained.
  - Over-segmentation: Almost no over-segmented objects, wrongly divided into multiple objects.
  - Under-segmentation: Almost no under-segmented objects, wrongly merged into one object.
- n/a - Not applicable

### 5.5.1 Segment anything Model (SAM)

Doing a qualitative comparison between the generated results and the ground truth images, it is fair to say that the *SAM* model did a fairly good job segmenting the solid waste objects on the test images. There were some instances, in cluttered scenes, in which parts that belong to one object were segmented to two individual objects. This was observed to happen when there was an overlap or intersection of two separate objects, which ended up confusing the mask decoder to predict two separate

masks for the partially hidden object usually lying underneath another object. This model as a result had some instances of over-segmentation.

### 5.5.2 YOLO-NAS

Qualitatively comparing the results obtained by the closed-set detector *YOLO-NAS*, one can say that are good in terms of localizing solid waste objects but far from great in classifying them into the correct sub-class. We obtained a fair number of misclassifications from this detector and it could be that if the *YOLO-NAS-M* or *YOLO-NAS-L* variants were available we could obtain better results at the expense of greater training times.

### 5.5.3 YOLO-World

Upon qualitative inspection, the results of the open-set object detector *YOLO-World* had occasional misclassifications and multiple bounding boxes of various sizes for the same object tend to be generated in cluttered regions of the image where multiple objects are apparent. Overall, the results were decent.

### 5.5.4 GroundingDINO

Contrasting this model's results with the open-set object detector *YOLO-World*, less misclassifications were observed. Moreover, *GroundingDINO* using qualitative analysis, had issues accurately detecting smaller objects like plastic bottle caps and cigarette butts. Overall, it performed relatively better than *YOLO-World* or *YOLO-NAS* for the selected testing sample.

### 5.5.5 Auto-labelling capabilities with open-set detectors

Open-set detectors also referred to as Open vocabulary Detectors (OVDs) like *GroundingDINO* and *YOLO-World*, have shown quite promising results for automating the labeling of data in a zero-shot manner, judging from the test sample. With some adjustments like correcting misclassifications or even using combinations of computer vision models like *Grounded SAM* (Ren et al., 2024) which combines *GroundingDINO* with *SAM* can greatly accelerate dataset annotation. While man-in-the-loop and domain knowledge is still necessary, those models can save valuable time by avoiding having to manually go through each image and annotating all solid waste objects from scratch. If I had to pick between *GroundingDINO* and *YOLO-World*, I would go for *GroundingDINO* because of the superior quality of the solid waste detection, which would eventually mean that less manual corrections would be needed overall.

Moreover, in the future, pre-trained open-set detectors can serve as great foundation models that can do a decent initial data annotation, and after correcting for mistakes, the same dataset can be used to train a smaller, faster and more specialized closed-set object detector e.g. *YOLO variant* that will be able to produce decent results because of the available amount of input data and quality annotations produced in the previous step.

# 6 Conclusion

If I were to start again, I would be more mindful on which libraries to work with for the experimental part, to avoid wasting time trying to reverse engineer how the code works as a result of poor documentation. While some libraries are good for fast prototyping e.g. *Hugging Face or Roboflow*, others are not actively managed, and they end up having multiple issues that cannot be dealt with in time e.g. *Datumaro*. There are also actively maintained libraries that are well packaged, cloud native friendly (e.g. with a Docker container implementation) and ready to deploy in production across different devices e.g. *edge devices using Roboflow platform*.

As the results obtained with the closed-set and open-set object detectors above are far from perfect, there is for sure room for improvement and some areas that can be improved or are worth considering for future works are:

1. Make the label taxonomy less complex with less granular sub-classes and thus multi-class object detection will become less complex and allegedly more accurate. Preliminary tests performed in this work on vision-language models, showed that it is easier for models to learn to predict one class *e.g. garbage* versus multi-class dictionaries. The complexity of the task at hand increases as the number of unseen input classes increase.

2. Use even bigger datasets e.g. also include PlastoPol in the combined dataset.

3. Use open-set vocabulary object detectors to annotate open datasets that have no annotations to make the final size of the combined dataset even bigger.

4. Open-sourcing of proprietary datasets to accelerate the solid waste detection efforts worldwide.

All in all, although efficient waste management for reducing the amount of generated waste and especially plastic packaging is important, detecting solid waste reliably and fast is equally important for optimizing collection, recovery, and correct recycling, preferably at the site of disposal. For this cause, Artificial Intelligence can play a significant role on automating the detection of solid waste with various use cases including in natural settings.

# References

Alfeo, A. L., Ferrer, E. C., Carrillo, Y. L., Grignard, A., Pastor, L. A., Sleeper, D. T., ... Pentland, A. (2019, May). Urban Swarms: A new approach for autonomous waste management. In *2019 International Conference on Robotics and Automation (ICRA)* (pp. 4233–4240). (ISSN: 2577-087X) doi: 10.1109/ICRA.2019.8794020

Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., & Shan, Y. (2024, February). *YOLO-World: Real-Time Open-Vocabulary Object Detection.* arXiv. Retrieved 2024-02-22, from http://arxiv.org/abs/2401.17270 (arXiv:2401.17270 [cs]) doi: 10.48550/arXiv.2401.17270

*Cigarette Butt Dataset.* (n.d.). Retrieved 2022-09-25, from https://www.immersivelimit.com/datasets/cigarette-butts

CleanRobotics. (2024). *TrashBot: The smart recycling bin that sorts at the point of disposal.* Retrieved 2024-06-15, from https://cleanrobotics.com/trashbot/

Deci.ai. (2023, May). *YOLO-NAS by Deci Achieves State-of-the-Art Performance on Object Detection Using Neural Architecture Search.* Retrieved 2024-06-30, from https://deci.ai/blog/yolo-nas-object-detection-foundation-model/

Eurostat. (2023). *EU packaging waste generation with record increase.* Retrieved 2024-06-19, from https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20231019-1

Face, H. (2024, June). *Hugging Face – The AI community building the future.* Retrieved 2024-06-21, from https://huggingface.co/

Fang, B., Yu, J., Chen, Z., Osman, A. I., Farghali, M., Ihara, I., ... Yap, P.-S. (2023, August). Artificial intelligence for waste management in smart cities: a review. *Environmental Chemistry Letters*, *21*(4), 1959–1989. Retrieved 2024-06-17, from https://doi.org/10.1007/s10311-023-01604-3 doi: 10.1007/s10311-023-01604-3

Grove, S. (2024). *YOLO World - a Hugging Face Space by stevengrove.* Retrieved 2024-06-21, from https://huggingface.co/spaces/stevengrove/YOLO-World

Hong, J., Fulton, M., & Sattar, J. (2020, July). *TrashCan: A Semantically-Segmented Dataset towards Visual Detection of Marine Debris* (Tech. Rep. No. arXiv:2007.08097). arXiv. Retrieved 2022-05-22, from http://arxiv.org/abs/2007.08097 (arXiv:2007.08097 [cs] type: article) doi: 10.48550/arXiv.2007.08097

Issac, M. N., & Kandasubramanian, B. (2021, April). Effect of microplastics in water and aquatic systems. *Environmental Science and Pollution Research*, *28*(16), 19544–19562. Retrieved 2024-06-22, from https://doi.org/10.1007/s11356-021-13184-2 doi: 10.1007/s11356-021-13184-2

Kaza, S., Yao, L., Bhada-Tata, P., & Van Woerden, F. (2018). *What a Waste 2. 0: A Global Snapshot of Solid Waste Management To 2050*. Washington, D. C., UNITED STATES: World Bank Publications. Retrieved 2022-04-23, from http://ebookcentral.proquest.com/lib/rug/detail.action?docID=5614550

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... Girshick, R. (2023, April). *Segment Anything.* arXiv. Retrieved 2024-02-22, from http://arxiv.org/abs/2304.02643 (arXiv:2304.02643 [cs]) doi: 10.48550/arXiv.2304.02643

Kraft, M., Piechocki, M., Ptak, B., & Walas, K. (2021, January). Autonomous, Onboard Vision-Based Trash and Litter Detection in Low Altitude Aerial Images Collected by an Unmanned Aerial Vehicle. *Remote Sensing*, *13*(5), 965. Retrieved 2022-05-16, from https://www.mdpi.com/2072-4292/13/5/965 (Number: 5 Publisher: Multidisciplinary Digital Publishing Institute) doi: 10.3390/rs13050965

*Launch: Train and Deploy YOLO-NAS Models on Roboflow*. (2024, February). Retrieved 2024-02-22, from https://blog.roboflow.com/train-deploy-yolo-nas/

LeCun, Y., Bengio, Y., & Hinton, G. (2015, May). Deep learning. *Nature*, *521*(7553), 436–444. Retrieved 2024-06-23, from https://www-nature-com.proxy-ub.rug.nl/articles/nature14539 (Publisher: Nature Publishing Group) doi: 10.1038/nature14539

Lin, K., Zhao, Y., Kuo, J.-H., Deng, H., Cui, F., Zhang, Z., ... Wang, T. (2022, April). Toward smarter management and recovery of municipal solid waste: A critical review on deep learning approaches. *Journal of Cleaner Production*, *346*, 130943. Retrieved 2022-05-21, from https://www.sciencedirect.com/science/article/pii/S0959652622005807 doi: 10.1016/j.jclepro.2022.130943

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., ... Zhang, L. (2023, March). *Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection.* arXiv. Retrieved 2024-02-22, from http://arxiv.org/abs/2303.05499 (arXiv:2303.05499 [cs]) doi: 10.48550/arXiv.2303.05499

Lu, H., Diaz, D. J., Czarnecki, N. J., Zhu, C., Kim, W., Shroff, R., ... Alper, H. S. (2022, April). Machine learning-aided engineering of hydrolases for PET depolymerization. *Nature*, *604*(7907), 662–667. Retrieved 2024-06-20, from https://www-nature-com.proxy-ub.rug.nl/articles/s41586-022-04599-z (Publisher: Nature Publishing Group) doi: 10.1038/s41586-022-04599-z

Lu, W., & Chen, J. (2022, April). Computer vision for solid waste sorting: A critical review of academic research. *Waste Management*, *142*, 29–43. Retrieved 2022-04-30, from https://www.sciencedirect.com/science/article/pii/S0956053X22000678 doi: 10.1016/j.wasman.2022.02.009

Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M. A., Kwasigroch, A., & Majek, K. (2021, May). Waste detection in Pomerania: non-profit project for detecting waste in environment. *arXiv:2105.06808 [cs, eess]*. Retrieved 2022-04-25, from http://arxiv.org/abs/2105.06808 (arXiv: 2105.06808)

Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M. A., Kwasigroch, A., & Majek, K. (2022, February). Deep learning-based waste detection in natural and urban environments. *Waste Management*, *138*, 274–284. Retrieved 2022-04-22, from https://www.sciencedirect.com/science/article/pii/S0956053X21006474 doi: 10.1016/j.wasman.2021.12.001

Mikołajczyk, A. (2024, April). *AgaMiko/waste-datasets-review.* Retrieved 2024-04-03, from https://github.com/AgaMiko/waste-datasets-review (original-date: 2021-01-10T16:56:07Z)

National-Geographic-Society. (2019, March). *National Geographic Society | Plastic impact on marine ecosystem.* Retrieved 2024-06-22, from https://www.nationalgeographic.com/environment/article/whale-dies-88-pounds-plastic-philippines (Section: Environment)

NPR. (2024). *Plastics: What's Recyclable, What Becomes Trash — And Why.* Retrieved 2024-06-20, from https://apps.npr.org/plastics-recycling/

*openvinotoolkit/datumaro.* (2024, April). OpenVINO™ Toolkit. Retrieved 2024-04-15, from https://github.com/openvinotoolkit/datumaro (original-date: 2020-09-01T14:13:24Z)

Osman, A. I., Hosny, M., Eltaweil, A. S., Omar, S., Elgarahy, A. M., Farghali, M., … Akinyede, K. A. (2023, August). Microplastic sources, formation, toxicity and remediation: a review. *Environmental Chemistry Letters*, *21*(4), 2129–2169. Retrieved 2024-06-22, from https://doi.org/10.1007/s10311-023-01593-3 doi: 10.1007/s10311-023-01593-3

Pawaskar, A. R., & Dhanya, N. M. (2022). Waste Object Segmentation for Autonomous Waste Segregation. In S. Aurelia, S. S. Hiremath, K. Subramanian, & S. K. Biswas (Eds.), *Sustainable Advanced Computing* (pp. 147–159). Singapore: Springer. doi: 10.1007/978-981-16-9012-9_13

Proença, P. F., & Simões, P. (2020, March). TACO: Trash Annotations in Context for Litter Detection. *arXiv:2003.06975 [cs]*. Retrieved 2022-04-22, from http://arxiv.org/abs/2003.06975 (arXiv: 2003.06975)

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In (pp. 7263–7271). Retrieved 2022-05-22, from https://openaccess.thecvf.com/content_cvpr_2017/html/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.html

Ren, T., Liu, S., Zeng, A., Lin, J., Li, K., Cao, H., … Zhang, L. (2024, January). *Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks.* arXiv. Retrieved 2024-06-08, from http://arxiv.org/abs/2401.14159 (arXiv:2401.14159 [cs]) doi: 10.48550/arXiv.2401.14159

Roboflow. (2024). *Roboflow: Computer vision tools for developers and enterprises.* Retrieved 2024-05-26, from https://roboflow.com/

Sadako-Technologies. (2024). *Max-AI – Sadako Technologies.* Retrieved 2024-06-18, from https://sadako.es/max-ai/

*Segment Anything | Meta AI.* (2023). Retrieved 2024-05-26, from https://segment-anything.com/

Sensoneo. (2024). *Monitoring Waste Containers.* Retrieved 2024-06-15, from https://sensoneo.com/use-case/monitoring-waste-containers/

Shahab, S., Anjum, M., & Umar, M. S. (2022). Deep Learning Applications in Solid Waste Management: A Deep Literature Review. *International Journal of Advanced Computer Science and Applications*, *13*(3). Retrieved 2024-06-20, from http://thesai.org/Publications/ViewPaper?Volume=13&Issue=3&Code=IJACSA&SerialNo=47 doi: 10.14569/IJACSA.2022.0130347

Skalski, P. (2023, May). *How to Train YOLO-NAS on a Custom Dataset.* Retrieved 2024-05-02, from https://blog.roboflow.com/yolo-nas-how-to-train-on-custom-dataset/

Stone, W. (2024, May). Plastic junk? Researchers find tiny particles in men's testicles. *NPR*. Retrieved 2024-06-22, from https://www.npr.org/sections/health-shots/2024/05/22/1252831827/microplastics-testicles-humans-health

UCPH. (2024, June). *Researchers invent one hundred percent biodegradable "barley plastic"*. Retrieved 2024-06-20, from https://news.ku.dk/all_news/2024/06/researchers-invent-one-hundred-percent-biodegradable-barley-plastic/ (Publisher: University of Copenhagen)

UNESCO. (2017). *The Ocean Conference, 5-9 June, 2017 - United Nations, New York.* Retrieved 2024-06-22, from https://www.un.org/en/conf/ocean/ (Publisher: United Nations)

Unsplash. (2019, October). *Photo by Antoine GIRET on Unsplash.* Retrieved 2024-06-15, from https://unsplash.com/photos/garbage-near-forest-7_TSzqJms4w

Unsplash. (2021, May). *Photo by Naja Bertolt Jensen on Unsplash.* Retrieved 2024-06-15, from https://unsplash.com/photos/school-of-fish-in-water-BJUoZu0mpt0

Valdenegro-Toro, M. (2019, May). Deep Neural Networks for Marine Debris Detection in Sonar Images. *arXiv:1905.05241 [cs]*. Retrieved 2022-05-19, from http://arxiv.org/abs/1905.05241 (arXiv: 1905.05241)

Valdenegro-Toro, M. (2021, April). *I Find Your Lack of Uncertainty in Computer Vision Disturbing*. arXiv. Retrieved 2024-06-23, from http://arxiv.org/abs/2104.08188 (arXiv:2104.08188 [cs]) doi: 10.48550/arXiv.2104.08188

Vierah Hulley. (2020). *Waste Management*. Ashland: Delve Publishing. Retrieved 2022-04-23, from http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2725403&site=ehost-live&scope=site

Wayman, C., & Niemann, H. (2021, March). The fate of plastic in the ocean environment – a minireview. *Environmental Science: Processes & Impacts*, *23*(2), 198–212. Retrieved 2022-04-23, from https://pubs.rsc.org/en/content/articlelanding/2021/em/d0em00446d (Publisher: The Royal Society of Chemistry) doi: 10.1039/D0EM00446D

White, G., Cabrera, C., Palade, A., Li, F., & Clarke, S. (2020, June). WasteNet: Waste Classification at the Edge for Smart Bins. *arXiv:2006.05873 [cs]*. Retrieved 2022-04-22, from http://arxiv.org/abs/2006.05873 (arXiv: 2006.05873)

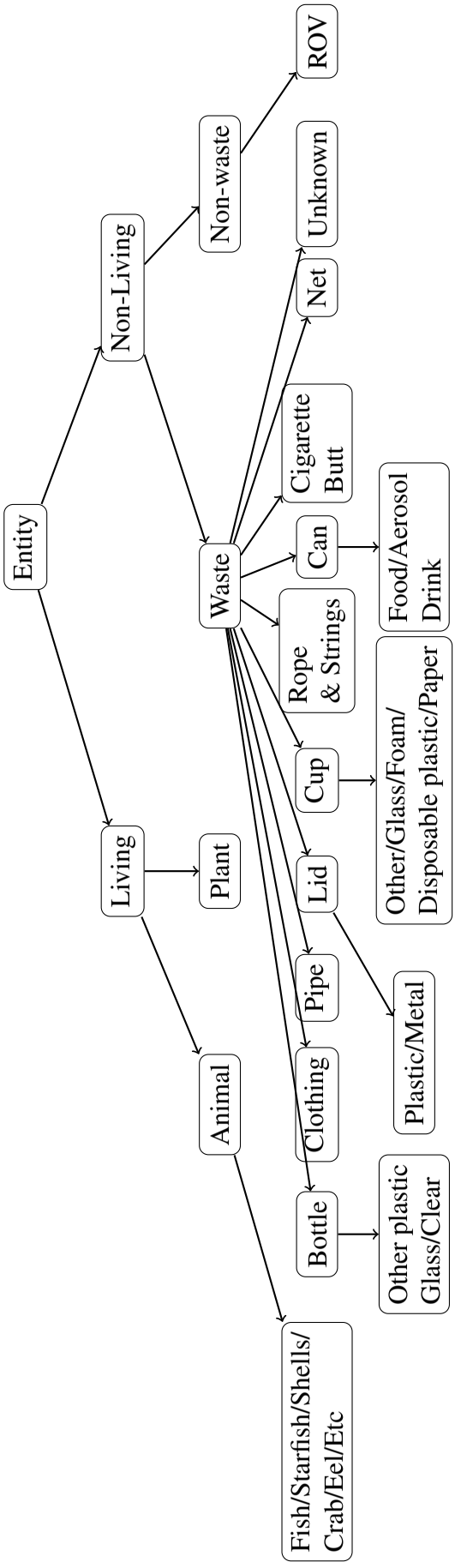
Xu, J., Sagnelli, D., Faisal, M., Perzon, A., Taresco, V., Mais, M., ... Blennow, A. (2021, February). Amylose/cellulose nanofiber composites for all-natural, fully biodegradable and flexible bioplastics. *Carbohydrate Polymers*, *253*, 117277. Retrieved 2024-06-20, from https://www.sciencedirect.com/science/article/pii/S0144861720314508 doi: 10.1016/j.carbpol.2020.117277

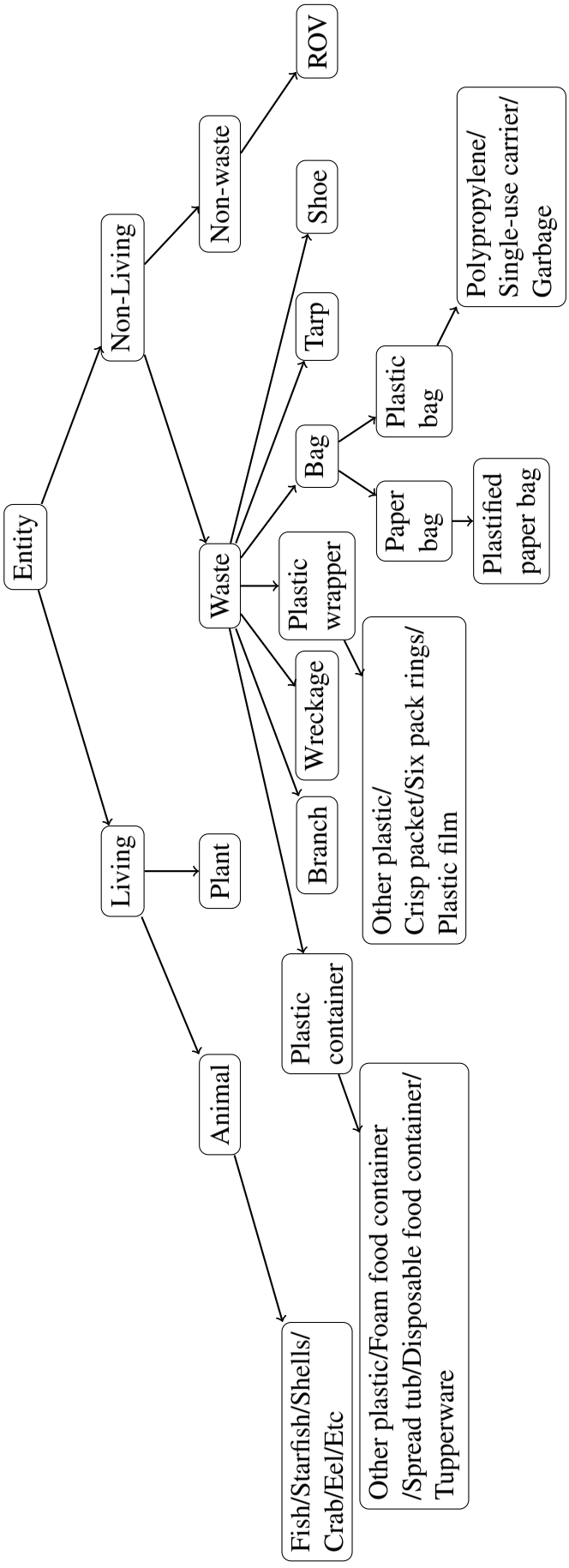
Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021, July). Dive into Deep Learning. *arXiv:2106.11342 [cs]*. Retrieved 2022-05-11, from http://arxiv.org/abs/2106.11342 (arXiv: 2106.11342)

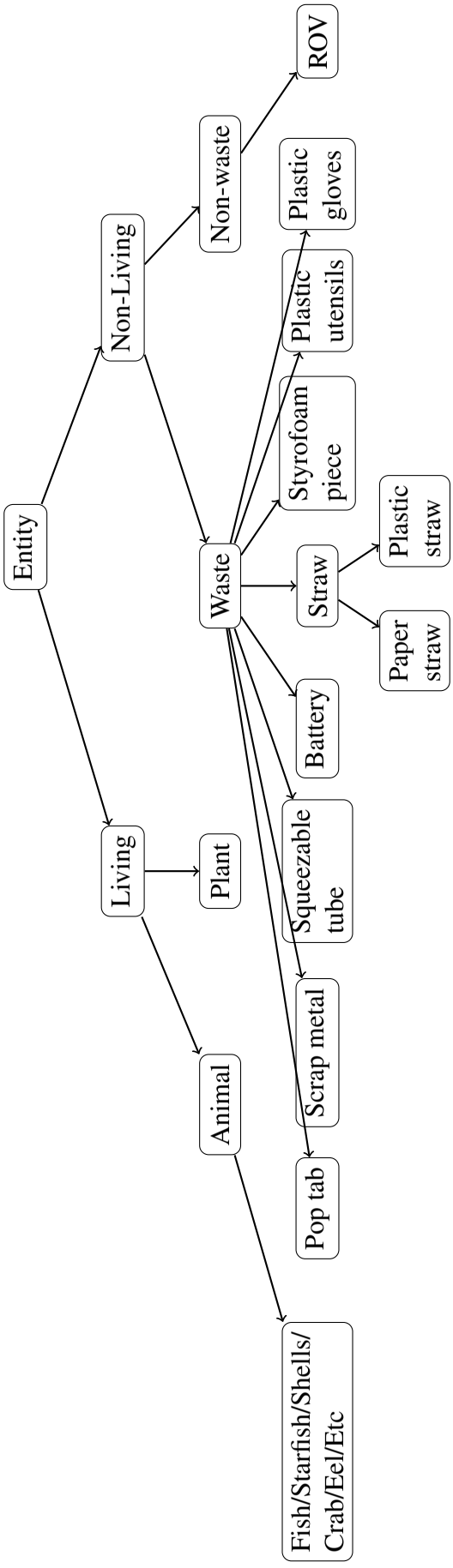
Zhang, P., Zhao, Q., Gao, J., Li, W., & Lu, J. (2019). Urban Street Cleanliness Assessment Using Mobile Edge Computing and Deep Learning. *IEEE Access*, *7*, 63550–63563. (Conference Name: IEEE Access) doi: 10.1109/ACCESS.2019.2914270

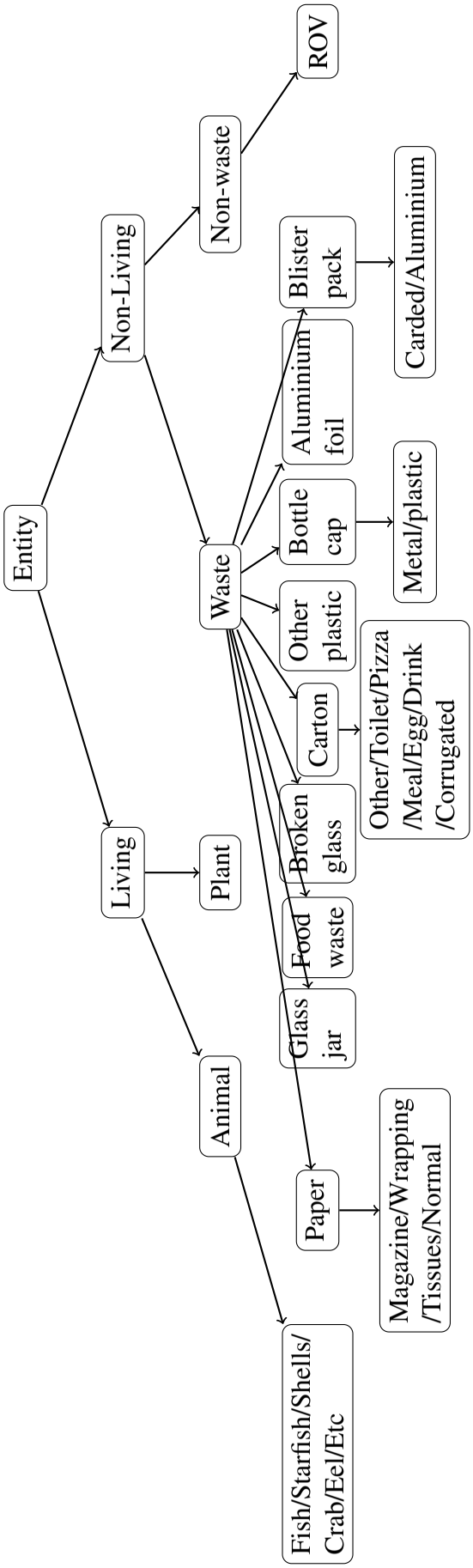
# Appendices

In this section, the decision trees of the label space taxonomy follow, to guide the reader that wants to dive deeper into each leaf of this multi-class solid waste detection problem. Each decision tree is supplementary to the next one and combining them all together adds up to the entire label space taxonomy.
The following pages are orientated as landscape pages to better accommodate each decision tree at hand. Those decision trees correspond to the label space taxonomy, which in turn correspond to the mapping file used in the experiments section.


Entity
Non-Living
Non-waste
ROV
Living
Plant
Animal
Waste
Unknown
Net
Cigarette Butt
Can
Food/Aerosol Drink
Rope & Strings
Cup
Other/Glass/Foam/ Disposable plastic/Paper
Lid
Pipe
Plastic/Metal
Clothing
Bottle
Other plastic Glass/Clear
Fish/Starfish/Shells/ Crab/Eel/Etc


Entity
Non-Living
Non-waste
ROV
Waste
Shoe
Tarp
Bag
Plastic bag
Polypropylene/ Single-use carrier/ Garbage
Paper bag
Plastified paper bag
Plastic wrapper
Other plastic/ Crisp packet/Six pack rings/ Plastic film
Wreckage
Branch
Living
Plant
Animal
Plastic container
Other plastic/Foam food container /Spread tub/Disposable food container/ Tupperware
Fish/Starfish/Shells/ Crab/Eel/Etc


Entity
Non-Living
Non-waste
ROV
Waste
Plastic gloves
Plastic utensils
Styrofoam piece
Straw
Plastic straw
Paper straw
Battery
Living
Plant
Squeezable tube
Animal
Scrap metal
Pop tab
Fish/Starfish/Shells/ Crab/Eel/Etc


Entity
Non-Living
Living
Non-waste
ROV
Waste
Plant
Animal
Blister pack
Carded/Aluminium
Aluminium foil
Bottle cap
Metal/plastic
Other plastic
Carton
Other/Toilet/Pizza /Meal/Egg/Drink /Corrugated
Broken glass
Food waste
Glass jar
Paper
Magazine/Wrapping /Tissues/Normal
Fish/Starfish/Shells/ Crab/Eel/Etc