Towards Sustainable Waste Solutions: Deep Learning-Driven Waste Detection and Sorting Technologies

Beatrice Bonato

April 30, 2024

Abstract

This project explores the application of powerful Deep Learning techniques in addressing the urgent problem of waste management. Utilizing the state-of-the-art YOLOv8 model, this study demonstrates how computer vision can be leveraged to identify and classify various types of waste, improving the efficiency of recycling processes. Through the training of a neural network model on the TACO dataset [1], which includes a diversified range of waste images, my work highlights the potential of integrating these types of trained models into robotic systems, creating the possibility to automate the waste sorting process effectively. These results show promising directions for real-time waste detection, suggesting significant potentials for sustainable waste management practices globally. The findings call for the continued development of more balanced and richer datasets, and integrated robotic systems to further improve the accuracy and applicability of waste detection technologies.

1. Introduction

The growing global waste management crisis is a significant challenge, and it's predicted an alarming increase in municipal solid waste from 2.24 billion tons in 2020 to 3.88 billion tons by 2050. This increase is mostly driven by rapid urbanization and economic growth, particularly in developing regions where waste management infrastructure often lacks capacity or efficiency. As a result, a significant portion of waste is not managed in an environmentally safe manner, worsening public health and environmental problems [2].

As global waste pollution increases, it's essential to implement innovative solutions to advance sustainable waste management practices. This project leverages deep learning, specifically the YOLOv8 model, to develop a vision-based model capable of identifying and classifying various types of waste for recycling purposes. The YOLOv8 model is a state-of-the-art computer vision tool known for its great efficiency in real-time object detection tasks [3]. Despite its capabilities, the model has limitations due to the complexity and imbalance inherent in the waste dataset used, leading to promising but limited accuracy in classifying waste types. This reflects a common obstacle in the application of AI technologies, where data quality has a significant impact on the outcome. The integration of deep learning into waste management is not entirely new, but it builds on previous research. For instance, studies by Mittal et al. and Thung and Yang have demonstrated the use of machine learning models to classify waste types from images, achieving relevant accuracy improvements over traditional methods [4],[5]. However, these studies also highlight the problem of data imbalance, a common obstacle that can worsen model performance and limit its practical utility. Furthermore, the research on robotic sorting systems, such as those presented by Sahani et al. illustrates the potential of combining deep learning with mechanical sorting mechanisms to enhance recycling processes [6]. These systems, however, require highly accurate detection and classification models to be implemented in the real-world.

Addressing these challenges would improve waste management and pave the way for more advanced and sustainable practices worldwide. By improving data processing and model training, we can get more accurate and reliable systems that could automatically identify and sort waste.

2. Analyzing the Dataset: Key Features and Challenges

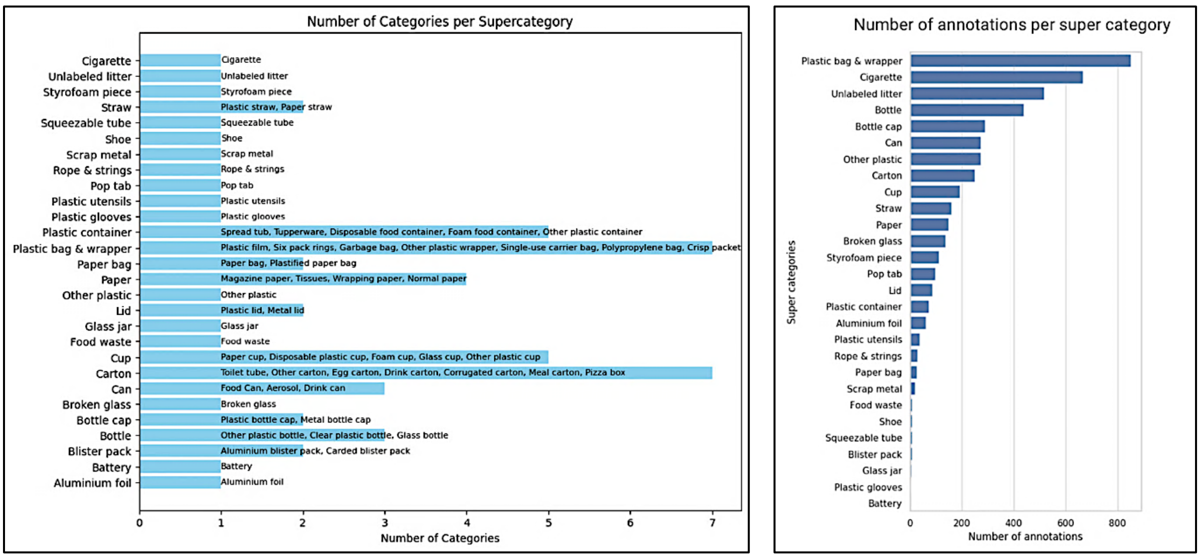
An overview of the waste datasets on the web, as shown in Table1 below, reveals a bias toward classification-type datasets with only few detection-type datasets available for the purpose. These classification-type datasets have limited utility for the specific requirements of detection models that rely on the precise location of objects within images. Detection tasks require datasets annotated with bounding boxes, which are particularly poor. Considering the available data, the TACO dataset emerges as the only valid candidate for the scope of the project. Despite the dataset's limitations, which will be explained below in the report, it remains the only option that meets the requirements for detection-oriented data. Therefore, the TACO dataset was selected as the basis for developing the YOLOv8 model for this project.

Figure1: Waste Datasets Overview [7]

Dataset Name	No. categories	No. subcategories	No. images	Annotation	Description
DeepSeaWaste	5	-	3055	Classification	Underwater images
WaDaBa	8	color,size,shape, or material	4000	Classification	Plastic dataset, clear background
Trashnet	6	-	2527	Classification	Clear background
GLASSENSE-VISION	7	136	2000	Classification	Home-supplies, clear background
spotgarbage	3	-	~2400	Classification	Scraped from Bing search
waste_pictures	34	-	~24000	Classification	Scraped from google search
Waste Classification data	2	-	~25000	Classification	Scraped from google search
Waste Images from Sushi Restaurant	16	-	500	Classification	Clear background
Waste Classification Data v2	3	-	~27500	Classification	Scraped from google search
TrashBox	7	25	17785	Classification	Scraped from web
Litter	24	size, shape, or material	~14000	Detection	Waste in the wild, paid license
Trash-ICRA19	3	34	5700	Detection	Underwater images
TACO bboxes	7	60	WIP	Detection	Waste in the wild
Drinking Waste Classification	4	-	9640	Detection	Clear background, (cans and bottles)
Cigarette butt dataset	1	-	2200	Detection	Waste in the wild, synthetic images
TrashCan 1.0	3	34	7212	Instance-Segmentation	Underwater images
Open litter map	11	187	>100k	Multilabel classification	Waste in the wild
MJU-Waste v1.0	1	-	2475	Segmentation	Plain background, indoor RGBD
TACO	28	60	1500	Segmentation	Waste in the wild
UAVVaste	1	-	772	Segmentation	Drone dataset

The TACO (Trash Annotations in Context) dataset is an open image dataset of waste in the wild and is a significant resource for litter detection and segmentation in diversified environments. It includes 1500 images with 4784 annotations that enable the training of deep learning models for detecting litter in different natural environments. Despite its relatively small size, TACO was chosen as the dataset for this project because of its bounding box annotation which are essential for detection tasks, and thus suitable for the development of the YOLOv8 model. Currently the 4784 annotations are labeled in 60 categories which belong to 28 super-categories as you can see in Figure2.

Figure2: Composition and distribution of the TACO dataset.



The problem with the TACO dataset is that it has a level of specificity that is not necessary or convenient for the purpose of this project. With 60 categories distributed across 28 super categories, it provides a number of classes that goes beyond the requirements for training a waste detection model focused on broader categories relevant to waste management and recycling processes. Moreover, the distribution of annotations across these categories shows a strong imbalance. Some classes, such as “Plastic bag and wrapper,” “Cigarette,” and “Bottle,” have a high number of annotations. In contrast, many other categories are poorly annotated. This imbalance could lead to a biased model, which performs well with overrepresented classes but fails to recognize underrepresented ones.

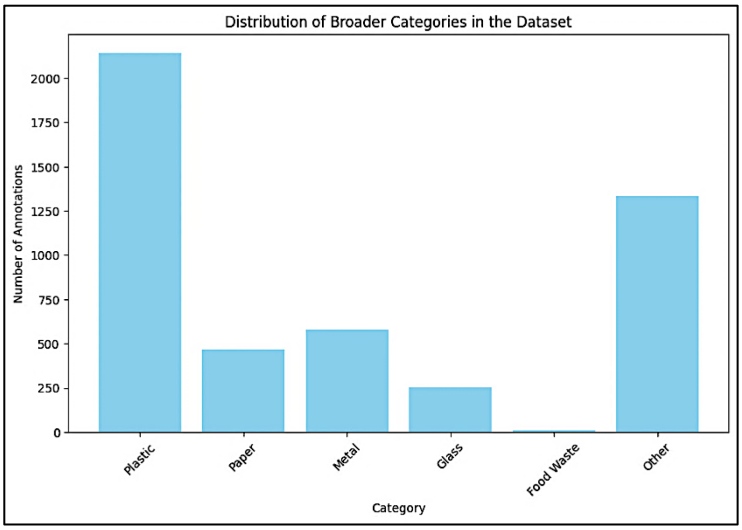
3. Data Processing

3.1 Categories Mapping and Data format transformation

For this project, the entire work, from data processing to model training, was done within Google Colab, taking advantage of its cloud computing capabilities. The first task in the data preparation phase was to address the complex structure of the TACO dataset, which originally contained 60 fine-grained categories of waste items.

Given the too detailed taxonomy of the TACO dataset, an important step was to group the 60 distinct categories into six more relevant waste types: Plastic, Paper, Metal, Glass, Food Waste, and Other. To accomplish this, I performed a mapping, translating the 60 original categories into the six broader classes. This mapping was done to ensure that the dataset would be meaningful for training a model aimed at detecting waste types that are relevant to recycling applications. The result of the mapping is shown in Figure3.

Figure3: Result of the mapping of the 60 categories into 6 main categories.



WASTE TYPE	N° OF ANNOTATIONS
Plastic:	2141
Paper:	467
Metal:	580
Glass:	254
Food Waste:	8
Other:	1334

After the mapping the TACO dataset then required conversion from COCO to YOLO annotation format to train the YOLOv8 model. The coding script processed each annotation from the COCO to the YOLO format within the available batches. This process had to be careful done since the original TACO dataset was organized in 15 batches, and images within them often shared filenames among batches, risking data overwriting during the transformation process. To avoid this, the code prefixed the filenames with their respective batch identifiers, making sure each image and its annotations remained unique. The script then moved and renamed the files, merging them into a single dataset ready for the model training stage. This attention to file renaming was fundamental to ensure that each image's annotations corresponded to the correct instance.

3.2 Data Augmentation 1

After the transformation of the TACO dataset's annotations to the YOLO format and combining the dataset into a single folder, the next phase involved dataset augmentation. This process was done using Roboflow platform. In Roboflow, the dataset of 1500 images was expanded to create a new version consisting of 3500 images through the application of various data augmentation techniques:

- Rotation by 90 degrees in different orientations (clockwise, counter-clockwise, and upside down) to ensure the model's rotational invariance.
- Random cropping with a range of 0% to 20% zoom to simulate the effect of objects appearing at different scales in images.
- Slight rotation within a range of -15 to +15 degrees to mimic the natural variations and slight tilts that are present in real-world scenarios.
- Introduction of noise to up to 1.01% of pixels in an image to enhance the model's robustness against variations in image quality.

These augmentations aimed to address the unbalanced and complex nature of the original dataset, which otherwise might limit the model's ability to generalize to new data. Indeed, these techniques help mitigate this problem by introducing controlled variability into the dataset, which is also beneficial for dealing with real-world conditions where waste items can be oriented differently, partially obscured, or captured in varying lighting conditions and qualities. The augmented dataset was then divided into training, validation, and test sets, with 3150, 375, and 75 images respectively.

The benefits of such a diversified augmentation process are many:

- It can lead to improved model accuracy and generalization by giving a wide range of scenarios during training.
- It helps to prevent overfitting, as the model is less likely to learn noise and too specific details from the training data.

- It can be helpful for classes with fewer samples, as augmented data can provide additional information that the model may not have learned otherwise.

4. Model Training

4.1 Model Training 1

After the dataset preparation, I could start with the training phase. Among the various models available for object detection, YOLOv8 was chosen for its very good performances. As the latest in the YOLO series, YOLOv8 excels in real-time detection and accuracy [3]. Moreover, it showed improved generalization, which is fundamental for handling the diversified conditions of waste items. This state-of-the-art model maintains high performance even in challenging environments, making it an appropriate choice for this application.

YOLOv8 has several variants, such as YOLOv8n (nano), YOLOv8m (medium), and others, each with different trade-offs between speed, size, and accuracy. YOLOv8s (small) was chosen for this project since it provides a good balance between performance and resource usage, which must be taken in consideration when dealing within the computational limitations of Google Colab. In the model training phase, it's important to fine-tune the hyperparameters to achieve a good performance. Below I explain the chosen hyperparameter for training the YOLOv8s model, compared to the default ones as you can see in Table2.

Table2: Default hyperparameters vs Chosen hyperparameters.

hyperparameters	DEFAULT	hyperparameters	USED
epochs	100	epochs	40
batch	16	batch	8
cls	0.5	cls	1
box	7	box	5.5
dfl	1.5	dfl	2.0
label_smoothing	0.0	label_smoothing	0.1

1. Batch Size: The batch size was decreased from 16 to 8 allow for more frequent model updates, which can be beneficial for learning from an unbalanced and not too wide dataset.

2. Class Weight (cls): The class weight was increased from 0.5 to 1 to give more importance on class prediction accuracy. In the context of an unbalanced dataset, this

adjustment helps the model to focus more on learning to distinguish between different classes.

3. Box Weight (box): The weight for bounding box predictions was decreased from 7 to 5.5. This was done to balance the model's focus between precise box predictions and correct classification.

4. Distribution Focal Loss (dfl): The dfl parameter was increased from 1.5 to 2.0 to enhance the model's focus on harder-to-classify examples.

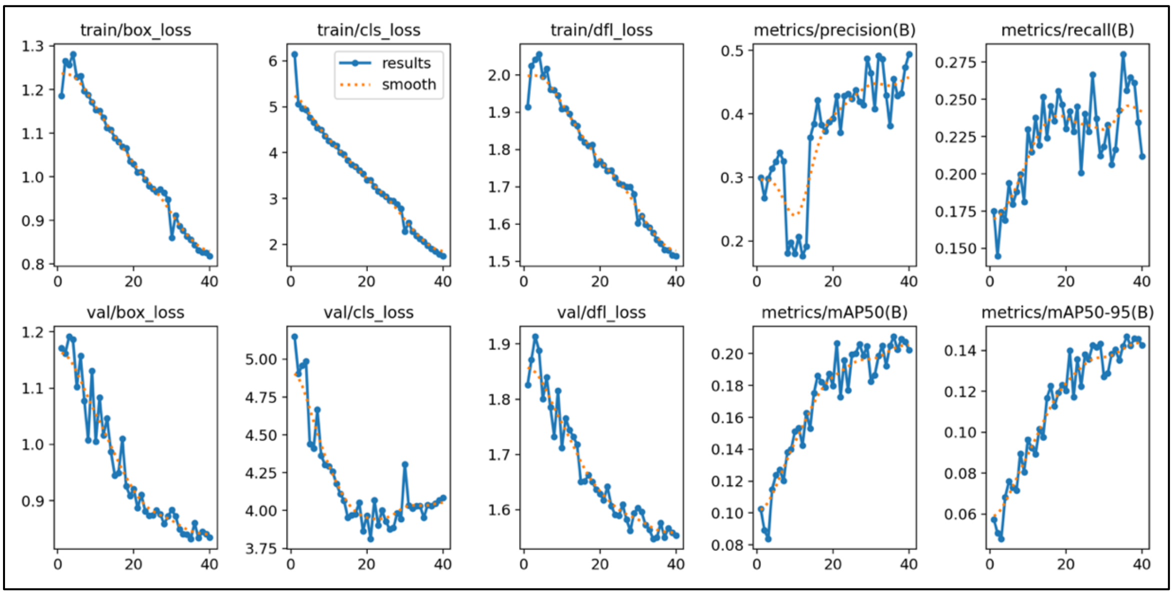
5. Label Smoothing (Label_smoothing): A label smoothing value of 0.1 was introduced from the default 0, which leads to improved generalization and robustness against noisy labels.

These choices were made to mitigate the challenges coming from the TACO dataset, thus enhancing the model's ability to learn from an uneven distribution of data and improve its predictive performance across all classes.

4.2 Results 1

Analyzing the results from the various plots and graphs generated during training we could see the efficacy of the training on the dataset. Looking at the Precision-Recall Curve, I could notice that while some classes like Plastic and Metal achieve reasonable precision across varying recall levels, classes like Food Waste exhibit poor performance. This indicates challenges the model faces with less represented classes despite augmentation.

Figure4: Dashboard of training and validation metrics for training 1.



Looking at the training results of Figure4, what one can notice is that even dough all the metrics improve over the epochs in the validation set, the classification loss starts to increase instead of decreasing after the 20th epoch of training, showing that the model is apparently not getting better at classifying on unseen data going on with the training.

<u>Figure5: Normalized Confusion Matrix for training 1</u>

The normalized confusion matrix in Figure5 shows some expected outcomes. As indicated by the Precision-Recall graph, the model performs well enough in classifying plastic and metal but encounters more challenges with paper and glass. This is consistent with the class imbalance, as you could also see from the absence of results for Food Waste, which is an underrepresented class in the dataset.

Many waste objects have been classified as background, which can be also attributed to the dataset's complexity, where small pieces of trash in natural settings may be easily mistaken for the background. Additionally, we must consider that a significant portion of the original dataset, nearly 700 annotations, were about cigarette butts. Therefore, you can easily understand the high likelihood of these items being confused with background due to their small size.

This outcome highlights the need for further dataset refinement or specialized training techniques that can help the model distinguish small or complex waste items from their environment. Overall, I got some encouraging results in the test set. The classification results showed a bias over the Plastic class as expected, but in many cases the model is accurately able to detect different waste types that are not too complex.

BAD RESULTS:

GOOD RESULTS:

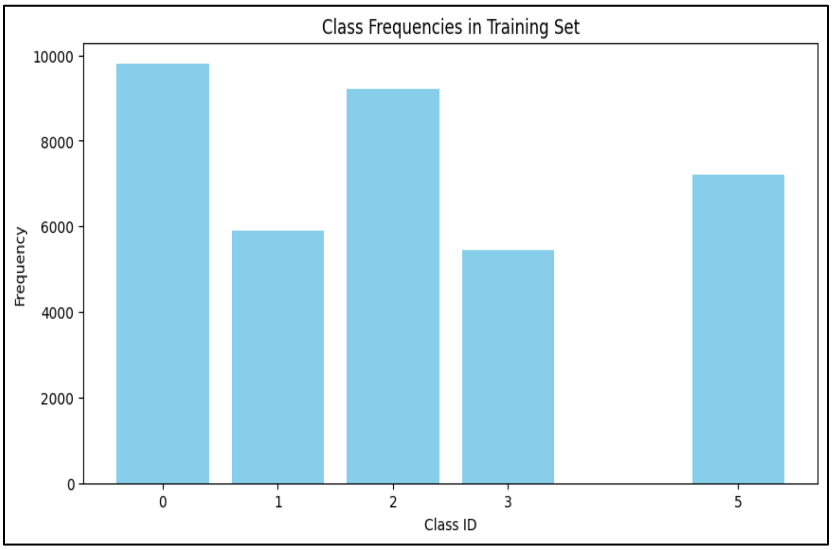
4.3 Data Augmentation 2

After the first training I wanted to see whether with a more balanced dataset I could get some better accuracy overall. To do that I applied an iterative oversampling. The first oversampling I took was aimed to address class imbalance by replicating instances of the underrepresented classes within the dataset: Paper was tripled, Metal tripled, Glass quadrupled, and Food Waste increased by a factor of thirty. However, this method also amplified the presence of already overrepresented classes such as Plastic and Other due to their co-occurrence in images with underrepresented classes.

To fix the approach, I applied a second oversampling, but this time targeting images that contained only objects from underrepresented categories. Paper was tripled, Metal tripled and Glass was tripled as well. By doing so, the process ensured that the augmentation did not create any other instance of the already overrepresented 'Plastic' and 'Other' categories. Additionally, I decided to merge the 'Food Waste' category with the 'Other' category. This was done because the class was too minimally represented to be significant.

This second oversampling approach led to a final training set with 13,086 images, a significant increase resulting in a dataset that, while still unbalanced, has a better distribution of classes than the original as you can see in Figure6 below.

Figure6: Class frequencies in the new training set. 0: Plastic, 1: Paper, 2: Metal, 3: Glass, 5: Other.



4.4 Model Training 2

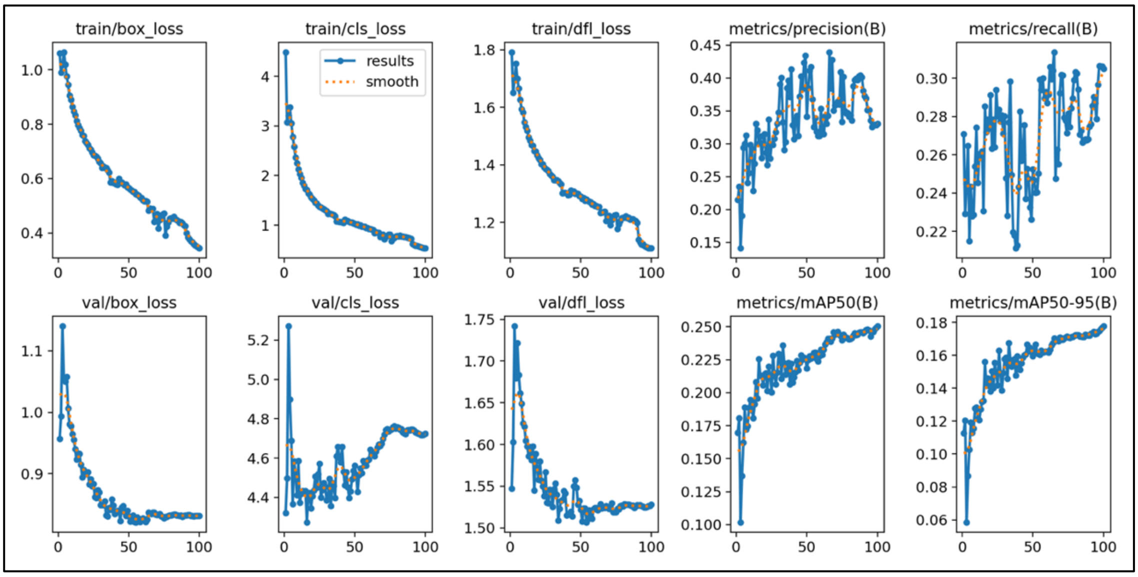
Table3: Chosen hyperparameters for Round1 and Round2

PARAMETERS	ROUND1
model	Yolov8s
epochs	First 70/100
cls	1
dfl	2.0
label_smoothing	0.1
batch	16
box	5.5

PARAMETERS	ROUND 2
model	Yolov8s
epochs	Last 30/100
cls	3
dfl	3.0
label_smoothing	0.1
batch	16
box	2.5

After the second process of data augmentation another training was performed. This time, since having a wider training dataset of 13,086 images, I explored a different hyperparameter setting. Therefore, I trained the model with the same parameters of the first training for the first 70/100 epochs, and changed the parameters for the last 30/100 epochs, weighting more the classification loss compared to the box loss.

Figure7: Dashboard of training and validation metrics for training 2.



Observations from the Training Results in Figure7:

Loss Metrics: There is an improvement in both the training box loss and the training classification loss, which suggests that the model has become more accurate in bounding box predictions and class identifications.

Validation Set Performance:

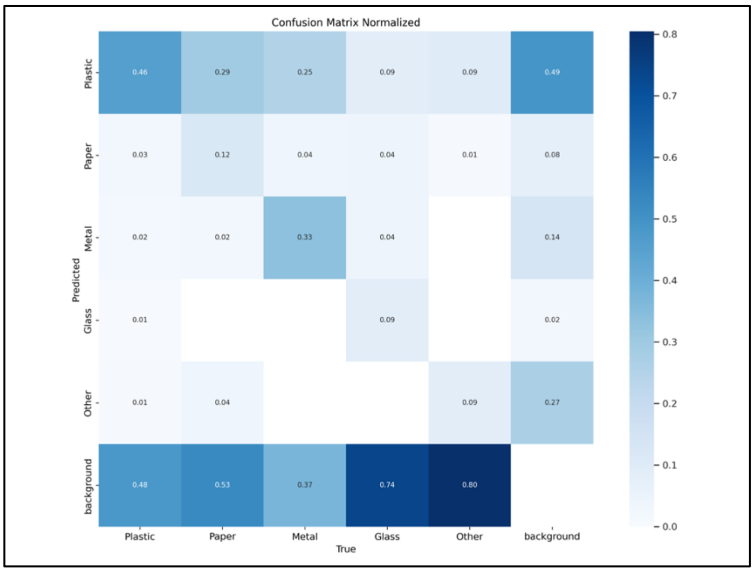
- The validation box loss has shown improvements, coherently with the training loss, which indicates that the model's ability to generalize bounding box predictions has enhanced.
- However, the classification loss on the validation set initially decreases but then shows a rising trend. This increase in validation classification loss could be a signal of overfitting, meaning the model is not generalizing well enough on unseen data.

Model Generalization: The increasing classification error on the validation set suggests possible overfitting issues, even with the use of an augmented dataset. The lack of overall significant improvement could be due to several reasons:

- Inherent Complexity: The complexity of the dataset, with small and often indistinguishable objects, limits the performance improvements.
- Hyperparameter Tuning: Finding the optimal set of hyperparameters is a complex process that often requires multiple iterations and it’s mainly based on a trial-and-error approach.
- Unbalanced Dataset: Oversampling underrepresented classes by replicating existing images may not sufficiently improve the model, since this method does not really introduce new information about the classes.

An important consideration must be done regarding the fact that the Validation set remained unchanged between the first and the second training. This set may not provide a fully accurate measure of the new model's performance across all classes due to its original bias towards the more represented classes which can lead to an underestimation of the model’s accuracy on minority classes. To achieve a more accurate evaluation, it would be better to rebalance the validation set to ensure the same proportions of classes as in the latest dataset.

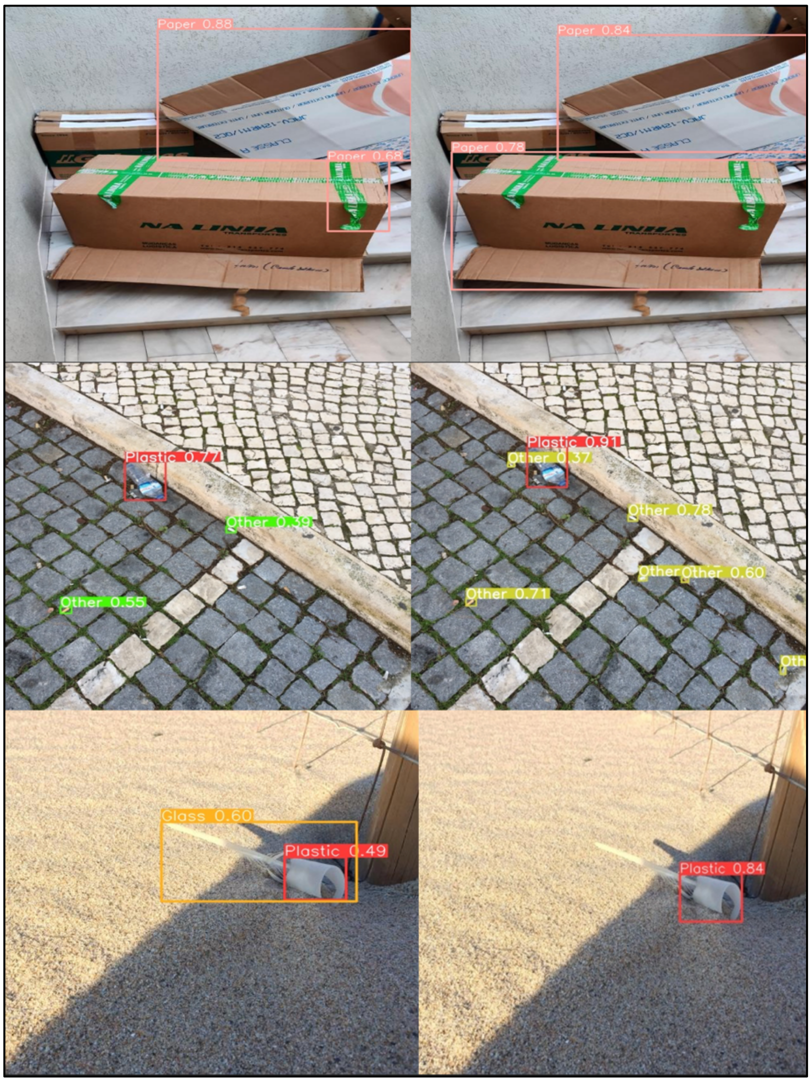
Figure8: Normalized Confusion Matrix for training 2



Overall, the adjustments made from the first training have brought the model to become better at detecting waste. However, achieving a balanced performance across all classes remains a challenging task that needs further strategic data augmentation, more sophisticated oversampling, or advanced training techniques.

4.5 Results 2

SOME OF THE BETTER RESULTS COMPARED TO MODEL 1:



For future enhancements of my project, several key improvements can be done, and I would personally like to work on them:

1. Data Enrichment: Acquiring a richer and more balanced dataset is essential. More data, especially for underrepresented classes, can help the model learn to better classify different classes.

2. Hyperparameter Optimization: Experimentation with different hyperparameters can lead to better model performance.

3. Diverse Augmentation Techniques: Exploring different and more sophisticated data augmentation techniques can increase the robustness of the model against real-world variations and also help with dealing with unbalanced dataset.

The main idea for the future would be to test the refined model through the innovative platform of AWS RoboMaker. The AWS RoboMaker simulation environment is perfect for integrating the trained YOLOv8 model into robotic systems programmed to recognize and sort various types of waste. The World generator feature of AWS RoboMaker allows for the creation of different simulation environments, which mimic many real-world scenarios. It also enables numerous simulations to run in parallel, which accelerates the model's refinement process and ensures its reliability before deployment in real-world robotic systems [8].

To test my model for potential integration into robots, I did some experiments using video inference from my phone camera, which showed promising results. The model was able to correctly identify different waste types in real time if the camera movement was not too rapid. This indicates that a robot equipped with a simple camera and this detection model could effectively distinguish various waste items. After successfully setting up the AWS RoboMaker environment, my future goal is to learn how to use the platform to perform simulations of my model on a robotic system.

5. Conclusions

In conclusion, this project is a beginning to address global waste management issues using deep learning, specifically with YOLOv8 model. Despite the unbalanced and complex dataset, I managed to make some improvements which allowed to enhance the model's ability to detect different types of waste. Through data augmentation and adjusting model settings, the model showed slightly better detection performance, still highlighting the need for a more balanced and richer dataset. The encouraging results from video testing show that, even with some accuracy limits, the model can effectively identify waste types in real time, proving its usefulness in automated waste sorting systems. The potential to integrate these models into robotic systems using platforms like AWS RoboMaker is the next step for getting closer to real-world applicability.

My project paves the way for future research and improvements, highlighting opportunities for richer datasets, better model adjustments, and advanced simulation tests. These efforts could lead to a cleaner future, where deep learning and robotic integration could play a crucial role in advancing environmental health.

References

[1] http://tacodataset.org

[2] United Nations Environment Programme, "Global Waste Management Outlook 2024", Report, 28 FEBRUARY 2024

[3] https://blog.roboflow.com/whats-new-in-yolov8/

[4] Mittal et al., "Application of Machine Learning Techniques for Waste Classification," Journal of Waste Management, 2019.

[5] Thung and Yang, "Deep Learning for Smart Waste Management Systems," Environmental Technology & Innovation, 2016.

[6] Sahani et al., "Integration of Robotic Automation and Deep Learning in Waste Sorting," IEEE Transactions on Sustainable Computing, 2020.

[7] Agnieszka Mikołajczyk, “Waste datasets review”, https://github.com/AgaMiko/waste-datasets-review

[8] https://docs.aws.amazon.com/pdfs/robomaker/latest/dg/aws-robomaker-dg.pdf