Recyclable Trash Detection: Handling Imbalances in DeepTrash Dataset

Nikil Posadi Sivanath Babu

Master of Science
School of Informatics
University of Edinburgh

2022

Abstract

The importance of recycling has only increased as time progressed. Creating systems that improve the accessibility of recycling is of great importance. DeepTrash is a private dataset procured by Danu Robotics by collecting real-world images of unsegregated garbage on the conveyor belt at a garbage segregation facility. The dataset contains 9618 photos and 44k annotations, with 5 different categories of recyclable waste. This dataset contains many imbalances such as the number of objects and image imbalance between classes, foreground-background and foreground-foreground imbalance. The objective was to implement an object detector that maximises the mean Average Precision while reducing the differences in classwise Average Precision.

The object detector VarifocalNet was chosen as it contains the loss function, varifocal loss which mitigates the issues of foreground-background and foreground-foreground imbalance. The addition of Auto Augment, the cosine annealing learning rate scheduler and pre-training on the TACO dataset provided improvements to the overall performance of the object detector. The augmentation of the size of the DeepTrash dataset led to the inference that certain object classes in the DeepTrash dataset require a greater number of images and annotations compared to others for the classwise AP to be similar among the object categories. The best performing object detector was achieved by modifying the varifocal loss hyperparameter gamma and training on a reduced DeepTrash dataset which contains 56.9% of images and 29.82% of the annotations. The mAP of this object detector was 53.3 which is 2.5 mAP higher than the baseline performance which was trained in the entire DeepTrash dataset. The classwise AP of this object detector is higher and consistent throughout most of the object classes. Additionally, the issues of missing labels and their effects are discussed.

In general, the research provides methods for improving the overall and classwise performance of the VarifocalNet object detector on the DeepTrash dataset and the limitations of the dataset.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Nikil Posadi Sivanath Babu*)

Acknowledgements

I want to start by expressing my gratitude to my family and friends for their unwavering support throughout the year. They continually kept an eye on me, read drafts, offered recommendations, and boosted my motivation.

I am thankful to Xiaoyan MA and Danu Robotics for allowing me to use the DeepTrash dataset throughout the dissertation. Furthermore, I want to express my gratitude to Muhammad Ahmed Raza and Tiago Lé for their continual advice, support, and insightful criticism. I want to thank my supervisor, Nigel Goddard, for his helpful advice on how to write the dissertation in the best way possible.

Finally, I would like to express my gratitude to The University of Edinburgh and everyone who contributed to its success for constantly motivating me to keep pursue new opportunities.

Table of Contents

Chapter 1

Introduction

The industrial revolution of the 18^{th} century [3], which sparked waves of global development, is notorious for its exponential resource consumption. This era saw an acceleration of globalisation and the emergence of various types of pollutants that contributed to the increase in global temperature. Since the pre-industrial times, the global temperature has risen by 1.5°C [19]. If global temperatures rise by another 0.5°C, the Earth's ecosystem might suffer catastrophic and irreparable damage, such as coral reef degradation, depletion of fisheries and food production damage, among other issues [10][35][13]. This has a direct impact on humans since it affects food supply. Due to the growing consumption of commodities as a result of this global economic growth, waste has become a problem that many developing nations are starting to grapple with on a major scale. The large-scale consumption produces rubbish that the planet cannot simply decompose. As a result, the development of garbage management systems has been eclipsed by the astronomical generation of waste. Rubbish on the streets, overflowing trash cans, waste in waterbodies like rivers and lakes, and garbage dumps located in and around major cities are common sights in many third-world nations. This leads to the spread of many illnesses in areas where most people travel, contaminating the resources available to the people in the region and increasing bug infestations that can spread many deadly diseases. This consequent decline in the standard of living that many of the country's citizens experience undermines the economic progress brought on by industrialisation.

The earliest waste management systems were developed at the dawn of the industrial age since it was an essential pillar for modern society to progress. This is when the first recycling practices were established to recover recyclable resources such as wood, metal, paper, etc [27][2]. The world wars of the 20^{th} century resulted in a material shortage and an economic crisis. To combat these issues, the first modern industrial recycling techniques were created [45]. Since the turn of the century, there has been a rise in environmental awareness and recycling has taken on more significance in today's socio-economic culture [11] [24]. Although recycling is beneficial to the environment and natural resources, the process is highly human resource intensive. Human labour is incredibly expensive; according to a CBSNewYork article, the city of New York pays $300 million per year to collect garbage [7] which is a staggering sum of money for many third-world countries. Thus, they implement subpar solutions for managing their waste. By significantly lowering the cost, automating waste management systems can contribute to their improvement. These enhancements strengthen the economic feasibility of recycling and assist in expanding its reach.

In recent years, the development in deep learning has increased drastically which has provided unparalleled improvement in computer vision. The introduction of the Convolutional Neural Network(CNN) [8], allowed itself to be used in various applications such as classification, object detection and segmentation. Object detectors are deep learning models which identify and locate an image. Such object detectors have been used for various tasks but for this dissertation, the task at hand is to identify and localise waste.

This dissertation focuses on implementing an object detector trained on the DeepTrash dataset. DeepTrash is a private dataset provided by Danu Robotics. The implementation used aims to maximise the mAP and the classwise AP of the object detector. The project used various techniques to improve the object detector and mitigate the imbalances present in the dataset. Through this approach, the methods that affect/improve the performance of the object detector are understood.

The structure is as follows: Chapter 2 provides some background on object detectors, trash object detectors and the two architectures that were implemented: RetinaNet and VarifocalNet. An overview of the supplementary datasets and the API used were provided. In Chapter 3, the methods for modifying the supplementary datasets, data analysis on the DeepTrash dataset, data augmentations, evaluation techniques, and lastly, how the object detector was trained were covered. Chapter 4 provides the DeepTrash data analysis, details of the experiments and results of the object detection training. Chapter 5 presents an overview of the findings, examines them, points out their shortcomings, speculates on potential future improvements and offers a conclusion.

Chapter 2

Background

2.1 Object Detectors

Computer vision, especially object identification, has advanced significantly as a direct consequence of the increased research into deep learning. Object detection, the extension of object identification, is performed by localisation following which a bounding box is drawn around the identified object. The object detectors are divided into two parts: two-stage and single stage[47]. Two-stage object detectors have separate modules for the identification of the image and localisation of the object. Since these systems have two stages, they are generally more complicated and slower. The single-stage system classifies and localises the objects in a single sweep using dense sampling[30]. They use bounding boxes present in the labelled data to localise the objects. These are generally slightly quicker and simpler compared to the two-stage systems.

Two-stage: Two-stage detectors consist of various architectures. The Region-based Convolutional Neural Network (R-CNN) [15] was one of the first systems to demonstrate the use of CNNs for object detection. SPP-net [16] used the Spatial Pyramid Pooling (SPP) layer to process images of any size and ratio. R-CNN was a revolutionary technology, but it was very slow. SPP-net improved the speed of the system greatly. Owing to the fact that it can accept an image of any size/ratio, it reduced the artefacts caused by input deformation. Fast and faster R-CNN [14, 33] reduced the complexity of training compared to the R-CNN and SPP-net. The speed of the system increased dramatically while the accuracy improved slightly. Mask R-CNN [18] improves on Faster R-CNN by adding another parallel branch for pixel-level object instance segmentation. This helped this system outperform all existing state-of-the-art object detectors at the time. It also introduced the additional functionality of image segmentation.

Single stage: The two-stage system solved the problem of object detection by separating it into two main tasks; object identification and localisation. Here, You Only Look Once (YOLO) [31], a single-stage detector re-framed this problem by making this a regression problem and directly predicting the location of the bounding boxes. When it was first released, YOLO outperformed any single-stage detector, but it had several flaws. The main disadvantage was the loss of localisation accuracy in small and clustered objects. The Single Shot MultiBox Detector (SSD) [25] was built on VGG-16 [37] with additional layers to improve performance. SSD became the first single-stage detector that matched the state-of-the-art two-stage detectors in terms of performance metrics. Even though SD outperformed YOLO and faster R-CNN in terms of speed and accuracy, it still had several drawbacks. The object detector had difficulty predicting smaller objects in the image. The future versions of YOLO such as YOLOv2, YOLOv3 [32], and YOLOv4 [6] improved on the initial version in terms of speed, accuracy and solved the main issues YOLO was facing. CenterNet [51] proposed a very different approach by taking the objects as points instead of the traditional bounding boxes. This change increased the accuracy in various tasks like 3D object detection, keypoint estimation, pose and instance segmentation. CenterNet's drawback, however, is that because of its distinctive backbone, it is challenging to integrate it with other systems and backbones, which leads to poor performance. The introduction of RetinaNet [21] brought in a new loss function called focal loss. In comparison to two-stage detectors, the system showed increased accuracy and speed after the addition of this loss function and the ResNet [17]backbone. RetinaNet is also easy to implement, train and converge faster.

The performance of object detection/classification has increased dramatically in recent years. Many object detection systems are utilised for applications requiring real-time detection. Due to their slow nature and complexity during training, two-stage detectors are not suitable for such applications. However, more systems are being adopted for real-time detection with improvements in single-stage detector accuracy.

2.2 Trash Object Detectors

Object detection systems can be used for varying tasks. The objective at hand is to identify and locate the garbage in the image. Object identifying/detecting systems based on convolutional neural networks have been employed extensively in current work on detecting waste in images. Since deep learning object detectors can be trained on a different dataset to adapt to a certain task, the open-source dataset TrashNet [46] was used. Despite being classification-based, earlier works nonetheless provide a wealth of information regarding useful techniques [5, 38]. [44] has used the TrashNet dataset [46] but also implemented localisation. This was implemented by using a CNN and a gaussian clustering method to locate the trash on the image. [4] proposed a method with the use of an R-CNN.[46] used an augmented version of the dataset. They augmented the dataset with annotations and increased its size to 10000 images. [39] implemented a method of trash detection by using systems such SSD[25] ,YOLO-v3 [32],YOLO-v3-Tiny [1] and PeeleNet [42]. The dataset used was a custom dataset consisting of 30 videos each of 60 minutes. As a result, they were able to obtain more than 48k objects. These were divided into 3 different sizes (small, medium, and large). While there are some limitations, such as the inability to identify small objects in the image, [39] demonstrates the success of this object detector. They overcame the issues by proposing a new log-based layer which improves the object detector's performance on small objects.

2.3 MMdetection

MMdetection is an object detection and image segmentation toolbox [9]. It is a toolbox that runs on the python API pytorch [29]. This toolbox provides a unified platform for training, evaluation and testing. The major advantages of MMdetection include modular design, multiple framework compatibility, native support for graphics processing units (GPUs) for increased efficiency and frequent framework updates. This toolbox provides the frameworks and weights for over 200 popular one-stage, two-stage and multistage models. The architecture of the models in the toolbox is represented using the backbone, neck, densehead and RoIExtractor. Here, the backbone (e.g. ResNet) is the main body of the network which extracts features while excluding the last fully connected layer. The neck (e.g. Feature Pyramid Network) is the part that connects the head to the backbone and performs refinements on the raw feature mappings. The densehead (e.g. RetinaHead) is the part that locates the objects in the image and provides dense locations of the feature maps. The RoIExtractor is the component that uses RoIPooling-like operators to extract RoIwise features from single or multiple feature maps. The RoIHead is the part that takes RoI features as input and makes RoI-wise task-specific predictions, such as bounding box classification/regression and mask prediction. MMdetection also has support for scaling with multiple GPUs. The scalability and adaptability of the system allow for the adoption and customization of a wide range of frameworks to satisfy the needs and expectations of the user.

2.4 RetinaNet

RetinaNet is a one-stage object detector that uses a novel loss function called the focal loss to address the classwise imbalances during training [21]. RetinaNet is a single, integrated network made up of two task-specific subnetworks and a backbone network. The backbone, which is an off-the-self convolutional network, computes a convolutional feature map over the whole input picture. On the output of the backbone, the first subnet applies convolutional object classification and the second subnet applies convolutional bounding box regression. The authors' straightforward architecture for the two subnetworks is intended primarily for dense one-stage detection. The new focal loss is inspired by two-stage object detectors, in which the imbalance is addressed utilising two-stage cascade and sampling heuristics. To keep the foreground and background in a tolerable balance, sampling algorithms like a fixed foreground-to-background ratio or Online Hard Example Mining (OHEM) [36] are used in the second classification step. The collection of potential object locations that must be processed by a one-stage detector is substantially greater and is frequently sampled from all around an image. RetinaNet utilises a focal loss function, a cross-entropy loss that is dynamically scaled, to address this issue. As confidence in the correct class improves, the scaling factor decays to zero. Intuitively, this scaling factor can quickly focus the Object Detector on difficult cases while automatically de-weighting the contribution of simple examples during training. In the dissertation, the RetinaNet is used from MMdetection. The backbone is the X-101-64x4d-FPN. The weights used for training will be the same as the weights from the MMdetection [22], which was obtained from training on COCO [23].

2.5 VarifocalNet

VarifocalNet (VFnet) [49] is a dense object detector which consists of a new loss function called the Varifocal loss. It uses an IoU Aware Classification Score (IACS) as the representation of both presence confidence and localisation accuracy. The Varifocal loss is used to predict the IACS. For IACS prediction and bounding box refining, [49] suggested a star-shaped bounding box feature representation. VFnet is based on the architecture of FCOS [40] and ATSS [50]. Fully Convolutional One-Stage Object Detection (FCOS) [40] is an anchor-box free, proposal free, single-stage object detection model. By eliminating the predefined set of anchor boxes, FCOS avoids computation related to anchor boxes such as calculating overlapping during training. Adaptive Training Sample Selection (ATSS) [50] is a method to automatically select positive and negative samples according to the statistical characteristics of the object. It bridges the gap between anchor-based and anchor-free detectors. Varifocal loss takes inspiration from focal loss. The difference is that focal loss treats all negative and positive examples equally, while varifocal loss treats them asymmetrically. This is done to preserve the signals from the positive examples as they are rare. The performance on the standardised COCO test-dev is cutting-edge, with a score of 51.3 AP. This object detector from MMdetection is utilised in the dissertation. The backbone used is the R-101 [17] with DCN [43]. The weights used are the same as the weights acquired via MMdetection [48] through training on COCO [23].

2.6 Datasets

2.6.1 TACO

The Trash Annotations in Context for Litter Detection (TACO) [28] is an open-source dataset that contains images of various types of garbage in real-world scenarios. These are captured mainly using mobile phones and stored on Flickr. The dataset contains high resolution RGB images. There are 60 categories and 28 super categories. In the dataset, there are 1500 images with 4784 annotations. These annotations are in the COCO format. The images of this dataset contain multiple object classes. An example of an image belonging to this dataset with its annotations is shown in the figure 2.1.

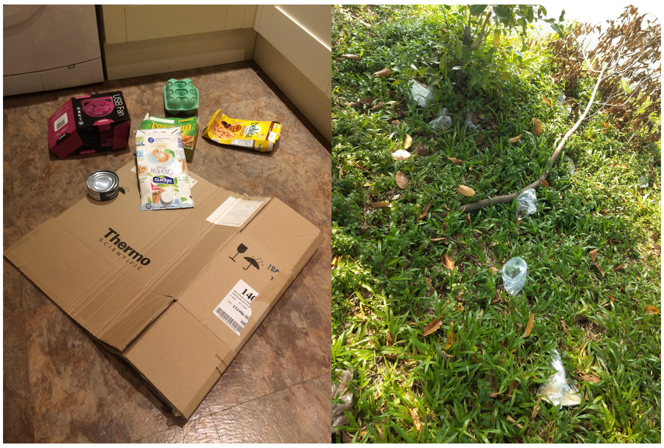


Figure 2.1: Examples of TACO Images

2.6.2 TrashNet

TrashNet [46] is an open-source dataset. This dataset contains images of glass, paper, cardboard, plastic, metal and trash. The images of this dataset were captured using Apple iPhone 7 Plus, Apple iPhone 5S and Apple iPhone SE. The dataset contains 2527 images split into 501 glass, 594 paper, 403 cardboard, 482 plastics, 410 metal and 137 trash images. The resolution of the images was downsized from 4034x3024 to 512x314. This dataset was not created for object detection but rather for the classification of images. Thus, none of the images is annotated with the objects, rather just split into different categories. Examples of images from this dataset are shown in the figure 2.2.

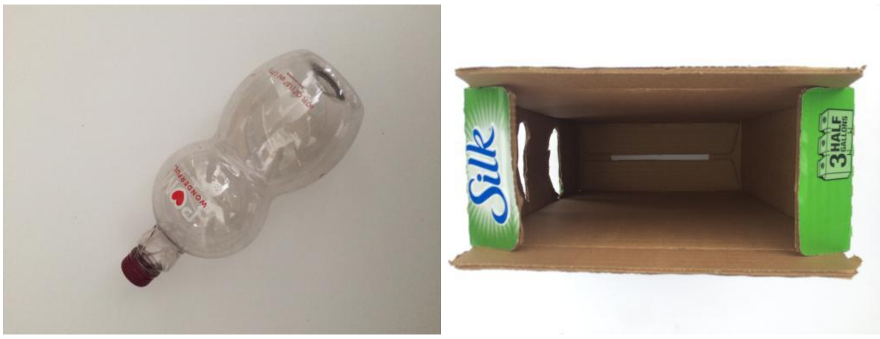


Figure 2.2: Examples of TrashNet Images

Chapter 3

Methods

3.1 Data Analysis

The dataset used to train and test the object detectors has several characteristics. Understanding these characteristics will help create a more tailored object detector for various tasks. This dataset has a variety of features, such as the ratio of foreground to background, the number of objects per class in the categories, the number of objects per image and each category, the size of the objects and their location. Many imbalances caused by these qualities, such as class, scale and spatial or objective imbalance could potentially cause the performance to drop [26]. The dataset analysed is the DeepTrash[1] collected by Danu Robotics. This is a private dataset. Its intellectual property is owned by Danu Robotics.

3.2 Datasets

3.2.1 TACO

The TACO [28] dataset has 60 categories which cover a wide variety of waste categories. Some of the categories from the TACO dataset are similar to the ones present in the DeepTrash dataset. These categories are Other plastic bottles, Clear plastic bottle, Other plastic cup for the class Plastic Juice Water Bottle; Egg carton, Drink carton,

[1] The intellectual property of the DeepTrash dataset is owned by Danu Robotics and is copyright protected. The dataset will not be publicly released and all the photos are used for demonstration purposes only. Any use of the under-discussion images in research or industrial is illegal.

Meal carton, Pizza box for Paper Cardboard Container; Plastified paper bag, Single-use carrier bag, Polypropylene bag for Plastic Shopping Bag; Other carton, Corrugated carton for Cardboard and Magazine paper, Normal paper for Paper Newspaper. The category labels for the images and annotations have been changed to match those in the DeepTrash dataset. This allows the use of additional images from the TACO dataset for training the object detector.

Category	Number of images	Number of objects
Plastic Juice Water Bottle	388	566
Paper Newspaper	71	94
Cardboard	120	162
Paper Cardboard Container	76	89
Plastic Shopping Bag	52	64

Table 3.1: Number of images and objects in modified TACO

3.2.2 TrashNet

The TrashNet [46] dataset was created with the exclusive purpose of classifying garbage, as opposed to both classifying and locating the trash. This results in the dataset having images with only its category defined. To convert this dataset into one which can be used for training an object detector, the categories relevant to the DeepTrash dataset were chosen. These categories were cardboard, paper and plastic. These images were then annotated in the pascal VOC format[12] using labelImg. LabelImg [41] is a graphical image annotation tool. It annotates the images in the format of pascal VOC. This pascal VOC annotation file cannot be used with the DeepTrash dataset or TACO as they both have annotations files in the COCO format [23]. To convert the pascal VOC annotation to COCO, a tool called roboflow [34] was used.

3.3 Evaluation

The evaluation of the object detector is conducted at both the training and testing stages. The metrics used are the same for both stages. The metrics used are Mean Average Precision (mAP) and Average Precision (AP). Intersection over Union (IoU) is the ratio

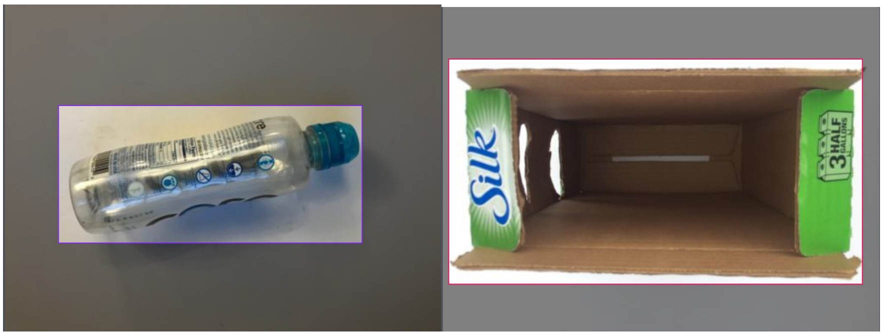


Figure 3.1: Examples of TrashNet Images Annotated

of overlap between the predicted and ground truth bounding box (bbox). A threshold is used to determine whether the prediction is either a true positive or a false positive. If no object is detected, then it will be considered a false negative. Using the equations 3.1, 3.2 and the threshold for IoU, the precision for each class is calculated. For the comparison of performance between the various detectors, both the average precision per class and the mean average precision (mAP) for all the classes are used. The Average Precision (AP) is the area under the precision-recall curve, calculated using the formula 3.3, where n is the number of IoU thresholds , $Recall(n) = 0$ and $Precision(n) = 1$. The mean average precision is the mean of AP when the AP is calculated for each object class. The formula for mAP is shown in 3.4, where AP_c is the AP of class c and n_c is number of object classes.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.2}$$

$$AP = \sum_{k=0}^{k=n-1} [Recall(k) - Recall(k+1)] * Precision(k) \tag{3.3}$$

$$mAP = \frac{1}{n} \sum_{c=1}^{c=n_c} AP_c \tag{3.4}$$

The evaluation method used for this project is the COCO evaluator [23]. This was selected since it is the most widely used evaluation system for object detectors. This method has also been preferred over the pascal VOC system as the COCO evaluator calculates the Average Precision(AP) over 10 IoU thresholds across all the object classes. This system has two main metrics for AP; the AP across different IoU and the AP across different scales. The AP across the IoU range consists of the bbox mAP which measures the AP through the IoU thresholds of 0.5 to 0.95 with a step of 0.05. The AP at 0.5 IoU is the Pascal VOC metric while the AP at 0.75 IoU is the strict metric. The AP across scales is divided into three measurements: small, medium and large. Small are objects with an area less than 32^2, medium refers to objects with an area between 32^2 and 96^2 and large is for objects with an area greater than 96^2.

3.4 Data Augmentations

3.4.1 Auto Augment

Google created a set of data augmentation policies called ”Auto Augment” to enhance object detection performance [52]. The policies include various augmentations that apply to colour, geometry and bounding boxes. This implementation for data augmentations was used at it resulted in an improvement of 2.3 mAP on the COCO dataset using the ResNet 50 backbone. The improvement was also carried over when the dataset was switched from COCO to pascal VOC. The sole drawback of this approach is that the COCO dataset is substantially larger than the DeepTrash dataset. The PyTorch implementations were used, which have all the augmentations used in the training of the COCO dataset [23].

3.4.2 Augmenting the Number of Objects in Dataset

The imbalances in the DeepTrash dataset can lead to a lot of undesirable results. Some of these imbalances can be artificially reduced to understand their impact. The DeepTrash dataset was first reduced such that all the object categories have a similar number of images and objects per image. The number of images was set per the object category with the fewest images/annotations. The annotations of the dataset were randomly chosen with a probability that depends on the ratio of the images of the desired object category and the lowest object category mentioned in equation 3.5, where N_f is the number of annotations from the object class with the fewest annotations and N_k is the number of annotation from the class k. Using this method, the number of images and objects per image remains consistent throughout the dataset. The dataset was also modified to keep the number of images in a similar distribution to the original dataset. The object per image also was kept consistent with the original dataset. The second modification was carried out in the same manner as the first, but this time all of the objects in the image of the randomly selected annotation's category were added.

$$Probability = \frac{N_f}{N_k} \tag{3.5}$$

3.4.3 Mixing datasets

The datasets were mixed using two methods:

1. The object detector was trained on a different dataset before being trained on the DeepTrash dataset
2. The object detector was trained on both the DeepTrash and supplementary datasets

The dataset used in the former method of pretraining the object detector is the TACO dataset. Compared to the DeepTrash dataset, the TACO dataset has a large number of trash categories, some of which are also present in the TACO dataset. This dataset was chosen as it has a variety of objects in an image. The images are filled with multiple objects of different categories. The area of the images is also well distributed. This dataset provides a good platform for the object detector to learn to identify and locate the objects. The second method uses the TACO and TrashNet datasets. Here, the datasets are modified as detailed in section 3.2. These datasets are used as supplementary datasets with the DeepTrash to train the object detector. The supplementary datasets are only used when training while the DeepTrash validation dataset is used for validation.

3.5 Training

The object detectors were trained using the MMdetection[9] toolbox. The architectures and weights of the object detectors were used from MMdetection. The weights for the object detectors from MMdetection are those obtained after training these object detectors for 100 epochs on the COCO dataset. The dataset for training the object detectors is split as follows: 80% training, 10% validation and 10% testing. The dataset is split using the tool pycocosplit[20]. Pycocosplit is a program which divides the annotation files into the desired train, validation and test splits. All the object detectors make use of the vanilla frameworks, hyperparameters, weights and datasets unless specified. The object detectors were trained for 25 epochs, where validations occur after every epoch. The training of these object detectors was exclusively performed on

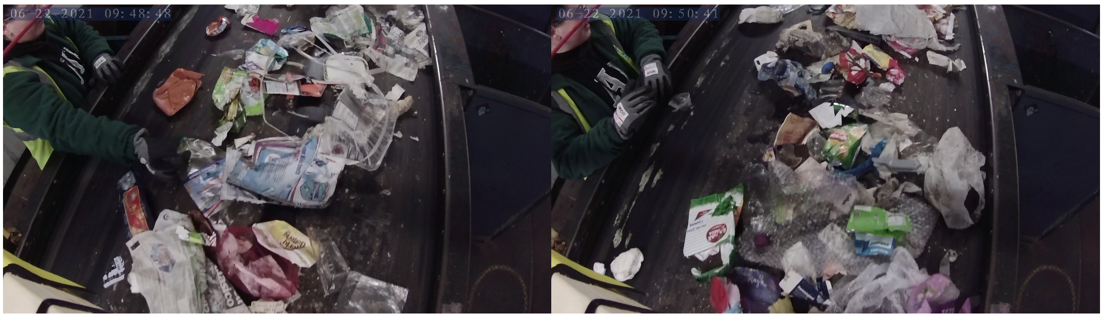
GPUs. The GPUs used were the RTX 2080Ti and the RTX 3070. Two different GPUs were used due to their varying amounts of video memory (11GB and 8GB). This was implemented because some RetinaNet and VarifocalNet object detectors require more than 8GB of VRAM.

Chapter 4

Experiments and Results

4.1 Data Analysis

The DeepTrash[1] dataset, collected in a real-world setting by Danu Robotics, has images of unsegregated garbage on the conveyor belt captured at a garbage segregation facility as seen in the figure 4.1. The dataset contains 9618 photos and 44k annotations in RGB and 1920x1080 resolution. The dataset contains 5 different classes: Plastic Juice Water Bottle (PWB), Paper Newspaper (PN), Plastic Shopping Bag (PSB), Cardboard (C) and Paper Cardboard Container (PCC). Each image belonging to this dataset can have 1 or multiple classes present as seen in the figure 4.1. The objects in the image are annotated using the COCO format [23].



(a) (b)

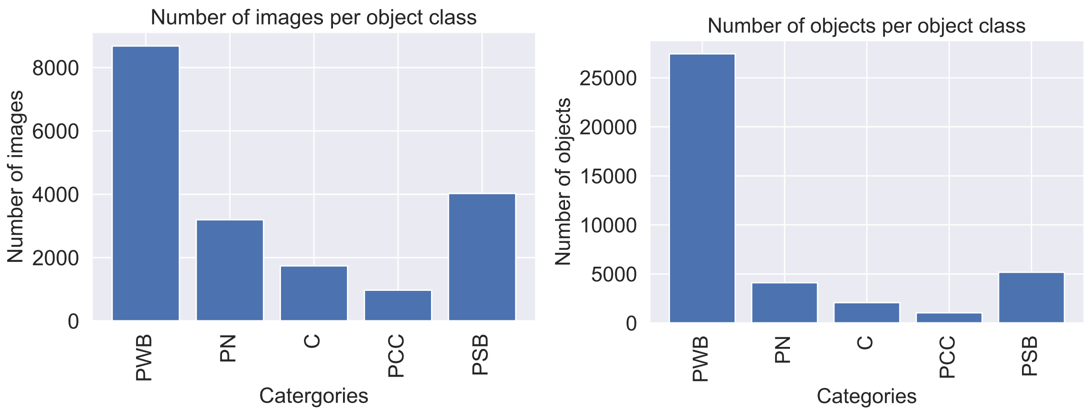
Figure 4.1: Examples of DeepTrash Images

[1] The intellectual property of the DeepTrash dataset is owned by Danu Robotics and is copyright protected. The dataset will not be publicly released and all the photos are used for demonstration purposes only. Any use of the under-discussion images in research or industrial is illegal.

4.1.1 Images and Objects Per Class

Analysing the images per class is one of the simplest techniques to understand the distribution and representation of object classes. This method of analysis is used widely for classification tasks as it is the easiest property of the dataset to analyse. From the figure 4.2a, it is seen that the DeepTrash dataset has a severe over-representation of the Plastic Juice Water Bottle object class. This is eight times larger than the object class of Paper Cardboard Containers, which is the least represented object class.

The distribution of the number of objects between the object classes is another method to understand the dataset's properties. Figure 4.2b shows the disparity between the Plastic Juice Water Bottle object class and the other classes. Compared to the worst represented class of Paper Cardboard Container, the Plastic Juice Water Bottle has a much higher degree of representation which is around 25 times more. The ratio between the Plastic Juice Water Bottle and other object classes has also increased. This can easily cause the object detector to overfit the Plastic Juice Water Bottle object class, which can cause a loss in the training performance.



(a) Distribution of Images

(b) Distribution of Objects

Figure 4.2: Distribution of objects and images

4.1.2 Foreground-Background Analysis

This analysis will focus on the difference between the foreground and background of the image. The foreground of the image is where the object in the image lies. For example in the case of the DeepTrash dataset, the foreground consists of objects from the classes Plastic Juice Water Bottle (PWB), Paper Newspaper (PN), Plastics Shopping Bag (PSB), Cardboard (C) and Paper Cardboard Container (PCC). The background of the image is everything that doesn't consist of the object that is being detected. The ratio of the foreground vs background is calculated by the area occupied by each of the classes in the image. The total area of the image for the calculation used the resolution of the image. Since the annotations are in the COCO format, the values from the bounding boxes of the object are used. The bounding box is in the format (top left x coordinate, top left y coordinate, width, height). The area of the object in the image is calculated from the annotations. The background is calculated by the difference between the total area of the image and the object area.

Table 4.1: Ratio of Background to foreground area

Categories	Ratio
PWB[2]	21.465
PN[3]	14.002
C[4]	14.208
PCC[5]	37.481
PSB[6]	7.943
Whole dataset	7.089

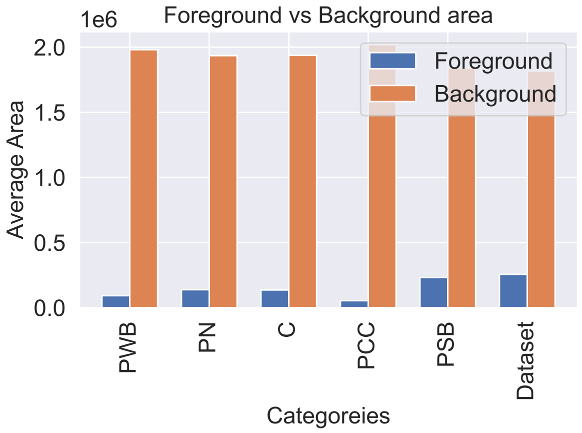


Figure 4.3: Ratio of foreground vs background

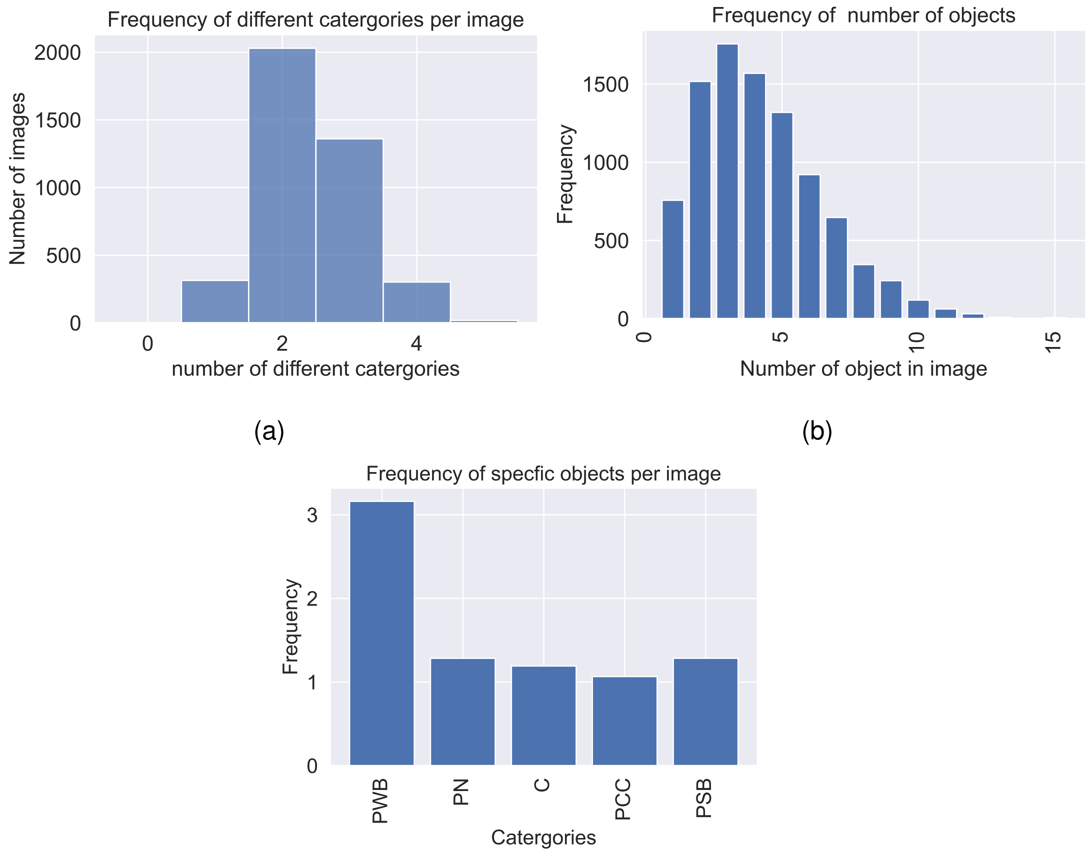
From the figure 4.3, it is seen that there is a significant disparity between the foreground and background classes. The ratio of the background to the foreground is shown in table 4.1. Here, it is seen that the ratio is ≈1:25. This is comparably low to the issue discussed in the focal loss [21] where the ratio was in the values of 1:1000. This is substantially lower, yet it could cause the background to be over-represented.

4.1.3 Foreground-Foreground Analysis

Here, the focus is on the foreground objects in the image. The analysis compares the representation of different foreground object classes. The figure 4.4c shows the frequency of the foreground classes in the image. This shows a huge disparity between

[2] Plastic Juice Water Bottle
[3] Paper Newspaper
[4] Cardboard
[5] Paper Cardboard Container
[6] Plastics Shopping Bag

the foreground class Plastic Juice Water Bottle and the rest of the classes. The Plastic Juice Water Bottle occurs multiple times in the images and has the most number of images as depicted in figure 4.2a



(a)

(b)

(c)

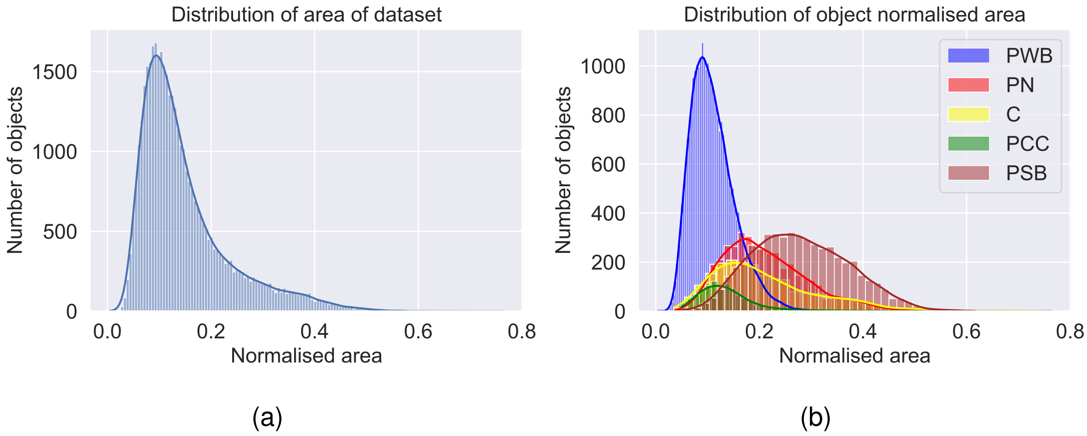
Figure 4.4: Foreground-foreground analysis

Another important property to analyse is the number of different classes present in the images. The figure 4.4a shows that majority of the dataset has 2-3 different foreground classes present in them. This is beneficial since it allows the object detector to detect several object classes in the image. The distribution of the number of objects in an image seen in figure 4.4b is also varied, which helps the object detector identify multiple objects.

4.1.4 Location and Area of the Objects

4.1.4.1 Distribution of Area

This section focuses on the distribution of the size of the objects across different categories and the dataset under study. The area is calculated from the bounding box value from the annotations. This analysis produces different distributions of object sizes throughout the dataset as seen in figures 4.5a and 4.5b. The table 4.2 shows the number of objects in the three different categories (small, medium and large). As mentioned in section 3.3, the COCO evaluator measures the performance concerning object sizes as well. From table 4.2, majority are classified as large objects. Thus, from the results, only the metric for the large objects should be considered.



(a)

(b)

Figure 4.5: Distribution of area (classwise and whole dataset)

Categories	Small	Medium	Large
Plastic Juice Water Bottle	4	3688	23750
Paper Newspaper	0	24	4078
Cardboard	0	36	2037
Paper Cardboard Container	0	35	1001
Plastic Shopping Bag	0	6	5163
Whole dataset	4	3789	36029

Table 4.2: Distribution of area in the dataset

4.1.4.2 Locations

The locations of the objects are also important properties because sliding window classifiers in modern deep object detectors use densely sampled anchors. A large number of present-day object detectors uniformly distribute the anchors across the image, giving each component of the image the same weight of importance [26]. For each occurrence of an object in an image, the value in its corresponding location in a 1920x1080 matrix is incremented by one. This then makes up a 2D heatmap of the image showing the frequency of object occurrence for every pixel.

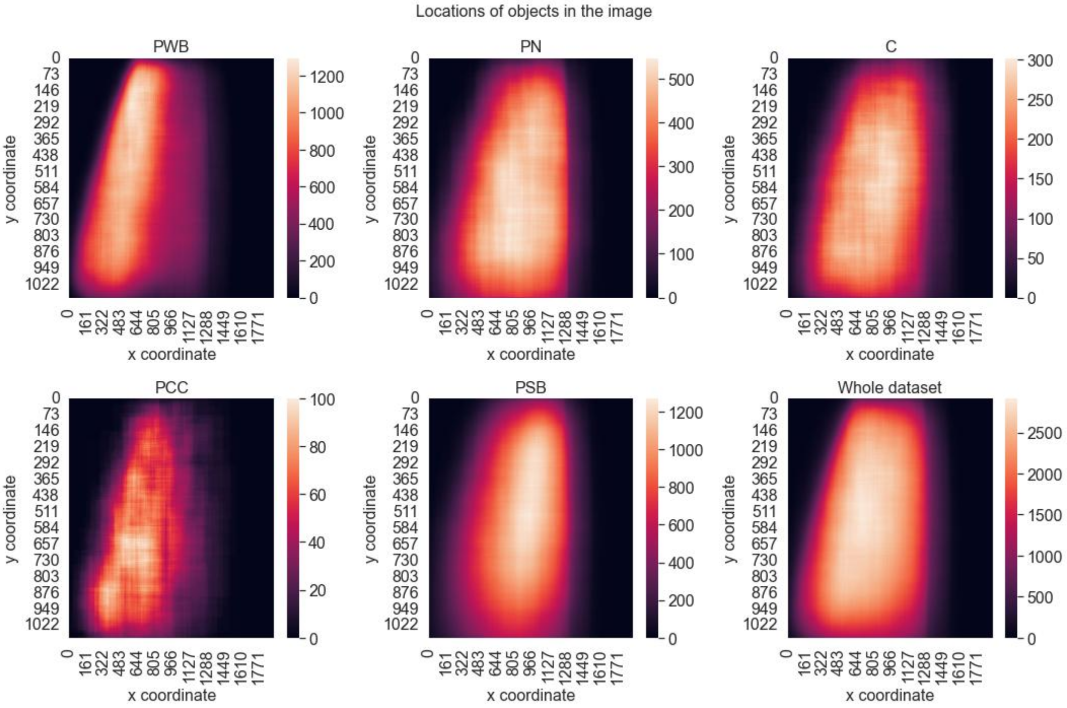


Figure 4.6: Heatmap of object location in an image

From figure 4.6, it is seen that most of the objects occur around a similar space in the image. This is understandable as the image contains a conveyor belt filled with trash. Since most of the heatmaps are evenly distributed, there is no specific location where a certain object category occurs more frequently than the others.

4.2 Baselines

To get baseline values for the classes and dataset, the object detectors were initially trained on the vanilla DeepTrash dataset. The object detectors trained on the dataset were RetinaNet and VarifocalNet. The object detectors were trained for 25 epochs using the vanilla framework, hyperparameters and weights.

	Category	PWB[7]	PN[8]	C[9]	PCC[10]	PSB[11]
AP	Vfnet	0.625	0.513	0.411	0.398	0.595
AP	RetinaNet	0.377	0.204	0.135	0.101	0.274

Table 4.3: Baseline results for Classwise Average Precision

	bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
Vfnet	0.508	0.661	0.562	0.516
RetinaNet	0.218	0.367	0.225	0.223

Table 4.4: Baseline results for mAP values

From the results in tables 4.4 and 4.3, it is seen that the imbalance in the data has impacted the classwise results directly. The classwise AP values directly correlate to the number of images and objects. The foreground-foreground imbalance could also play a role in influencing accuracy as the number and frequency of objects of Plastic Juice Water Bottle is much higher than any of the other object classes as seen in the section 4.1.3. The vanilla VarifocalNet object detector uses the step learning rate scheduler.

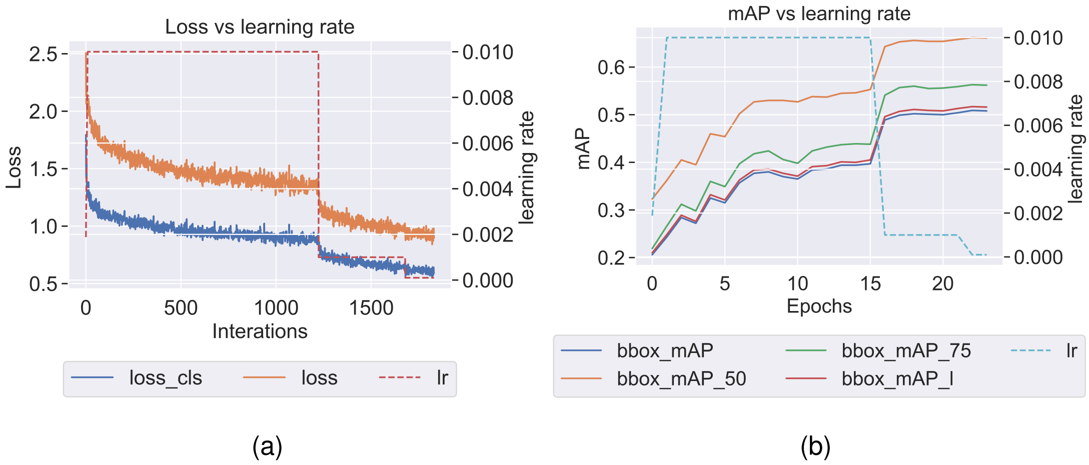


Figure 4.7: Baseline loss and mAP vs learning rate

[7] Plastic Juice Water Bottle
[8] Paper Newspaper
[9] Cardboard
[10] Paper Cardboard Container
[11] Plastics Shopping Bag

This is when the learning rate is reduced by a multiplier at a step. In the figures 4.7a and 4.7b, there is an observable stagnation in the loss and mAP values. This could be caused by the step learning rate scheduler. The constant learning rate caused the object detector to get stuck in a local minima. This can be viewed as a correlation between changes in loss and mAP values and changes in learning rate. The object detector can be enhanced by changing the learning rate scheduler so that the object detector's loss and accuracy do not regularly stagnate at local minima.

Inference Testing

The figures 4.8a and 4.8b show the output of the object detector with the class prediction and confidence score. In the graph, the distribution of confidence scores for the detection of objects of different classes is shown in 4.8a. It shows the distribution over the whole confidence score range. Here, a huge number of objects are detected in the confidence score range of $0-30\%$. This is expected due to the imbalance in the foreground-background and foreground-foreground imbalance. To get a better idea, the figure 4.8b represents the distribution of objects with a confidence score of $> 30\%$. This threshold was chosen as through visual inspection of 200 random images, the majority of the objects under consideration were identified. It can be observed here that the detector is learning well, with a majority of the Plastic Juice Water Bottle objects being predicted with a high degree of confidence. The other classes show a similar distribution but not at the same magnitude.

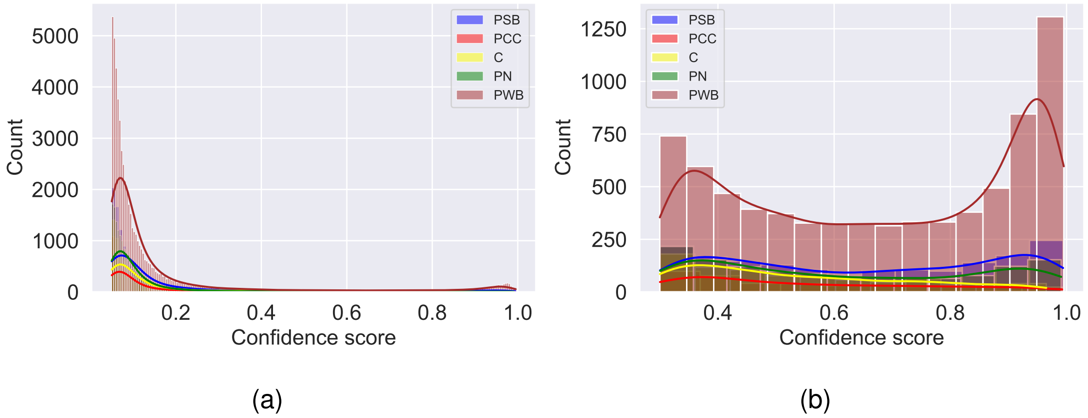
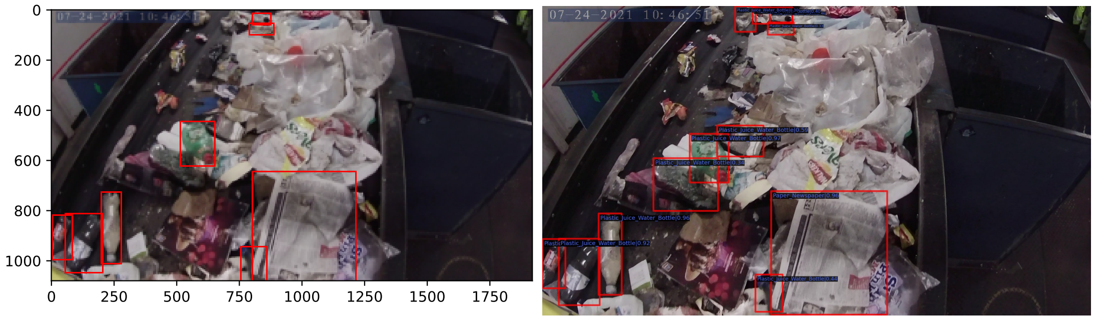


Figure 4.8: Baseline Histogram of Confidence Scores

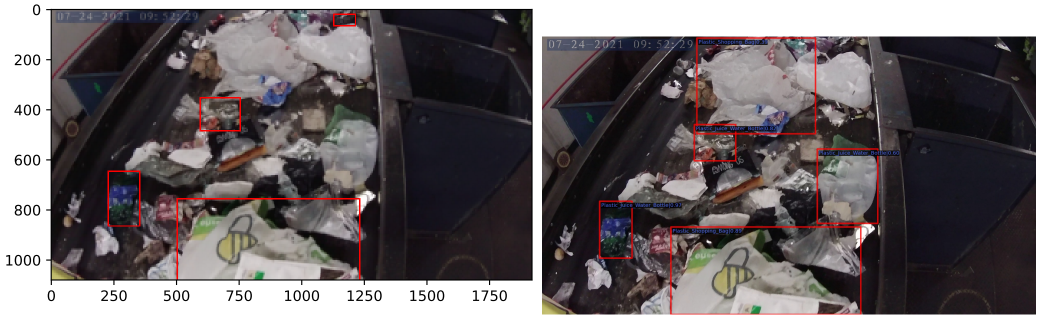
In the figures B.7 and 4.10, the issue of missing labels is observed. These figures are the output of the object detector with a confidence threshold of 30%. The figure 4.9b with the predicted objects from the object detector shows more objects than what is labelled in the data. This is fairly consistent throughout the dataset, where the object detector usually locates more objects than what has been labelled. Through visual inspection of 200 images, the prediction of these additional objects is accurate majority of the time. The object class that is most affected by this is the Plastic Juice Water Bottle. While it isn't as common as with the Plastic Juice Water Bottle class, the other classes also have extra objects that are predicted. Visual inspection reveals that while the majority of objects are correctly predicted, the object detector occasionally has trouble differentiating between Plastic Shopping Bags and Cardboard. The data's missing labels mean that the mAP measures observed should be interpreted cautiously due to the increase in the number of false positives, thus reducing the precision as well as Average Precision. The VarifocalNet object detector performance is significantly better, both



(a) Ground Truth

(b) Predicted Output

Figure 4.9: Example 1 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

Figure 4.10: Example 2 of comparing the ground truth to inference

in terms of absolute and relative performance. The relative performance is the relative difference of mAP between the categories. The VarifocalNet and RetinaNet training times for the entire DeepTrash dataset were 25 hours and 45+ hours, respectively.

Only the VarifocalNet will be tested with the modifications because it outperforms the RetinaNet significantly in terms of metrics and training times.

4.3 VarifocalNet Auto Augment

This experiment focuses on the impact of the Auto Augment data augmentation technique on the VarifocalNet object detector. The baseline methodology, in conjunction with the Auto Augment method, is used to train the VarifocalNet Object detector.

Category	PWB	PN	C	PCC	PSB
AP	0.627	0.515	0.413	0.403	0.600

Table 4.5: Auto Augment results for Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.512	0.665	0.568	0.519

Table 4.6: Auto Augment mAP values

The results shown in the tables 4.6, 4.5 indicate that the additional Auto Augment technique has a very minor effect on performance for this task. Thus, adding the Auto Augment technique does not positively affect the object detector according to the metrics observed.

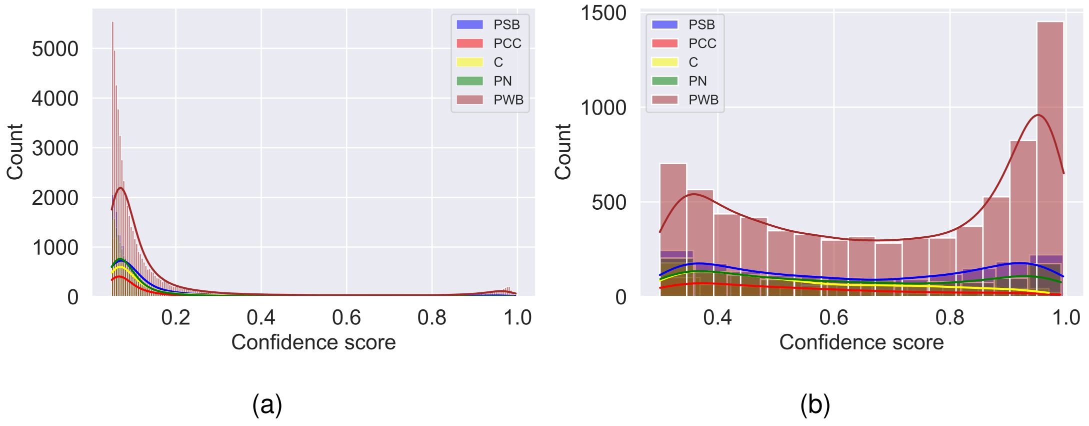


Figure 4.11: Auto Augment Histogram of Confidence Scores

Comparing the figures 4.11b and 4.8b, there is an improvement in what the object detector classifies as ’easy’ objects. ’Easy’ objects are the objects with high confidence scores. As a result, using the Auto Augment technique has advantages that allow it to be applied in the experiments that follow. The missing labels of the DeepTrash dataset hide the improvement in the mAP metrics. Thus, many of the mAP metrics are difficult to compare.

4.4 VarifocalNet Learning Rate Scheduler

The baseline experiments used the step learning rate scheduler which was changed in an effort to address the problem that was observed in the baseline results. The learning rate scheduler was altered from step to cosine annealing. This was implemented because the baseline training of the VarifocalNet object detector had quickly reached a local minima for both the mAP values and loss values before the learning rate was updated. The new learning rate schedule maintains a similar AP on the object class Plastic Juice Water Bottle while improving the AP of other object classes and the overall mAP values as seen in tables 4.7 and 4.8.

Category	PWB	PN	C	PCC	PSB
AP	0.620	0.519	0.431	0.415	0.613

Table 4.7: Cosine Annealing learning rate Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.518	0.67	0.575	0.526

Table 4.8: Cosine Annealing learning rate mAP values

In the figure 4.12, the impact of changing the learning rate scheduler is observed. The loss or mAP values do not stagnate around a local minima due to the constant updating of the learning rate values. This change was effective since the overall loss values at the end of training were lower than the baseline results. In the figure 4.12, there is constant improvement in the mAP values throughout the training period, although the rate of increase starts to reduce as the training period approaches 25 epochs. The distribution of the confidence score has also seen an improvement. From figure 4.13, there is a shift in the distribution in the confidence score towards the right. In the figure 4.13b, there is an increase in the objects detected with high confidence for the class Plastic Juice Water Bottle (PWB). The rest of the classes show a similar improvement in distribution

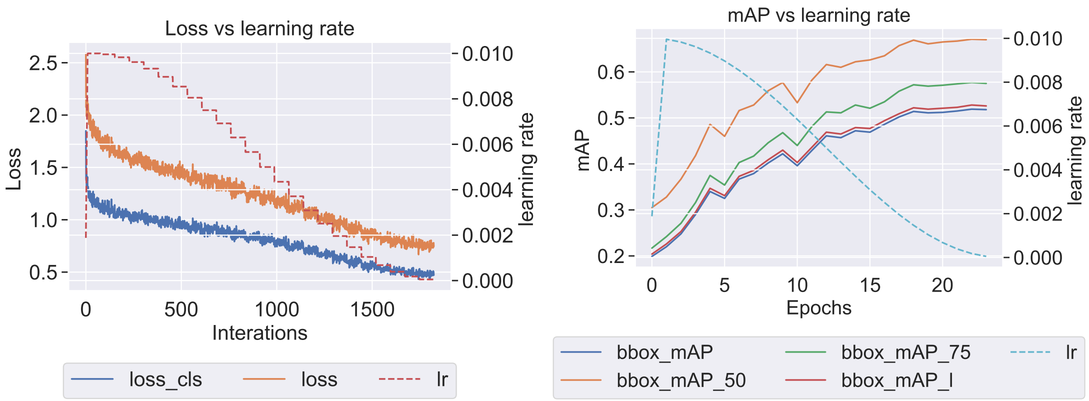
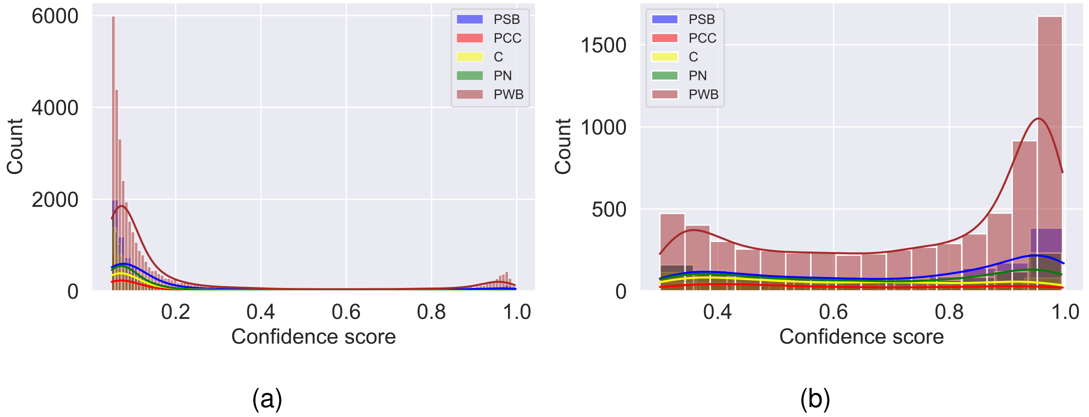


Figure 4.12: Cosine Annealing learning rate loss and mAP vs learning rate



(a)

(b)

Figure 4.13: Cosine Annealing learning rate Histogram of Confidence Scores

but the magnitude is not as extreme. The small improvements in the mAP values from table 4.8 could be attributed to this shift in confidence score.

4.5 DeepTrash Equal

In this experiment, the DeepTrash dataset was modified into two new datasets with a number of images similar to the object class with the fewest images. The first dataset, which contained roughly the same number of objects per image throughout, is labelled DeepTrash Equal (DpTrsh eq). The second dataset, which has the same number of objects per image as the original dataset, is labelled DeepTrash Equal2 (DpTrsh eq2). The number of objects and images for the new datasets are shown in the table 4.9. The table 4.9 shows that the Paper Cardboard Container class, which has the lowest number of images and objects, is kept consistent across the two datasets while the other object classes are modified. The reduction of the dataset is performed using the methods explained in section 3.4.2.

	Category	DpTrsh eq	DpTrsh eq2
Number of Images	PWB[12]	2140	2386
Number of Objects		2556	6740
Number of Images	PN[13]	1012	1179
Number of Objects		1098	1509
Number of Images	C[14]	999	1026
Number of Objects		1136	1165
Number of Images	PCC[15]	758	758
Number of Objects		812	812
Number of Images	PSB[16]	1027	1636
Number of Objects		1092	1300

Table 4.9: Number of images and objects in modified DeepTrash

VarifocalNet trained on DeepTrash Equal

The training of this object detector was performed using the same methods as used in section 4.4. The results demonstrated in table 4.10 show that most of the object categories have similar values of AP although the Paper Cardboard Container has the highest AP and the Plastic Juice Water Bottle has the lowest AP. The overall mAP in table 4.11 is much lower due to a massive difference in total objects and images. The classwise result is quite contrasting compared to the baseline. From tables 4.3 and 4.10, it is clear that the Paper Cardboard Container is a much simpler object for the object detector to learn, whereas the ability of the object detector to learn the object class Plastic Juice Water Bottle has been significantly impacted by the reduction in the number of objects per image.

From figure A.2, the object detector performs quite similarly to when it was trained on the entire DeepTrash dataset. The behaviour of the loss and mAP values follow the same trend as shown in the figures 4.12, though their overall magnitude is lower. Comparing the distribution of confidence scores of the objects predicted by the object

[12] Plastic Juice Water Bottle

[13] Paper Newspaper

[14] Cardboard

[15] Paper Cardboard Container

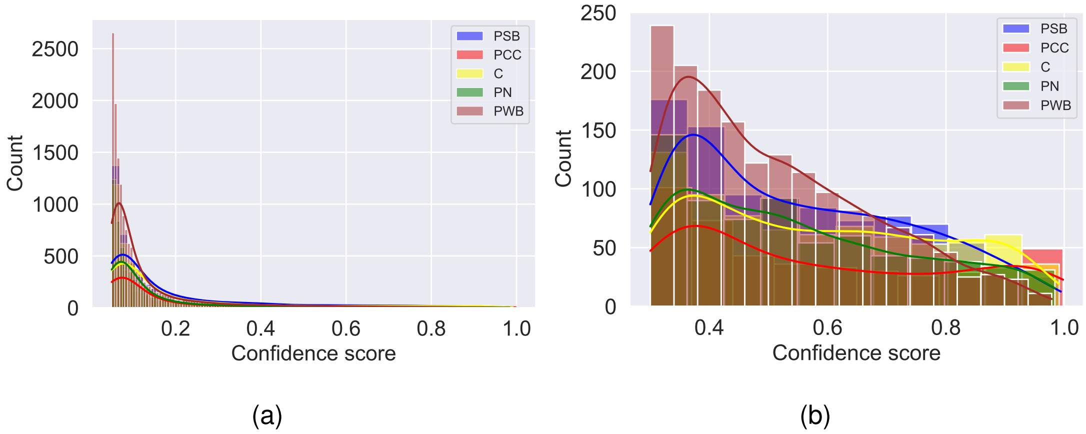
[16] Plastics Shopping Bag

Category	PWB	PN	C	PCC	PSB
AP	0.307	0.308	0.313	0.361	0.373

Table 4.10: DeepTrash Equal Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.33	0.467	0.369	0.339

Table 4.11: DeepTrash Equal mAP values



(a)

(b)

Figure 4.14: DeepTrash Equal Histogram of confidence scores

detector, the differences are seen. In figure 4.14, the majority of the predicted objects, both in the whole range and $> 30\%$, show a clear trend of decreasing number of objects as the confidence score increased. This trend is seen in all the classes. This is the main cause of the much lower mAP values. Thus, from the figure 4.14b, a conclusion that the object detector is underfitting can be derived.

VarifocalNet trained on DeepTrash Equal2

The second part of this experiment is to have a similar distribution of the images as shown in the table 4.9 but the objects per image remain the same as in the original DeepTrash dataset. The composition of the modified dataset is shown in the table 4.9. There is an increase in the number of objects, particularly in the Plastic Juice Water Bottle class, while the rest have a slight increase. This increase in the number of images throughout all the classes is to combat the underfitting observed in figure 4.14b.

The object detector was trained using the baseline hyperparameters and framework, but with the additional data augmentation technique, Auto Augment. The training was run

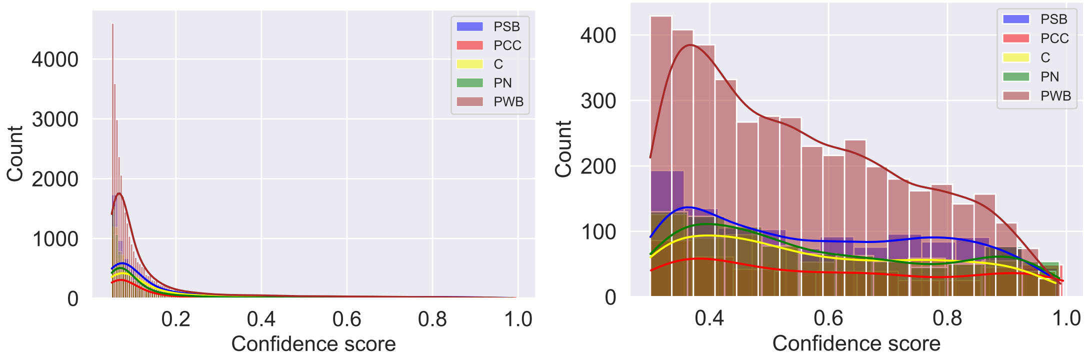
Category	PWB	PN	C	PCC	PSB
AP	0.418	0.364	0.322	0.391	0.417

Table 4.12: DeepTrash Equal2 Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.383	0.52	0.425	0.391

Table 4.13: DeepTrash Equal2 mAP values

for 25 epochs. The results of the training observed in table 4.12 using this dataset show that all the categories have similar AP. When comparing these results to table 4.10, the AP values for Plastic Juice Water Bottle have increased dramatically while the rest of the classes show a small increase in mAP. The increase in annotations, particularly for the Plastic Juice Water Bottle class, is responsible for this. The mAP of the class Paper Cardboard Container is also approaching the value seen in table 4.3. The results in table 4.10 demonstrate that the DeepTrash dataset does not require an equal distribution of objects/images throughout the dataset for classwise AP to be similar across the object classes.



(a)

(b)

Figure 4.15: DeepTrash Equal2 Histogram of confidence scores

When comparing the figures 4.14b and 4.15b, it is clear that there is a significant increase in the number of objects across all confidence scores, indicating an improvement in the underfitting caused by the previous experiment.

4.6 Pre-Trained VarifocalNet on TACO

In this set of experiments, the object detectors are pre-trained with the TACO datasets to understand its impacts. The VarifocalNet object detector was trained using the TACO dataset with the weights from the COCO dataset training. The performance of the TACO dataset is not particularly significant.

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.206	0.240	0.216	0.217

Table 4.14: TACO mAP values

Trained on TACO then DeepTrash

With the exception of changing the learning rate scheduler, adding Auto Augment, and using the initial weights from the object detector trained using the TACO dataset, this object detector's training was carried out using the same hyperparameters as the baseline in section 4.2.

Category	PWB	PN	C	PCC	PSB
AP	0.619	0.511	0.418	0.421	0.604

Table 4.15: Pre-Trained DeepTrash Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.515	0.665	0.571	0.521

Table 4.16: Pre-Trained DeepTrash mAP values

From the metrics in tables 4.16 and 4.15, changing the initial weights from the COCO dataset to the TACO dataset does not show a significant difference in the final mAP values, either classwise or the dataset-wide metrics. The loss pattern in the figure A.5 is similar to figure 4.12. The figure depicts the pattern of the mAP values, which exhibits a slight larger gradient in the early phases of the training period, as well as a smoother and more consistent trend throughout the training period. Comparing the distribution with figures 4.16b and 4.8b, there is a positive change in the distribution. When taking only the objects with a confidence score higher than 30%, the distribution shifts more towards higher confidence. Even when comparing this to figure 4.13b, the

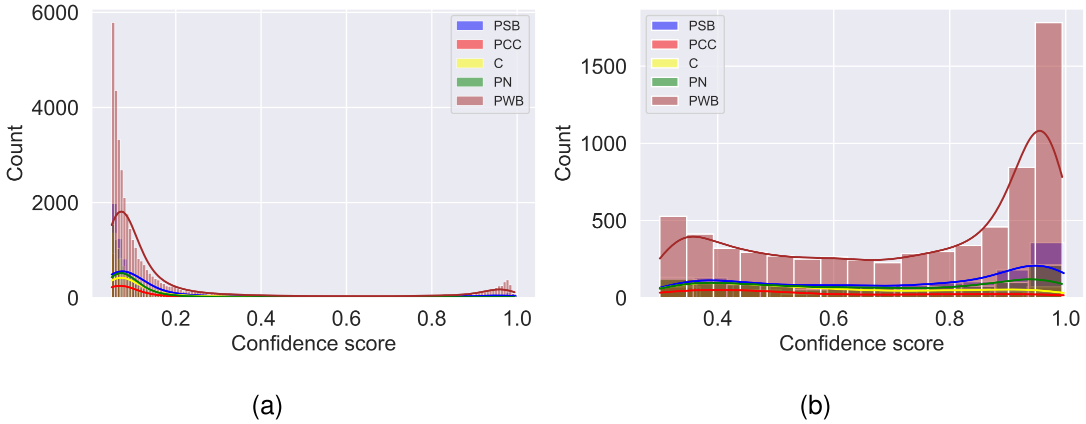


Figure 4.16: Pre-Trained DeepTrash Histogram of Confidence Scores

shift in distribution appears in the classes of Plastic Juice Water Bottles and shopping bags. When comparing the confidence scores in figures 4.16b and 4.8b, this method demonstrates a considerable difference. This improvement is not shown in the mAP values as discussed earlier in section 4.3, due to the missing labels of the DeepTrash dataset.

Trained on TACO then DeepTrash Equal2

This experiment aims to measure the difference in performance when the object detector is pre-trained using the TACO dataset, and then trained on a reduced dataset where the object classwise imbalance has been manually reduced. The reduced dataset is the DeepTrash Equal2. The training parameters are the same as in the previous experiment.

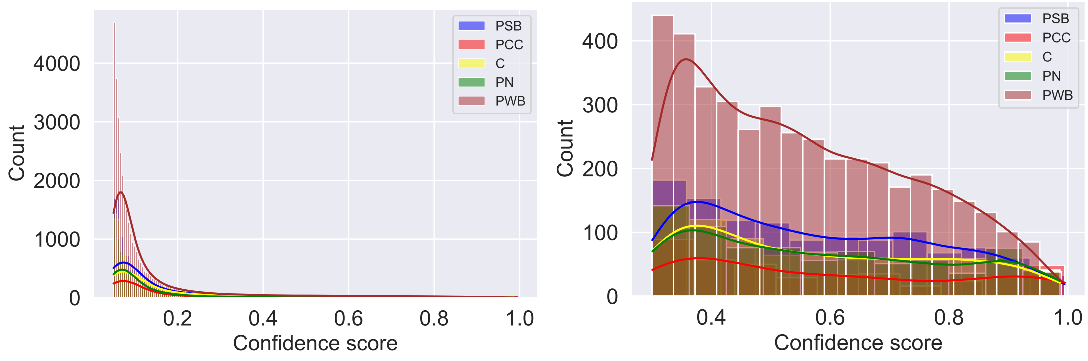
Category	PWB	PN	C	PCC	PSB
AP	0.425	0.346	0.314	0.365	0.404

Table 4.17: Pre-Trained DeepTrash Equal2 Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.371	0.512	0.408	0.379

Table 4.18: Pre-Trained DeepTrash Equal2 mAP values

The tables 4.17 and 4.18 show that the mAP has a very negligible change. The behaviour depicted in figure 4.17b is similar with and without the pre-trained weights of TACO.



(a) (b)

Figure 4.17: Pre-Trained DeepTrash Equal2 Histogram of Confidence Scores

4.7 VarifocalNet Datamix

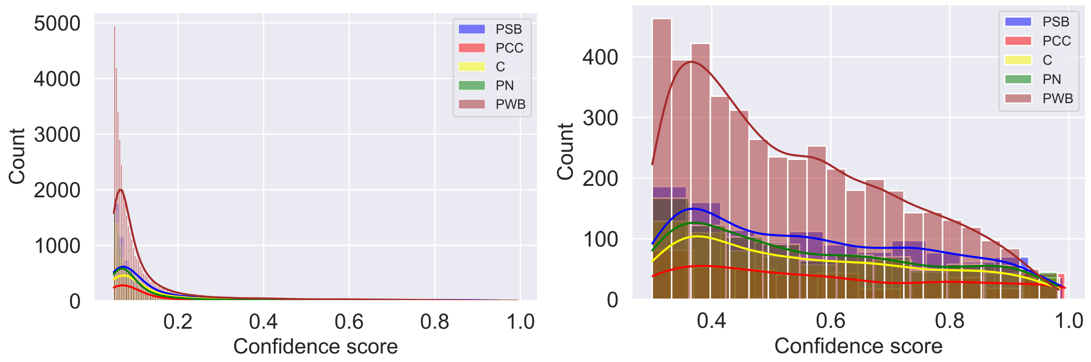
The experiment under discussion aims to understand the impact of adding supplementary datasets to the DeepTrash dataset. The DeepTrash dataset is not used for this experiment as the difference in the number of annotations and images added will be extremely negligible due to the difference in the size of the datasets. Thus, the dataset that will be modified is the DeepTrash Equal2. The VarifocalNet was trained using modified datasets that included the dptrsh eq2 described in section 3.4.3, as well as the modified datasets of TACO and TrashNet indicated in section 3.2. The validation data for this is the same as the one used in the rest of the experiments. The validation dataset has not been changed as the performance on the DeepTrash dataset is the main priority. While Auto Augment has been introduced and the learning rate scheduler has been altered, the object detector's hyperparameters remain the same as the baseline in section 4.2. The training was done for 25 epochs. From the results shown in tables 4.19 and 4.20, almost no change in performance is observed compared to the section 4.6.

Category	PWB	PN	C	PCC	PSB
AP	0.412	0.329	0.315	0.363	0.426

Table 4.19: Datamix Classwise Average Precision

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.369	0.514	0.402	0.377

Table 4.20: Datamix mAP values



(a) (b)

Figure 4.18: Datamix Histogram of Confidence Scores

The results of this experiment, as shown in tables 4.19 and 4.20, and figure 4.18b, reveal that there is less improvement than when the object detector was pretrained using TACO.

4.8 Varifocal Loss Hyperparameter

The distribution of the confidence score, as shown in figure 4.15b, has a decreasing trend. In order to address this, the gamma hyperparameter value of the varifocal loss was decreased. Gamma is the parameter that controls the loss amount given to each object given how hard or easy the object is. As shown in figure 4.15b, most of the objects are difficult to identify due to their low confidence score. Therefore, lowering the gamma value assists the object detector in preserving the signal from the objects with a high confidence score, making more of the objects easily identifiable as training progresses. The VarifocalNet was trained using the dataset DeepTrash Equal2 and using the same methods in section 4.6. The only change was the hyperparameter of the varifocal loss function, gamma. This was changed from 2 to 1.25.

Category	PWB	PN	C	PCC	PSB
AP	0.382	0.534	0.573	0.661	0.512

Table 4.21: Loss Hyperparameter Classwise Average Precision

The results in tables 4.21 and 4.22 reveal that there is a huge improvement in both the overall and classwise performance. Even when compared to the results with the baseline

bbox mAP	bbox mAP 50	bbox mAP 75	bbox mAP l
0.533	0.653	0.583	0.537

Table 4.22: Loss Hyperparameter mAP values

section 4.2 or the learning rate scheduling section 4.4 presented in tables 4.4, 4.3, and 4.7, 4.8, the performance of most classes has increased significantly. All the classes except for the Plastic Juice Water Bottle class have shown improvement. This could be due to the missing labels of the dataset or the lack of images and objects compared to the whole DeepTrash dataset. Figure 4.17b shows an improvement in the distribution of the object's confidence score for all the categories. The performance increase in tables 4.21 and 4.22 can be primarily attributed to this.

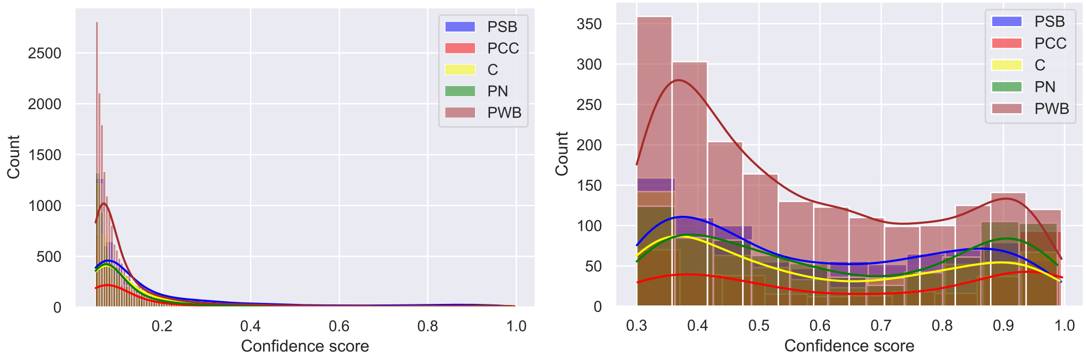


Figure 4.19: Loss Hyperparameter Histogram of Confidence Scores

Chapter 5

Discussion and Conclusions

In this dissertation, object detection techniques were analysed to detect 5 different categories of recyclable waste in the DeepTrash[1] dataset. The driving force behind the development of this dissertation is the problem with garbage disposal that plagues most major metropolitan areas worldwide. Garbage disposal and segregation need a significant amount of human resources, making it an expensive practice for many communities to undertake. To address this, developing an automated system that decreases the cost and time required to sort recycled waste will benefit both cities and the environment.

The dataset under research , DeepTrash (collected by Danu Robotics from a trash sorting facility) was observed to suffer from numerous imbalances, namely object and image imbalance between classes, and foreground-background and foreground-foreground imbalance. These imbalances prevent the object detector from correctly learning the dataset, which results in poor performance. To mitigate these problems, two object detectors were chosen, RetinaNet and VarifocalNet. They were chosen because they have a loss function that attempts to minimise the effects of class imbalance. First, baseline testing was performed on the two object detectors, VarifocalNet and RetinaNet. These tests demonstrated the problems of object classwise imbalance and loss and accuracy being trapped at local minima due to the learning rate. Due to this, additional experiments were performed such as changing the learning rate scheduler, adding data augmentation techniques, modifying the size and properties (objects per image) of the dataset, and pretraining on a dataset which is similar to the DeepTrash dataset and

[1] The intellectual property of the DeepTrash dataset is owned by Danu Robotics and is copyright protected. The dataset will not be publicly released and all the photos are used for demonstration purposes only. Any use of the under-discussion images in research or industrial is illegal.

modifying the hyperparameters of the loss function.

The outcome of these results was positive as they provided an understanding of the methods to improve the overall and classwise performance. The addition of Auto Augment provided a small improvement in the confidence of the object detector on a large number of images. In the vanilla implementation of the framework, the learning rate was set to be a step function, stepping at epochs 16 and 22. However, this was observed to get stuck in the local minima. To cater to this, cosine annealing function was proposed for the task at hand. This slightly improved the mAP while also lowering the overall loss and increasing the number of objects with high confidence. Changing the size of the dataset in terms of images and objects per image yielded some intriguing results. This demonstrated that for the object detector to learn the object classes equally, some object classes required a larger number of objects and images. The object detector was then trained using the reduced dataset as well as extra data from TACO and TrashNet. The results did not provide significant improvement. The object detector was then pretrained on TACO using transfer learning as it is a very common practice. This experiment did not provide a significant improvement in terms of the mAP values but showed that more objects from different categories have a higher confidence score (this issue is discussed in section 5.1). Finally, the hyperparameter of the loss functions was modified to study the effect of hyperparameter tuning. The findings suggest that hyperparameter optimisation is possible, as the Object Detector trained on DeepTrash Equal2 as given in section 4.8 exhibits a significant improvement in mAP values when compared to training on DeepTrash Equal2 or on TACO then DeepTrash as described in sections 4.8 and 4.6 respectively.

Following these results, several open questions arise: Can a class-wise varifocal loss improve the performance of the object detector? How may the effects of an incompletely labelled dataset be reduced? Before these open questions are discussed, the limitation so far will be discussed.

5.1 Limitations

5.1.1 Time and Resources

The amount of time and resources that are typically utilised for such deep learning research is one of the biggest limitations of this work. The training of most of the object detectors for 25 epochs took around a day. This hinders the object detector from being run for 100 epochs, as most object detectors are when trained on the COCO datasets. The limitations of VRAM also occurred as the GPUs for training, the 2080ti and 3070 have 11GB and 8GB respectively.Consequently, it was not possible to utilise a larger Object Detector, such as VarifocalNet -X, an object detector that attained state-of-the-art mAP values on the COCO test-dev.

5.1.2 Missing Labels in Dataset

After careful visual analysis, the DeepTrash dataset has missing labels. As in supervised learning, the dataset is extremely important as any deep learning object detector completely depends on it. The impact of this is seen in the mAP values as they do not fully correspond to how the object detector performs in the real-world scenario through visual analysis. The issue is that having incomplete labelling leads to many true positives being considered as false positives which affects the precision, thus affecting the AP and mAP values of the object detector. In the case of fully supervised networks, there is no temporary fix to get around this issue.

5.2 Open Questions

1. Can a class-wise varifocal loss improve the performance of the object detector?

   When the loss functions' hyperparameters were changed, different object classes were affected differently. Creating a varifocal loss that has been optimised for each class could potentially improve the performance based on the results observed. Additionally as different classes require different number of annotations and images to reach the same AP values. This property of the dataset could be used as an additional hyperparameter which could be optimised for improved performance

2. How may the effects of an incompletely labelled dataset be reduced?

   When the metric results are compared to visual inference, it is clear that the object detector performs better than the metrics indicate. This is owing to the dataset's incomplete labelling. As a result, the number of false positives increases when evaluating, lowering the AP and mAP. Given the increasing use of object detectors in many tasks, the fact that there are no measures that can resolve this problem poses a significant difficulty. There is currently no way to resolve this problem other than labelling the unlabelled objects in the dataset.

5.3 Conclusions

Throughout this dissertation, methods that can improve the performance of the object detector were assessed. The existence of object class imbalance was proven through baseline experiments and data analysis in section 4.2 and 4.1 respectively. The problem of incomplete dataset labelling was discovered. The results of the experiments indicated that using the Auto Augment data augmentation technique and the cosine annealing learning rate scheduler improved the object detector's ability to learn object categories.

The experiments involving training on the artificially reduced datasets yielded some intriguing results. When the number of objects per image was kept consistent, the object detector did not learn the object classes equally. The object class Plastic Juice Water Bottle needed many more objects to achieve the same AP as Paper Cardboard Container. The addition of supplementary datasets to the DeepTrash Equal2 while training did not provide any additional benefit. Training the object detector initially on TACO, then on the DeepTrash and DeepTrash Equal2 datasets showed minor improvements in terms of metrics. When comparing the confidence scores, a significant improvement was observed. The lack of observable improvement in the mAP metrics was attributable to the incomplete labelling of the DeepTrash dataset. Using the confidence score graphs from all of the experiments in sections 4.5 and 4.6, a hypothesis was developed that adjusting the varifocal loss hyperparameters would increase the object detector's performance. This was accurate because DeepTrash Equal2 with the modified varifocal loss hyperparameters in section 4.8 generated the best mAP and greater classwise AP for most object classes.

Overall, these experiments resulted in the best performing object detector with an mAP of 53.3 (section 4.8), which is a 2.5 mAP improvement over the baseline in section 4.2(mAP of 50.8) and a 1.5 mAP improvement over the best performing object detector trained on the DeepTrash dataset (mAP of 51.8) in section 4.4.There is an improvement of 14.5 mAP when comparing the experiments with and without the varifocal loss hyperparameter change in section 4.8 and 4.6 respectively. The best performing object detector's dataset, DeepTrash Equal2, contained 56.9% of the images and 29.82%

of the annotation when compared to the dataset used in training in section 4.4. The incomplete labelling of the dataset caused a problem when using only the mAP and AP for comparison. When comparing visually, the object detectors trained on DeepTrash data performed similarly or better than the object detector with an mAP of 53.3 in section 4.8. The Plastic Juice Water Bottle is the object class most affected by missing labels. This class is especially necessary because plastic bottles are the most profitable for recycling.

Labelling datasets with clustered objects such as trash, satellite pictures, etc, is a key topic that must be addressed carefully. Otherwise, the measurements will fail to accurately reflect true performance, compromising the research's validity.

Bibliography

[1] Pranav Adarsh, Pratibha Rathi, and Manoj Kumar. Yolo v3-tiny: Object detection and recognition using one stage improved model. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 687–694. IEEE, 2020.

[2] Daniel S Amick. Reflection on the origins of recycling: a paleolithic perspective. *Lithic technology*, 39(1):64–69, 2014.

[3] Thomas Southcliffe Ashton et al. The industrial revolution 1760-1830. *OUP Catalogue*, 1997.

[4] Oluwasanya Awe, Robel Mengistu, and Vikram Sreedhar. Smart trash net: Waste localization and classification. *arXiv preprint*, 2017.

[5] Cenk Bircanoğlu, Meltem Atay, Fuat Beşer, Özgün Genç, and Merve Ayyüce Kızrak. Recyclenet: Intelligent waste sorting using deep neural networks. In *2018 Innovations in intelligent systems and applications (INISTA)*, pages 1–7. IEEE, 2018.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[7] CBSNewYork. https://www.cbsnews.com/newyork/news/nyc-garbage-pickup-charges/, 2018.

[8] Rahul Chauhan, Kamal Kumar Ghanshala, and RC Joshi. Convolutional neural network (cnn) for image detection and recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pages 278–282. IEEE, 2018.

[9] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[10] William WL Cheung, Gabriel Reygondeau, and Thomas L Frölicher. Large benefits to marine fisheries of meeting the 1.5 c global warming target. *Science*, 354(6319):1591–1594, 2016.

[11] Laurie Davidson Cummings. Voluntary strategies in the environmental movement: recycling as cooptation. *Journal of Voluntary Action Research*, 6(3-4):153–160, 1977.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[13] Katja Frieler, M Meinshausen, A Golly, Matthias Mengel, K Lebek, SD Donner, and O Hoegh-Guldberg. Limiting global warming to 2 c is unlikely to save most coral reefs. *Nature Climate Change*, 3(2):165–170, 2013.

[14] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] Weifeng He, Caizhi Li, Xiangfan Nie, Xiaolong Wei, Yiwen Li, Yuqin Li, and Sihai Luo. Recognition and detection of aero-engine blade damage based on improved cascade mask r-cnn. *Applied Optics*, 60(17):5124–5133, 2021.

[19] ipcc. Global warming of 1.5 ºc. https://www.ipcc.ch/sr15/, 2020.

[20] Artur Karaźniewicz. pycocosplit. https://github.com/akarazniewicz/cocosplit, 2020.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[22] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Retinanet-mmdetection. `https://github.com/open-mmlab/mmdetection/tree/master/configs/retinanet`, 2019.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[24] C.H. Lipsett and M.S. Horowitz. *100 Years of Recycling History: From Yankee Tincart Peddlers to Wall Street Scrap Giants*. Atlas Publishing Company, 1974.

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[26] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3388–3415, 2020.

[27] Joy A Palmer. Environmental thinking in the early years: Understanding and misunderstanding of concepts related to waste management. *Environmental Education Research*, 1(1):35–45, 1995.

[28] Pedro F Proença and Pedro Simões. Taco: Trash annotations in context for litter detection. *arXiv preprint arXiv:2003.06975*, 2020.

[29] Automatic Differentiation In Pytorch. Pytorch, 2018.

[30] Hong Qin, Yirong Wu, Fangmin Dong, and Shuifa Sun. Dense sampling and detail enhancement network: Improved small object detection based on dense sampling and detail enhancement. *IET Computer Vision*, 2022.

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

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[34] Roboflow.

[35] Carl-Friedrich Schleussner, Tabea K Lissner, Erich M Fischer, Jan Wohland, Mahé Perrette, Antonius Golly, Joeri Rogelj, Katelin Childers, Jacob Schewe, Katja Frieler, et al. Differential climate impacts for policy-relevant limits to global warming: the case of 1.5 c and 2 c. *Earth system dynamics*, 7(2):327–351, 2016.

[36] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Rumana Sultana. *Trash and Recyclable Material Identification Using Convolutional Neural Networks (CNN)*. PhD thesis, Western Carolina University, 2020.

[39] Mohbat Tharani, Abdul Wahab Amin, Mohammad Maaz, and Murtaza Taj. Attention neural network for trash detection on water channels. *arXiv preprint arXiv:2007.04639*, 2020.

[40] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[41] Tzutalin. Labelimg. Free Software: MIT License, 2015.

[42] Robert J Wang, Xiang Li, and Charles X Ling. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018.

[43] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the Web Conference 2021*, pages 1785–1797, 2021.

[44] Yuheng Wang, Wen Jie Zhao, Jiahui Xu, and Raymond Hong. Recyclable waste identification using cnn image recognition and gaussian clustering. *arXiv preprint arXiv:2011.01353*, 2020.

[45] Terrence H Witkowski. World war ii poster campaigns–preaching frugality to american consumers. *Journal of advertising*, 32(1):69–82, 2003.

[46] Mindy Yang and Gary Thung. Classification of trash for recyclability status. *CS229 project report*, 2016(1):3, 2016.

[47] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, page 103514, 2022.

[48] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sünderhauf. Varifocalnet-mmdetection. `https://github.com/open-mmlab/mmdetection/tree/master/configs/vfnet`, 2020.

[49] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021.

[50] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020.

[51] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

[52] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020.

Appendix A

Additonal data from Experiments

A.1 Number of Images Predicted

Table A.1: Number of objects Predicted

Categories	GT	og	aa	lr	eq	eq2	td	tdeq2l
PWB[1]	5524	7547	7495	6820	1666	4284	6825	1978
PN[2]	799	1288	1267	1174	703	847	1111	780
C[3]	415	824	886	721	763	761	755	643
PCC[4]	224	416	419	338	453	450	345	351
PSB[5]	1008	1705	1714	1557	934	1170	1496	850

Where GT is the Ground Truth;og refers to section 4.2;aa refers to section 4.3;lr refers to section 4.4;eq refers to section 4.5;eq2 refers to section 4.5;td refers to section 4.6;tdeq2l refers to section 4.8;

A.2 Loss and mAP vs Learning Rate

[1] Plastic Juice Water Bottle

[2] Paper Newspaper

[3] Cardboard

[4] Paper Cardboard Container

[5] Plastics Shopping Bag

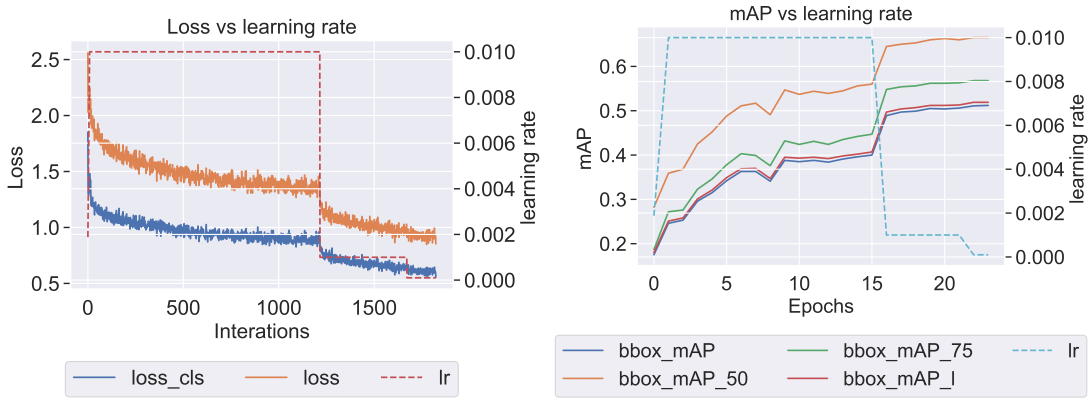


Figure A.1: Auto Augment loss and mAP vs learning rate

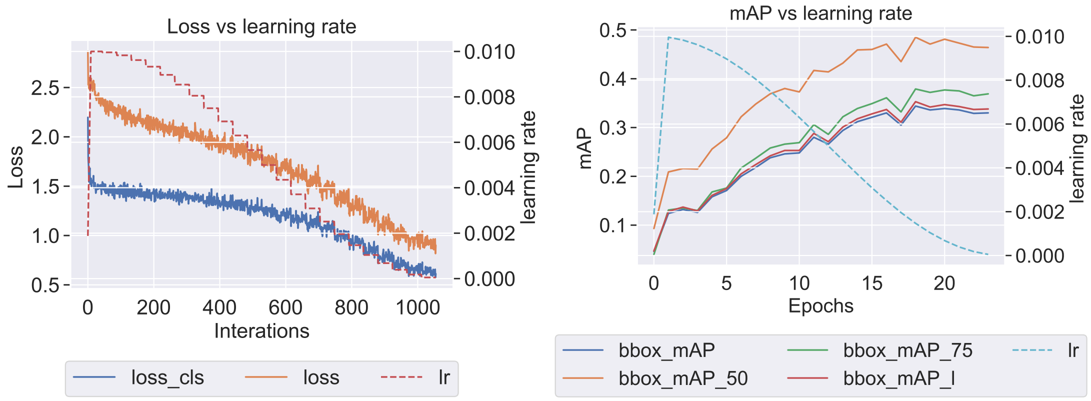


Figure A.2: DeepTrash Equal loss and mAP vs learning rate

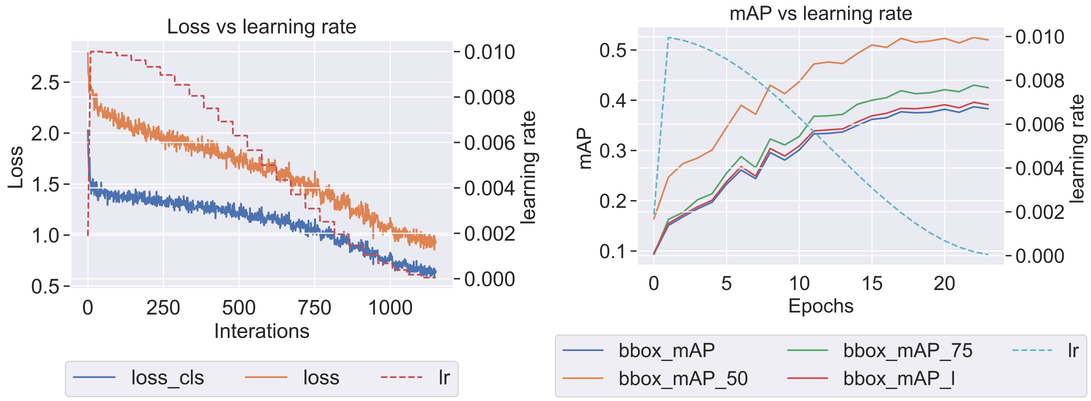


Figure A.3: DeepTrash Equal2 loss and mAP vs learning rate

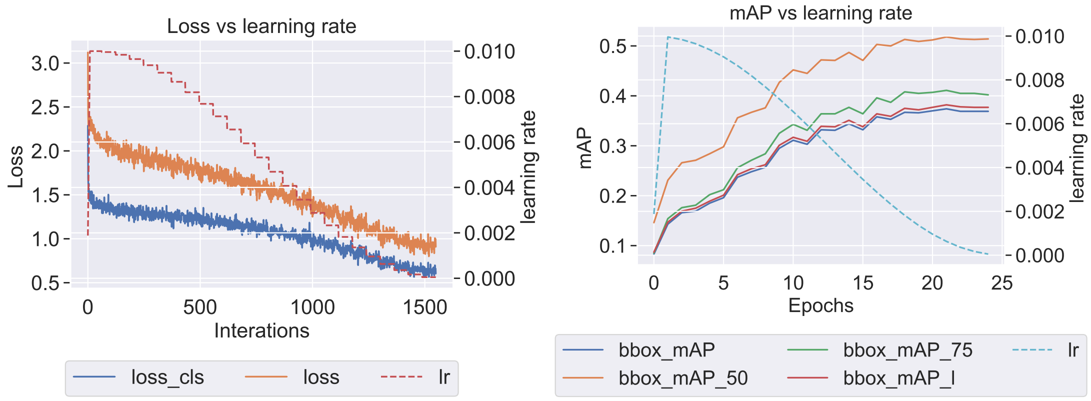
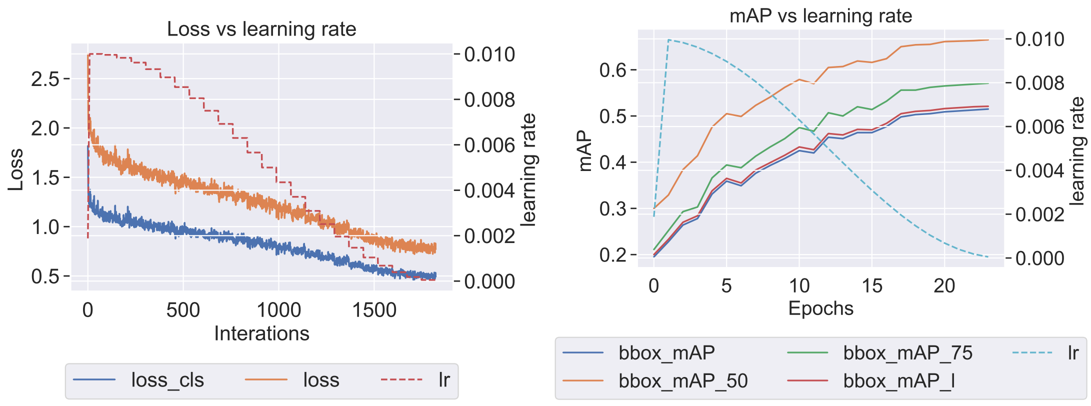


Figure A.4: Datamix loss and mAP vs learning rate



(a) (b)

Figure A.5: Pre-Trained DeepTrash loss and mAP vs learning rate

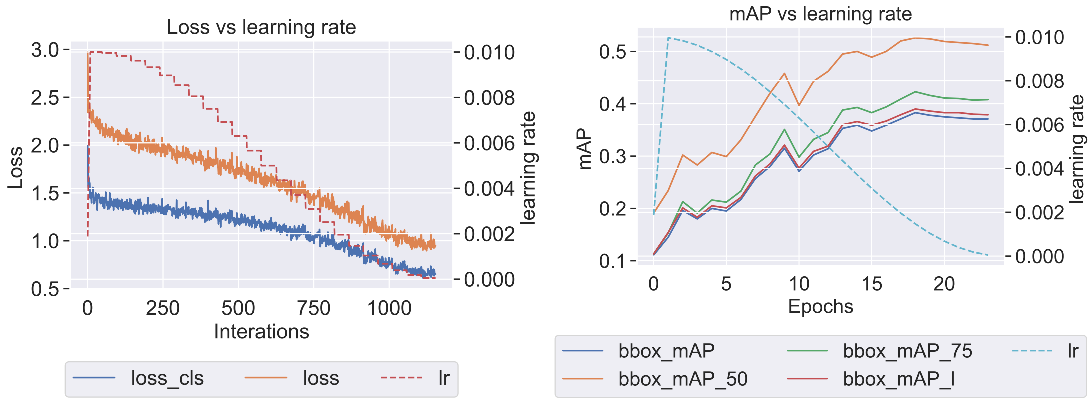


Figure A.6: Pre-Trained DeepTrash Equal2 and mAP vs learning rate

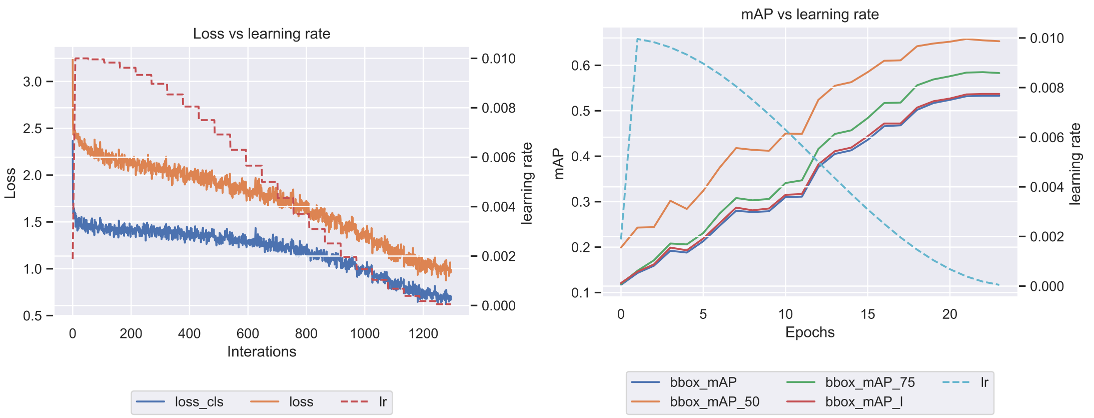
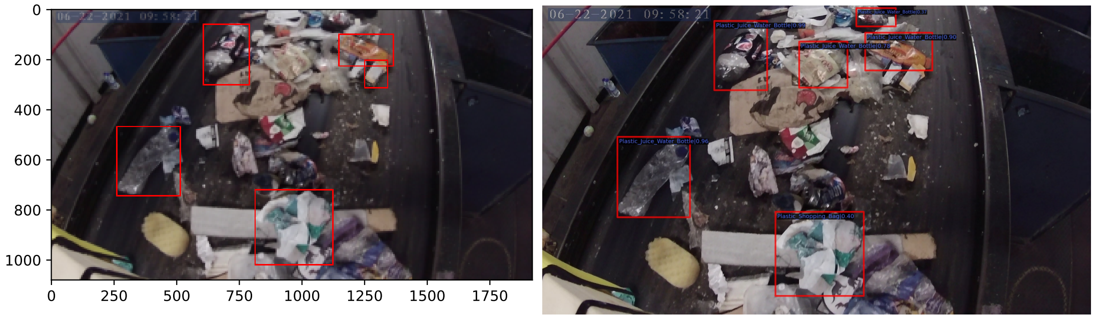


Figure A.7: Loss Hyperparameter loss and mAP vs learning rate

Appendix B

Baseline Visual Inference

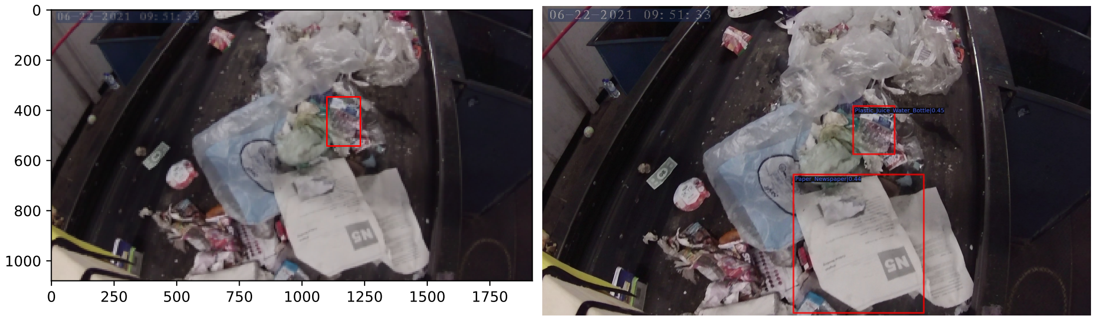


0
200
400
600
800
1000
0
250
500
750
1000
1250
1500
1750
06-22-2021 09:58:21
Plastic_Juice_Water_Bottle|0.99
Plastic_Juice_Water_Bottle|0.37
Plastic_Juice_Water_Bottle|0.90
Plastic_Juice_Water_Bottle|0.78
Plastic_Juice_Water_Bottle|0.96
Plastic_Shopping_Bag|0.40

(a) Ground Truth

(b) Predicted Output

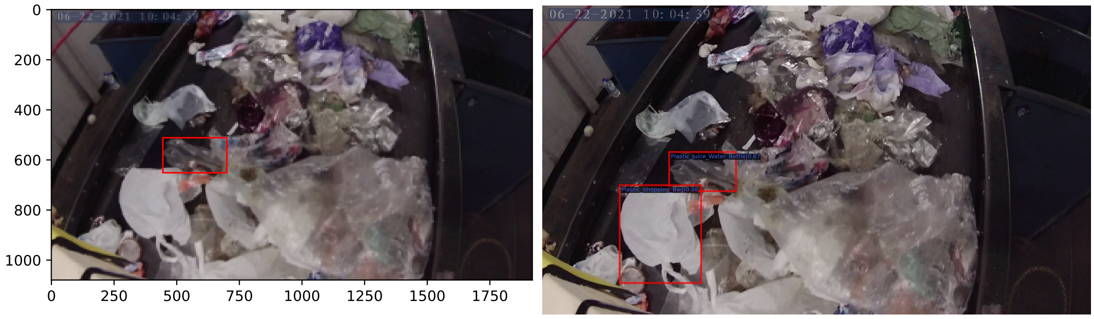
Figure B.1: Example 3 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

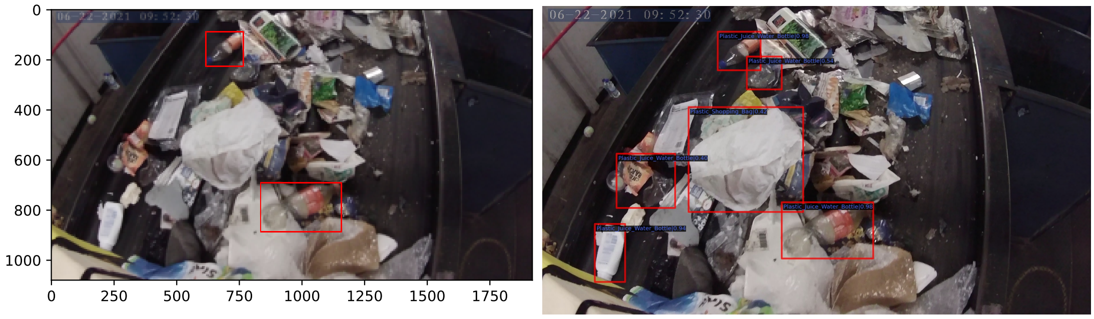
Figure B.2: Example 4 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

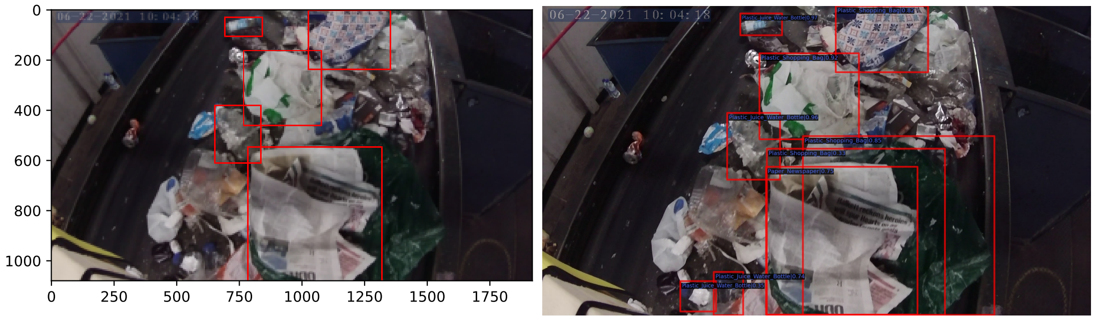
Figure B.3: Example 5 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

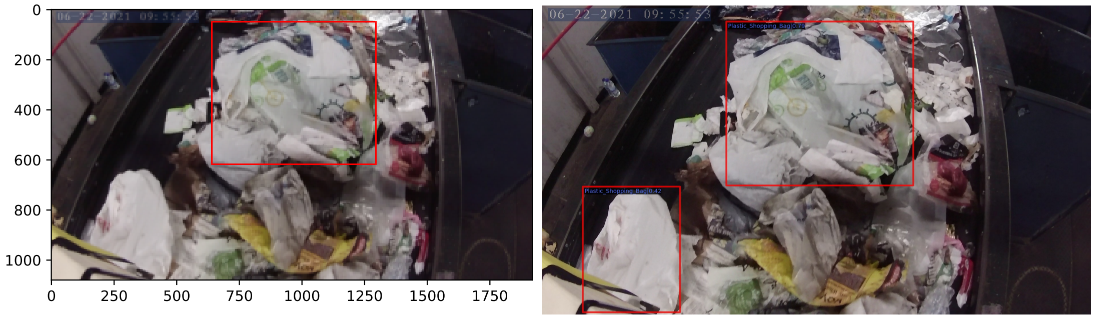
Figure B.4: Example 6 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

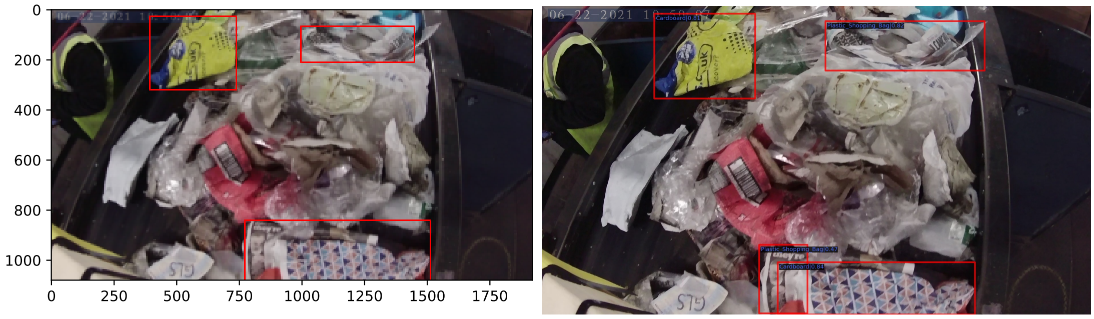
Figure B.5: Example 7 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

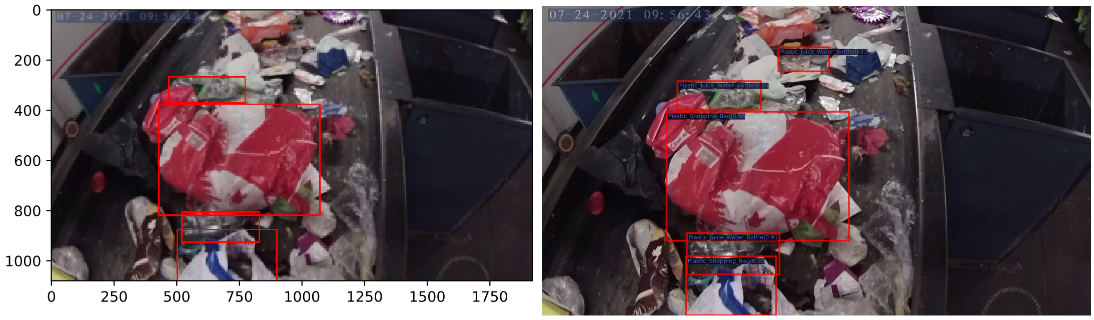
Figure B.6: Example 8 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

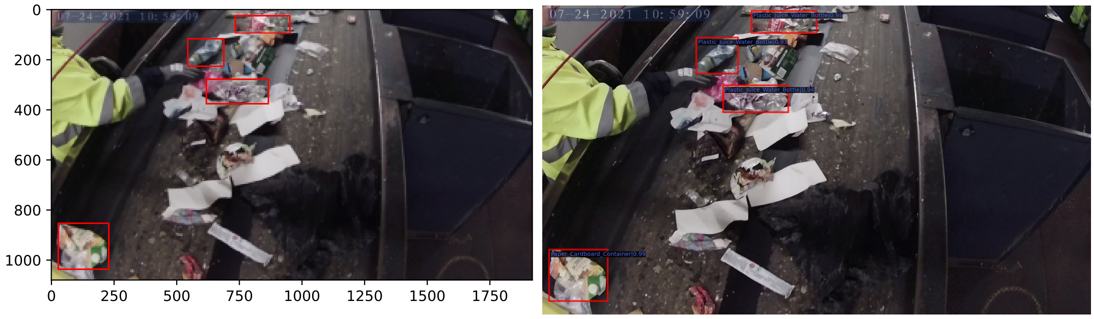
Figure B.7: Example 9 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

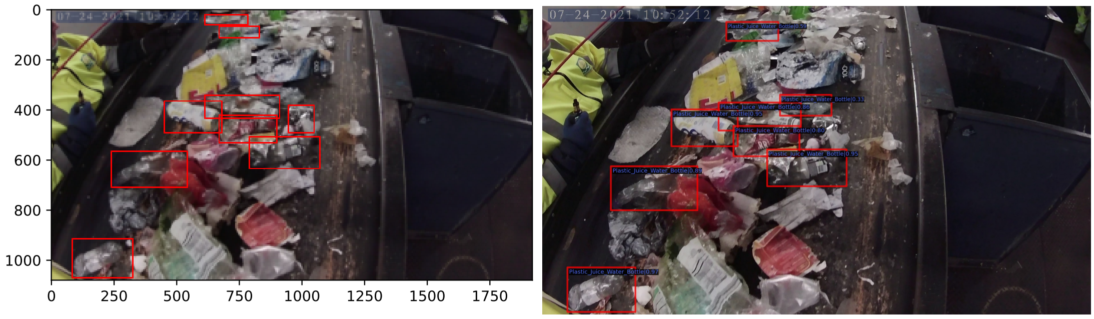
Figure B.8: Example 10 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

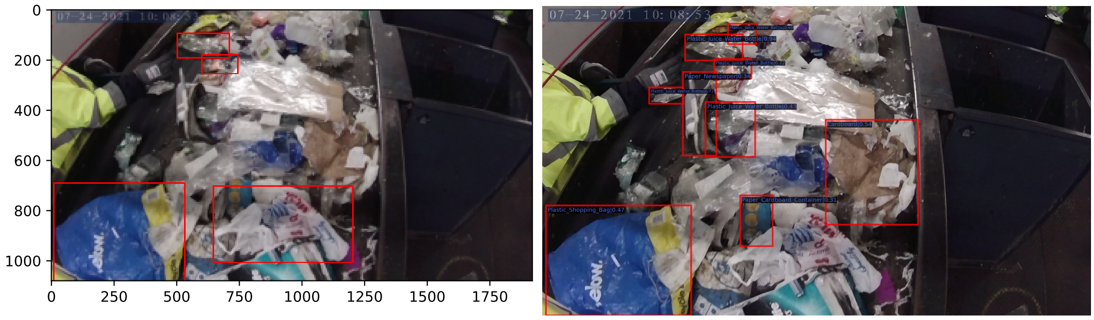
Figure B.9: Example 11 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

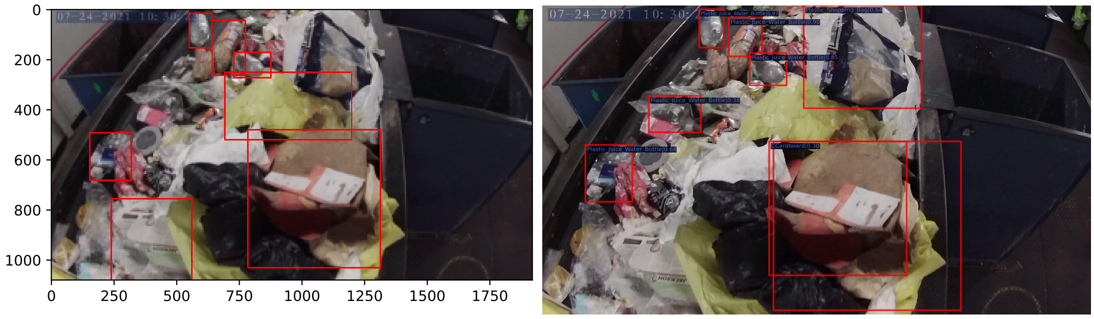
Figure B.10: Example 12 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

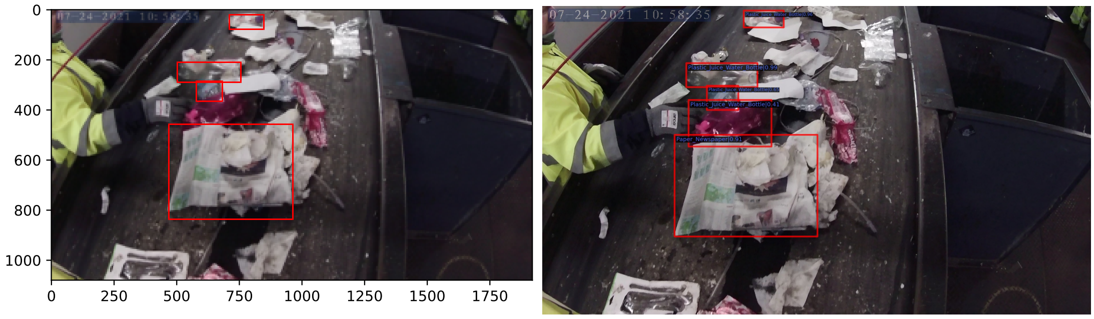
Figure B.11: Example 13 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

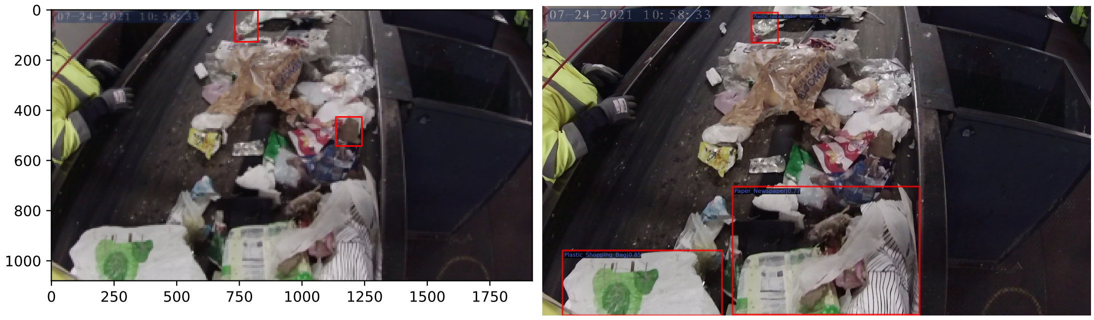
Figure B.12: Example 14 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

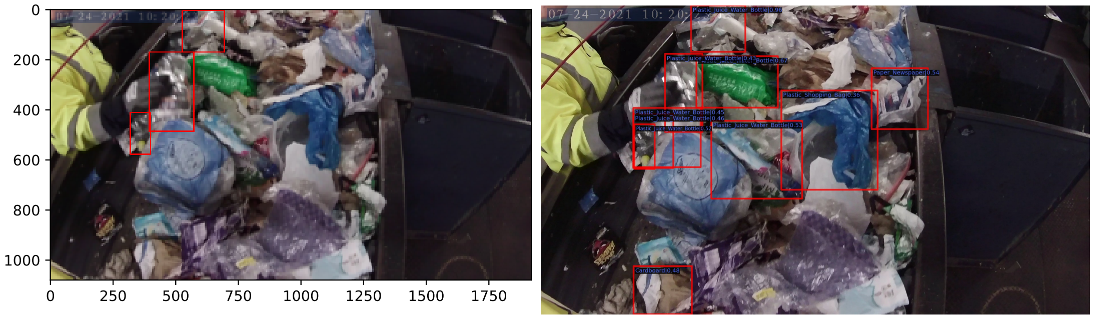
Figure B.13: Example 15 of comparing the ground truth to inference



(a) Ground Truth

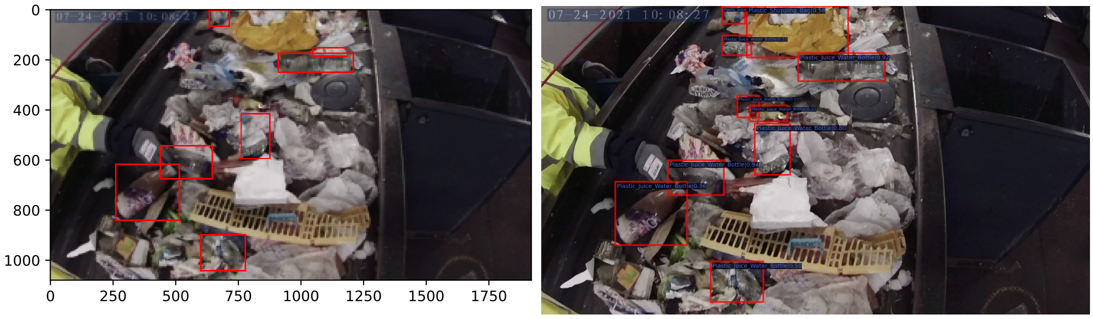
(b) Predicted Output

Figure B.14: Example 16 of comparing the ground truth to inference



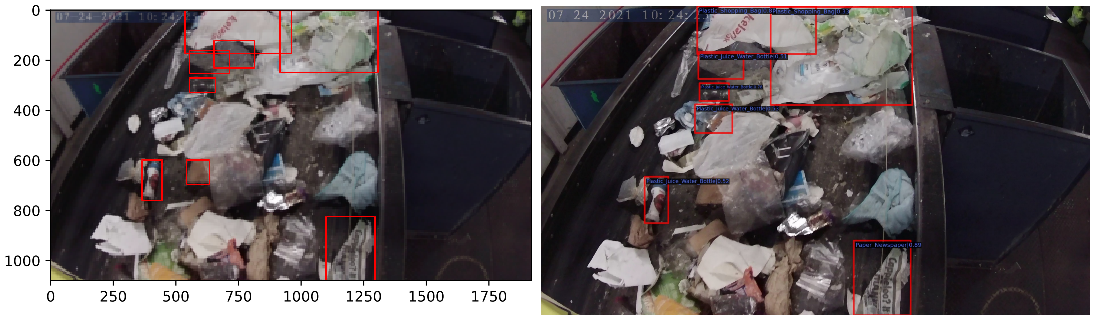
(a) Ground Truth (b) Predicted Output

Figure B.15: Example 17 of comparing the ground truth to inference



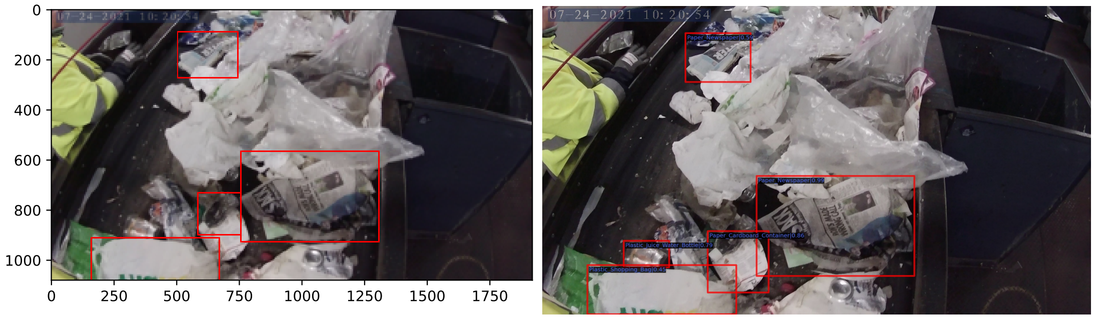
(a) Ground Truth (b) Predicted Output

Figure B.16: Example 18 of comparing the ground truth to inference



(a) Ground Truth (b) Predicted Output

Figure B.17: Example 19 of comparing the ground truth to inference



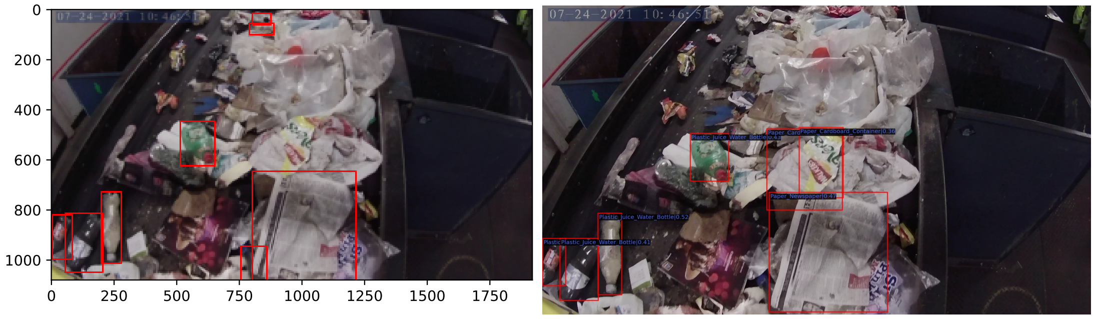
(a) Ground Truth

(b) Predicted Output

Figure B.18: Example 20 of comparing the ground truth to inference

Appendix C

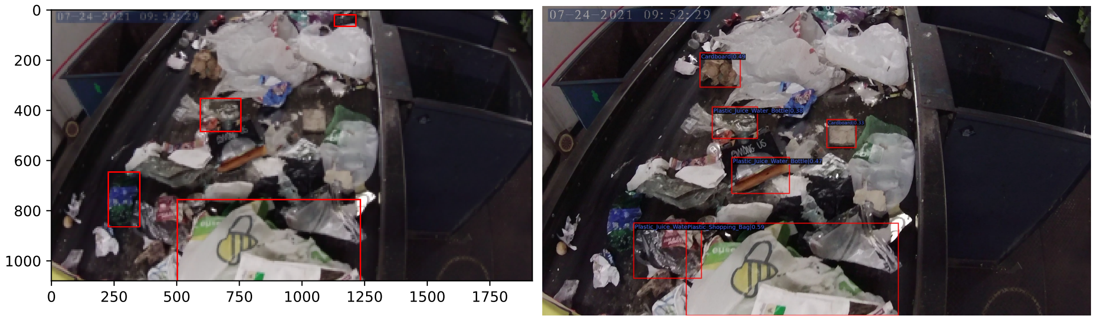
Varifocal Loss Hyperparameter Visual Inference



(a) Ground Truth

(b) Predicted Output

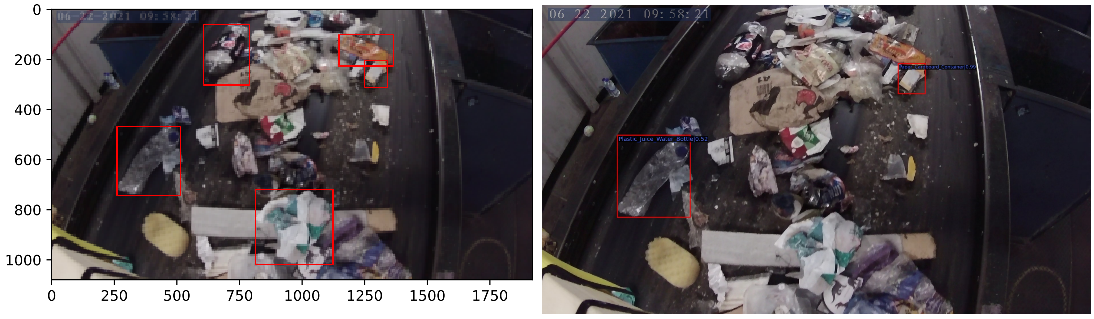
Figure C.1: Example 1 of comparing the ground truth to inference



(a) Ground Truth

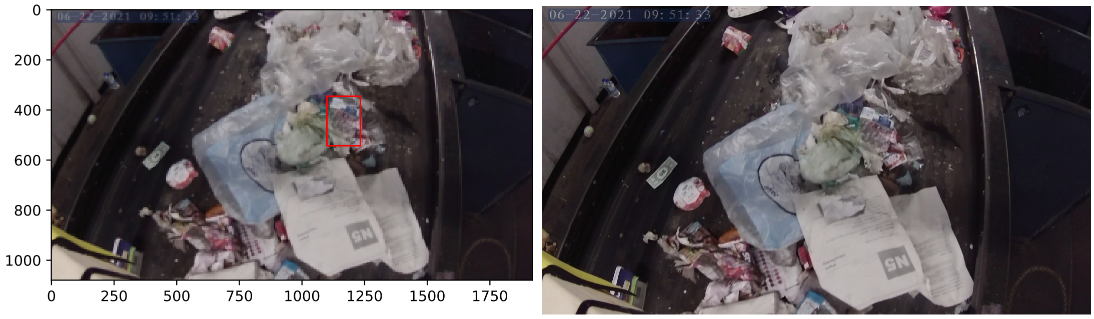
(b) Predicted Output

Figure C.2: Example 2 of comparing the ground truth to inference



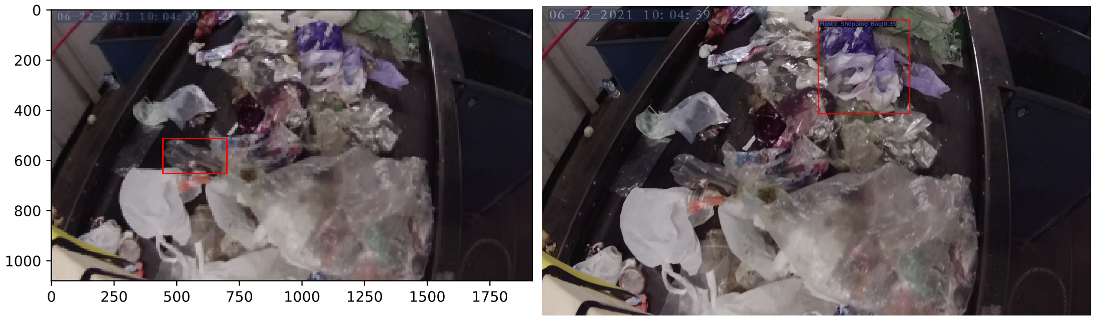
(a) Ground Truth (b) Predicted Output

Figure C.3: Example 3 of comparing the ground truth to inference



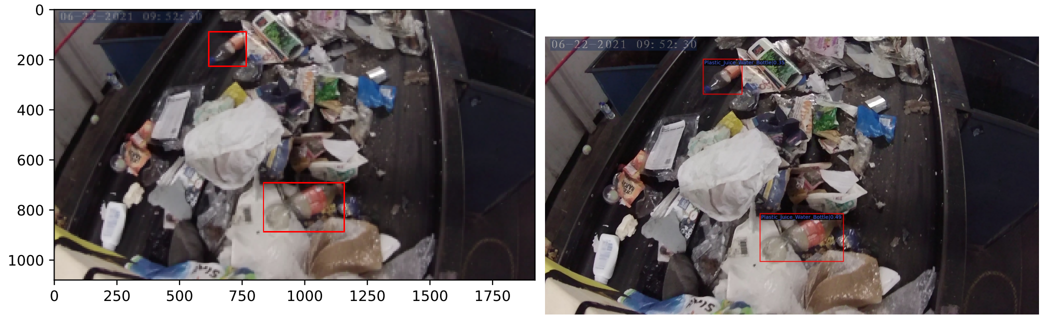
(a) Ground Truth (b) Predicted Output

Figure C.4: Example 4 of comparing the ground truth to inference



(a) Ground Truth (b) Predicted Output

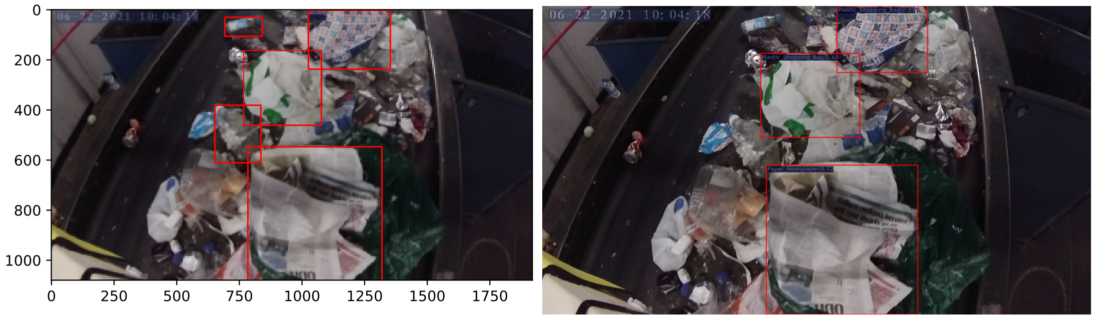
Figure C.5: Example 5 of comparing the ground truth to inference


0
200
400
600
800
1000
0
250
500
750
1000
1250
1500
1750
06-22-2021 09:52:30
Plastic_Juice_Water_Bottle|0.35
Plastic_Juice_Water_Bottle|0.49

(a) Ground Truth

(b) Predicted Output

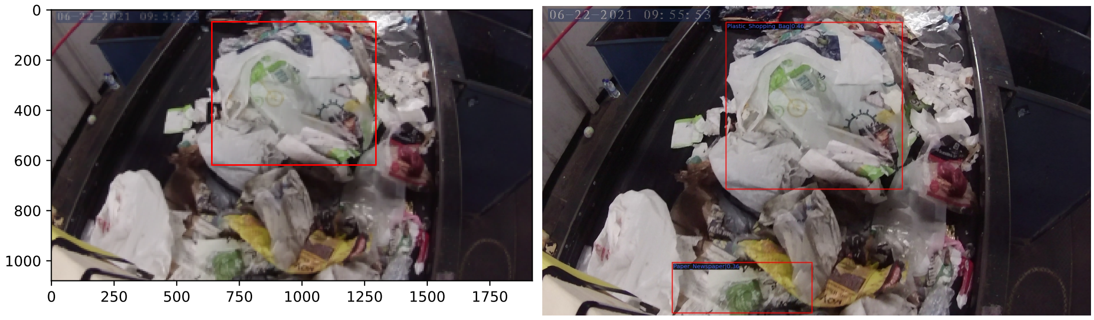
Figure C.6: Example 6 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

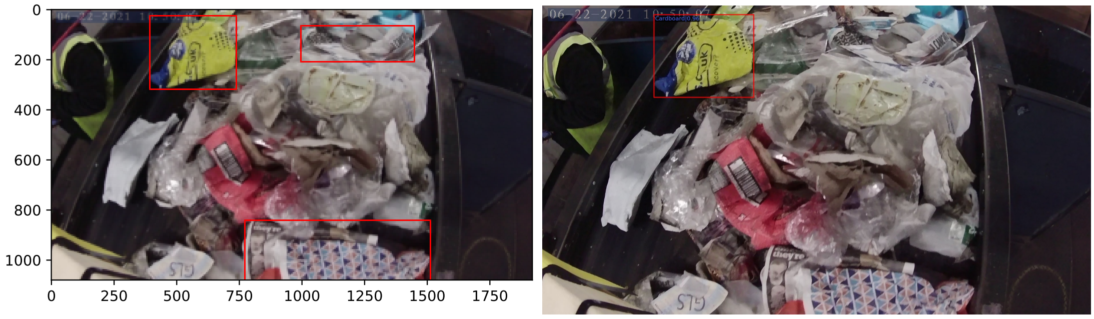
Figure C.7: Example 7 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

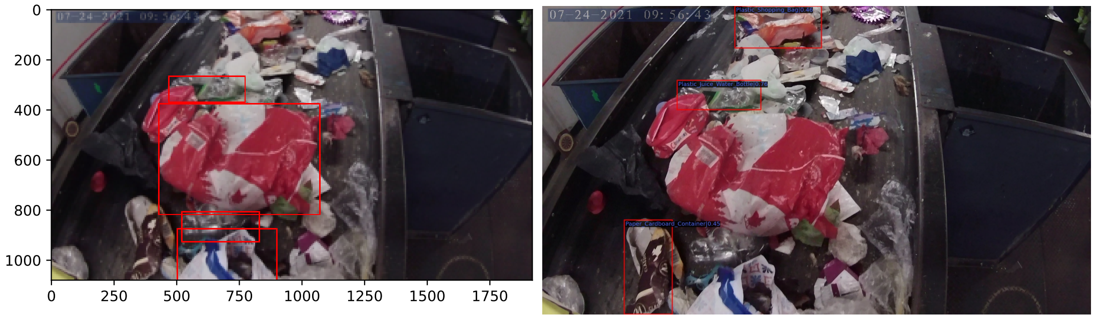
Figure C.8: Example 8 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

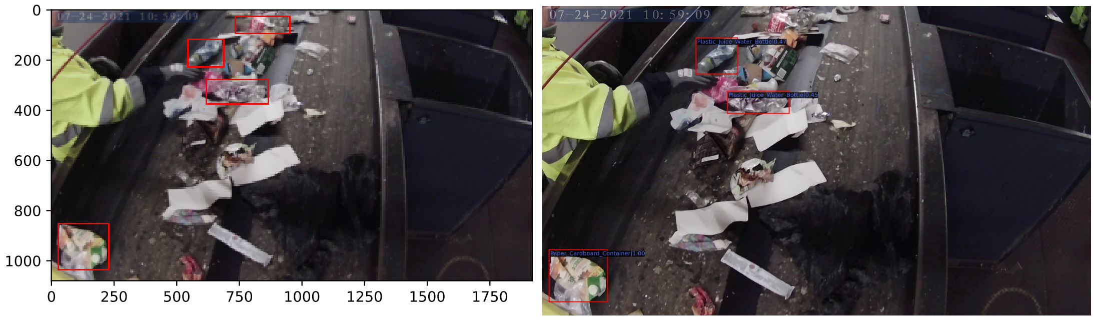
Figure C.9: Example 9 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

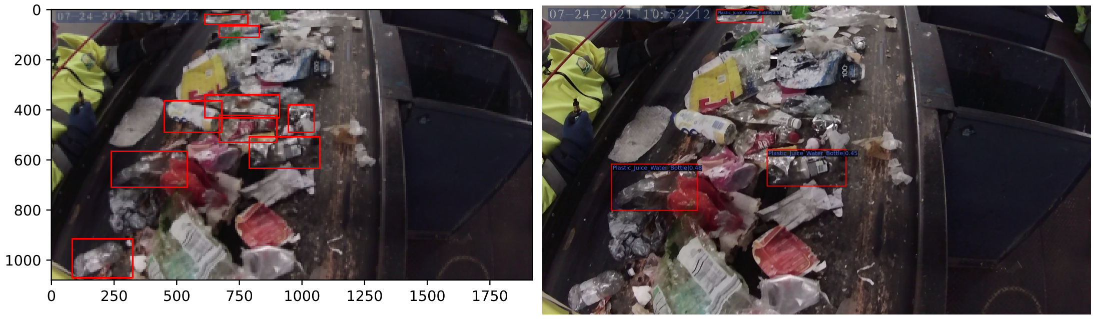
Figure C.10: Example 10 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

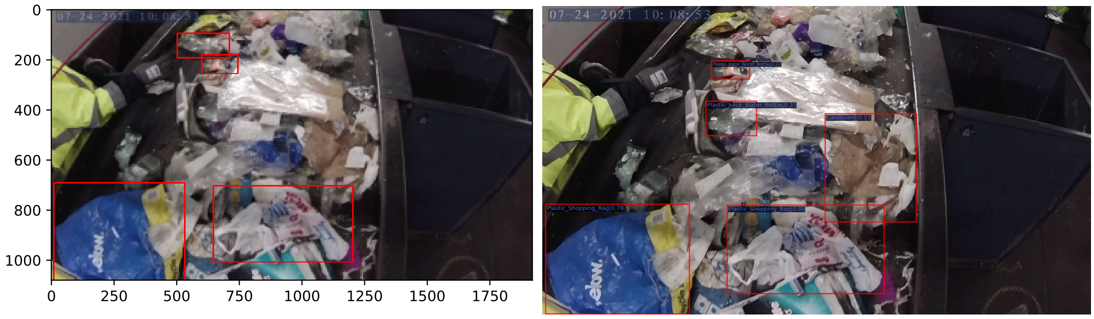
Figure C.11: Example 11 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

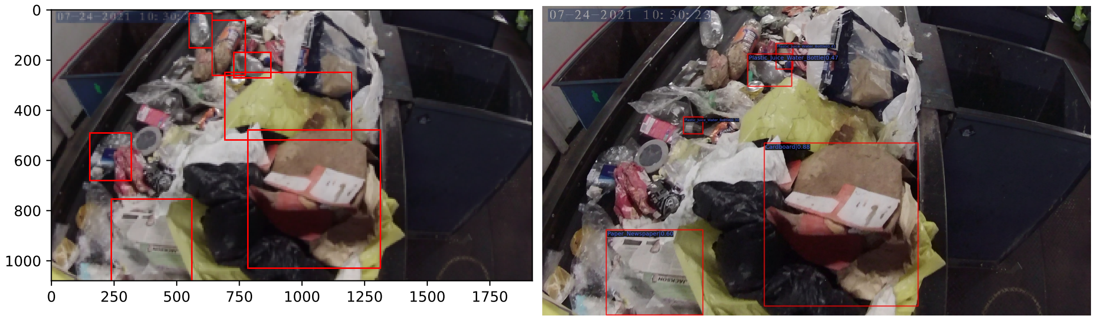
Figure C.12: Example 12 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

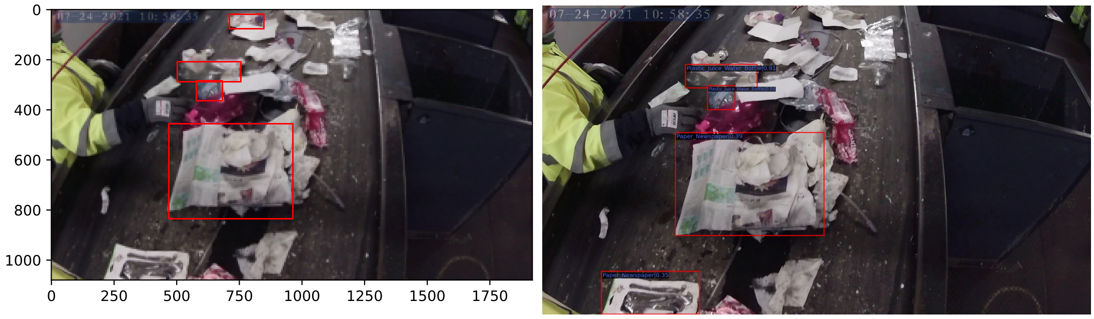
Figure C.13: Example 13 of comparing the ground truth to inference



(a) Ground Truth

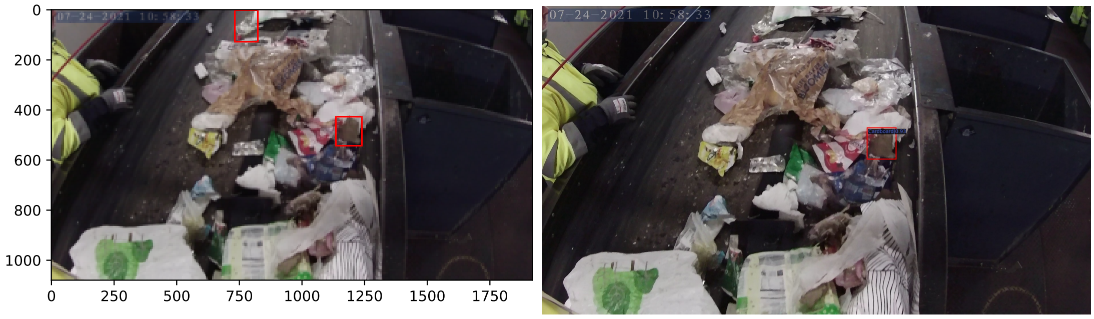
(b) Predicted Output

Figure C.14: Example 14 of comparing the ground truth to inference



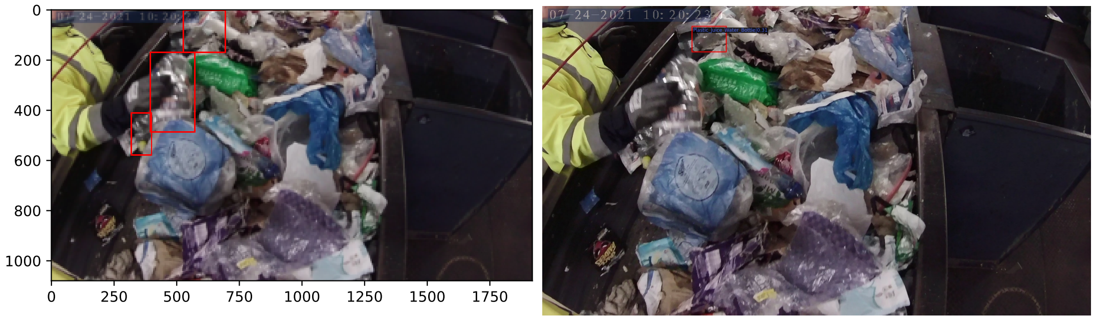
(a) Ground Truth (b) Predicted Output

Figure C.15: Example 15 of comparing the ground truth to inference



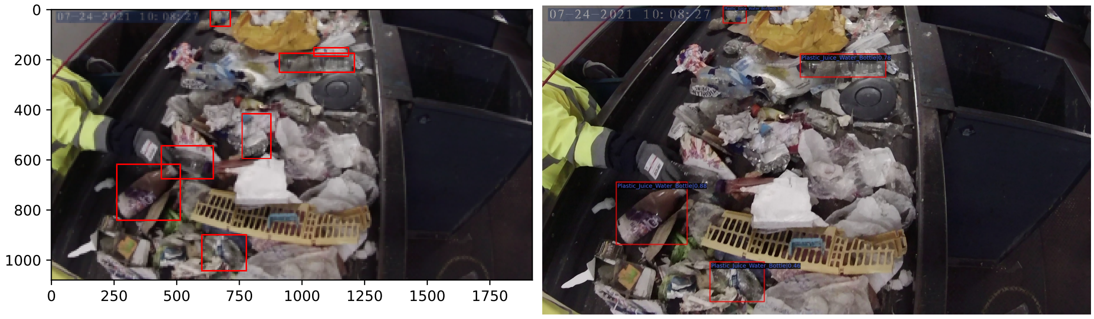
(a) Ground Truth (b) Predicted Output

Figure C.16: Example 16 of comparing the ground truth to inference



(a) Ground Truth (b) Predicted Output

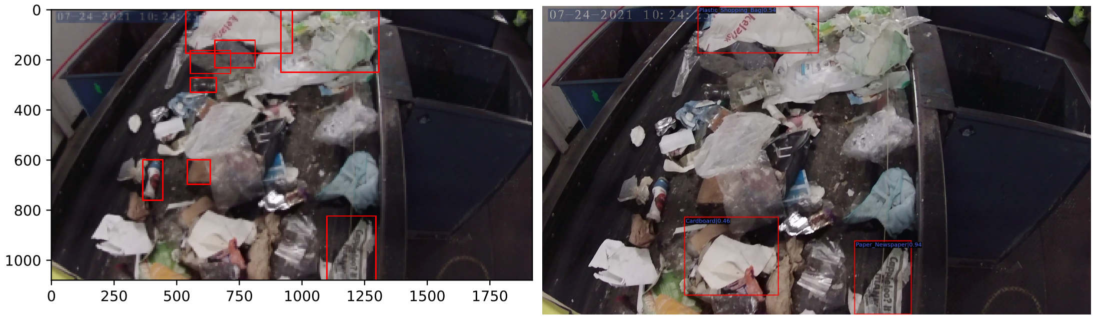
Figure C.17: Example 17 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

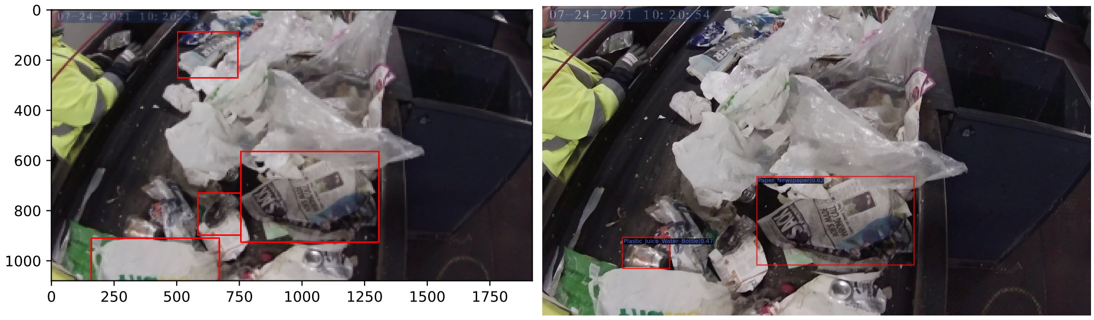
Figure C.18: Example 18 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

Figure C.19: Example 19 of comparing the ground truth to inference



(a) Ground Truth

(b) Predicted Output

Figure C.20: Example 20 of comparing the ground truth to inference