UPDATE: Recyclable Waste Detection on ZeroWaste

Tao Zhang, Zeyu Gu, Zhengqi Dong, Yufan Lin
{mtao, zgu, dong760,eric1025}@bu.edu,

1. Task

Currently the issue of waste detection has drawn the public's attention. Due to a recent report by The World Bank, the waste production is predicted to be 3.4 billion tons by 2050 which would become a disaster [1]. How to deal with waste will be an inevitable problem. So, effectively and accurately classifying waste has become a solution. In this project, our group mainly focuses on the detection of waste. We will use the Zero Waste dataset, which is the largest public waste detection dataset, as a foundation to start our implementation. We aim to use computer vision and deep learning techniques to build a more efficient and accurate model for detection.

2. Related Work

The task of identifying different types of waste in images boils down to the task of object detection. This section presents some state of the art architecture used in object detection[2, 3] as well as some publicly available waste dataset.

2.1 SOTA architectures

There have been several successful architectures proposed during the last ten years that have achieved state of the art results in object detection. Here, we discuss two of the state-of-the-art architectures for object detection.

YOLOv4 YOLOv4[13] is a one stage object detector that builds on previous YOLO models. It uses YOLOv3 as its head, SPP[19] and PANet[20] as the neck, and CSPDarknet53 as the backbone. The main contribution of YOLOv4 is that the authors introduced vision techniques called Bag of Freebies(BoF) and Bag of Specials(BoS) and how the combination of these techniques added to the network can create accurate and efficient object detectors

Scaled YOLOv4 Scaled YOLOv4 [3] is a redesign of the YOLOv4 network based on the CSP [5] approach and is able to be scaled both up and down to accommodate both small and large networks while maintaining optimal speed and accuracy. The CSP is a new way to architect the CNN that can save computations for various CNN networks. The authors of Scaled YOLOv4 propose a network scaling approach that can modify not only the depth, width, resolution, but also the structure of the network [3].

YOLOR YOLOR[21] stands for "You Only Learn One Representation". It is an object detection model, but it is not a continuation of the YOLOv1-v4 series. The idea behind YOLOR is to make the model learn implicit knowledge(knowledge learned subconsciously) and explicit knowledge(knowledge directly learned based on input) and combine these two kinds of knowledge to form a unified network.

Dynamic R-CNN Dynamic R-CNN focuses on adjusting the second stage classifier and regressor to fit the distribution change of proposals. It contains two main parts, the dynamic label assignment and dynamic smooth L1 loss. Compared to Faster R-CNN, these two methods will improve the overall performance of Dynamic R-CNN.[17]
In [7], Bashkirova et al. showed that object detection methods especially struggled with labeling small objects correctly, so this is one aspect we will pay attention to in our project.

2.2 Waste Detection Datasets

As automatic waste detection attracted more attention, several waste datasets [15, 16] have been created for researchers to develop better methodology to tackle the problem.

TACO TACO[15] is an open image dataset of waste in the wild. The TACO dataset currently contains 1500 annotated images with 60 classes[15]. The annotations are provided in the well-known COCO format. TACO is often used to evaluate the performance of object detection models. One downside of the TACO dataset is that it may contain some user induced errors and bias due to the crowd-sourcing nature of the dataset.

ReSort-IT ReSort-IT[16] is one of the more recent datasets created for the purpose of developing better object detection models based on deep learning. It contains 16000 synthetic images. The dataset is

publicly available on Github. One downside for this dataset is its synthetic nature, which may differ a lot from real world waste site scenarios.
We discuss our choice of dataset in section 4.

3. Approach

Object detection is to detect the classes of an object and its location information in the given image. In the ZeroWaste paper, authors have applied three detection models, RetinaNet, MaskRCNN, and TridentNet, to show the feasibility of this approach. The best result was shown in the RetinaNet model with AP=24.2, AP50=36.3, AP75=26.6, AP_s=4.8, AP_m=10.7, and AP_l=26.1, which will be used as the baseline model for performance comparison in our project.
Our work will be extended based on the ZeroWaste project and push the limit to the next level. Our approaches will first mainly focus on Object Detection. After achieving a desirable accuracy and detecting speed on `ZeroWaste-f`, the fully-supervised dataset, we will then consider to other techniques or approaches to further improve the performance, which potentially can make the training more efficient and scalable to the real-world image dataset, such as exploring an advanced model in image segmentation, solving the long-tailed problem, low accuracy with a small object, and laborious process of data collection and annotation.
In order to achieve this goal, there are several tasks we plan to implement:

1. Test and determine which is the best one-stage object detection model for our project, by leveraging the trade-off between accuracy and speed amount three models: YOLOv4 [12], scaled YOLOv4 [13], and YoloR [14].
2. Implement and test the two-stage object detection model Dynamic R-CNN based upon the dataset [17].
3. Data collection: What is the accuracy we can get from those models with different throughputs 15 fps, 30 fps, 45 fps, and 60 fps? Report the test/val data size, AP, AP50, AP75, AP_S AP_M, and AP_L on both validation and testing datasets (Read here to learn more about the evaluation metrics, COCO Evaluation Metrics, A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit, Object Detection Metrics With Worked Example)
4. Compare the result we collected from a one-stage object detection model with a two-stage model Dynamic R-CNN, and determine which one is more suitable for our project?

Extended works: After that, we will consider implementing at least one of the extended works described below. The actual accomplishment will mainly depend on the amount of time we have left.

1. Long-tailed problems are commonly seen in object detection, which is related to the imbalanced distribution of data size on categories. Read more about this paper here, Equalization Loss v2: A New Gradient Balance Approach for Long-tailed Object Detection, to see if there is any possible solution to minimize it in our project.
2. As mentioned in the ZeroWaste paper[7], data collection and annotation is expensive and laborious. Read some relevant papers (e.g., Few-shot, A Survey of Self-Supervised and Few-Shot Object Detection, or single-shot object detection) to see if there is any good approach to reduce the amount of training data while maintaining a good performance inaccuracy.
3. In the ZeroWaste paper [7], we noticed that small objects tend to have larger labeling errors than large objects (e.g., APs = 4.8, AP_m=10.7, AP_l=26.1). Therefore, we plan to read some relevant papers and see if there is any good approach to improve the accuracy on small objects. (e.g., Feature Pyramid Network, IPG-Net: Image Pyramid Guidance Network for Small Object Detection, or Awesome Tiny Object Detection)

3.1 Implementation of Yolo models

We trained our selected YOLO models(YOLOv4, Scaled YOLOv4, and YOLOR) based on WongkinYiu's PyTorch implementation of these models. We provide the link to these implementations in our repo readme as well as in the reference section of this report. These PyTorch implementations are close mimics of the original Darknet framework, which is highly customizable, allowing us to change the network structure and training options. At the current stage, we made a few modifications to the code for it to work on our dataset. We wrote a python notebook in our repo, which provides detailed instructions on how to set up the environment and train the YOLOv4 model. These instructions also apply to Scaled YOLOv4 and YOLOR.

3.2 Implementation of Dynamic R-CNN

We train and test the two-stage object detection model Dynamic R-CNN on SCC with a GPU node. The model is integrated in MMDetection, which is an object detection toolbox that contains a rich set of object detection and instance segmentation methods as well as related components and modules [18]. The full details of training Dynamic R-CNN is shown in the repo. To implement this two-stage detection model, we basically prepared the environment on the SCC to download the toolbox and tested the model within MMdetection.

3.3 Next Step

Now that we have some preliminary results by running both the one-stage detection model and two-stage detection model, we will shift our focus onto fine-tuning the two best-performing models so far(YOLOR and Dynamic R-CNN) to achieve better results. While trying to improve the mAPs of our model, we will specifically focus on improving the precisions on small object detection as it is a general challenge faced by many object detection models. One of the ways to improve prediction accuracy is through data augmentation. Data augmentation generates new images based on our dataset and will make our model less prone to overfit the training set. Some specific data augmentation techniques on images include random crop, random rotation, and mosaic augmentation. These techniques are especially useful for small object detection. We can also make use of the Bag of Freebies(BoF) techniques mentioned in [13], like adding in CutMix, adversarial learning, and drop block regularization in YOLOR. The PyTorch YOLOR implementation and MMdetection toolbox already provide implementations of a few of the techniques mentioned above, so we can adjust the configuration files to enable the corresponding augmentation techniques. For the techniques that are not provided by default but might be useful, like drop block regularization and CutMix, we will select the ones that are accomplishable to implement ourselves. If time permits, we also plan to make a simple UI that enables us to demonstrate our model detecting waste objects given an image.

4. Dataset and Metric

Our project, following Professor Kate's group's work on automated waste recycling, will be based on the industrial-grade waste detection and segmentation dataset named ZeroWaste, specifically the fully-labeled one `ZeroWaste-f`. The dataset was split into training, validation, and test sets and stored in the widely used MS COCO format for object detection and segmentation using the open-source Voxel51 toolkit [7]. There are 3002 training images, 572 validation images, and 929 test images in the dataset. Major classes include cardboard, soft plastic, rigid plastic, and metal. It will be suitable for fully-supervised learning.

One metric to be defined as a measurement for evaluating success is the **detection precision**, which is expressed in percentages to measure the accuracy of classifying the object compared to the ground truth. Consequently, we expect to reach higher detection precision than the detection methods illustrated in [7].

Other metrics include the **precision-call curve**, which is a plot of precision and recalls at varying values of confidence, and **IoU**, which evaluates the degree of overlap between the ground (gt) truth and prediction (pd). Based on this metric, **Average Precision** (AP) is induced. **AP@α** is Area Under the Precision-Recall Curve(AUC-PR) evaluated at α IoU threshold [8] We expect our model will achieve a higher precision-call curve and average precision than the object detection models stated in [7].

5. Preliminary Result

In this section, we show our experimental results after training each model on the ZeroWaste dataset. In our experiments, we trained each model for 300 epochs to produce comparable results.

5.1 Dynamic R-CNN Result

In our experiment, we used the Dynamic R-CNN network which is based on Faster R-CNN architecture, utilizing dynamic ROI head as ROI head and SmoothL1 Loss as loss function. Also, we used ResNet-50-FPN as the backbone. The below table shows the average precision of the test result after training the network for 10 epochs and 300 epochs respectively.

Dynamic R-CNN	10 epochs	300 epochs
AP@[0.5:0.95]	27.1	22.6
AP50	40.3	33.9
AP75	30.3	25.3

Table 1: Result trained with 10 epochs vs 300 epochs

We can see from the up table that both average precision for 10 epochs is better than 300 epochs.

Because we did not use other's pre-trained models, instead we trained from scratch. So, this problem might be caused by the initialization of the model. In the next step, we will focus on adjusting the hyperparameters such as learning rate, and batch norm in order to fix this problem. In addition, we will use cross-validation to check if our model is overfitting or not. We will split the original training data into training and extra validation sets; then learn model parameters on the training set, test and tune hyper-parameters on the validation set and find the best fitting model.

5.2 One-Stage Detector Testing Result

One-stage object detection models skip the process of region proposal and run the detection model directly over a dense sampling of location, and NMS (non-max suppression) is used to produce the bounding box with the highest IOU for each object. In our project, we applied two different one-stage object detection models, including YOLOv4 and YOLOR. For YOLOv4, we used the configuration based on CSPDarknet53 backbone with leaky relu activation function in convolutional layers. For YOLOR, we used the YOLOR p6 configuration, which uses silu activation function in convolutional layers. All models were trained with 300 epochs using MS-COCO-pretrained weights, and the test results are included in Table 2.

When evaluating the performance of the test dataset, we encountered some issues in getting the correct result from cocoapi, so there are only two statistics that were evaluated manually being reported for this update. In the next stage, we will dive into the issues and try to apply cocoapi to have all six evaluation criteria being computed. For Scaled YOLOv4, due to the incompatibility of file format between PyTorch implementation and the original Darknet version, some errors occurred during training and we still have not conquered them yet, but that result will be expected during the final report.

5.3 Overall Comparison

The experimental results of TridentNet [7], Dynamic R-CNN, YOLOv4, Scaled YOLOv4, and YOLOR indicates that the YoloR has outperformed all other methods by a large amount. The result of YoloR has doubled the performance of TridentNet (best result in [7]) (e.g., AP@[0.5:0.95] from 24.2 to 58.7, and AP50 from 36.3 to 72.2). Therefore, we will use it as the baseline model and devote more time to optimizing the performance, including different approaches mentioned in the extended works in Section 3 and some fine-tuning techniques in Section 3.3.

Model	AP@[0.5:0.95]	AP50	AP75	APs	APm	APl
TridentNet (best result shown in [7])	24.2	36.3	26.6	4.8	10.7	26.1
Dynamic R-CNN	22.6	33.9	25.3	1.6	12.2	24.9
YoloV4	5.37	10.6	N/A	N/A	N/A	N/A
Scaled YoloV4	N/A	N/A	N/A	N/A	N/A	N/A
YoloR	58.7	72.2	N/A	N/A	N/A	N/A

Table 2: Mean average precision on the test set of `ZeroWaste-f` of MS-COCO-pretrained TridentNet, Dynamic R-CNN, YoloV4, Scaled YoloV4, YoloR finetuned on `ZeroWaste-f`. Please refer to Appendix A for the prediction result on the test samples.

6. Approximate Timeline

Task	Deadline	Lead
Customize data preprocessing pipeline in Dynamic R-CNN to perform random crop and random flip	04/15/2022	Tao Zhang
Implement CutMix in YOLO models	04/20/2022	Zeyu Gu
Implement drop block in YOLO models	04/20/2022	Zhengqi Dong
Simple UI for demo purpose	04/15/2022	Yufan Lin
Prepare report and presentation	05/03/2022	all

7. Preliminary Code

Please check out our repository at this link: https://github.com/Boston-University-Projects/EC523_DL_CV_Project

References

[1] World Bank Group. "Global Waste to Grow by 70 Percent by 2050 Unless Urgent Action Is Taken: World Bank Report." World Bank, 24 Sept. 2018, www.worldbank.org/en/news/press-release/2018/09/20/global-waste-to-grow-by-70-percent-by-2050-unless-urgent-action-is-taken-world-bank-report.

[2] Y. Li, Y. Chen, N. Wang, and Z. Zhang, Scale-Aware Trident Networks for Object Detection. 2019.

[3] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, Scaled-YOLOv4: Scaling Cross Stage Partial Network. 2021.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, Mask R-CNN. 2018.
[5] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, CSPNet: A New Backbone that can Enhance Learning Capability of CNN. 2019.
[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. 2018.
[7] Dina Bashkirova and K. Saenko, "ZeroWaste: Towards Deformable Object Segmentation in Extreme Clutter," 2021.
[8] Koech, Kiprono Elijah. "On Object Detection Metrics with Worked Example." Medium, Towards Data Science, 18 Dec. 2021, https://towardsdatascience.com/on-object-detection-metrics-with-worked-example-216f173ed31e.
[9] Papers With Code, Semantic Segmentation Benchmarks, https://paperswithcode.com/task/semantic-segmentation
[10] Papers With Code, Object Detection Benchmarks, https://paperswithcode.com/task/object-detection
[11] L. H. Li *et al.*, "Grounded Language-Image Pre-training," *Arxiv*, 2021.
[12] C.-Y. Wang, A. Bochkovskiy, en H.-Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, bll 13029–13038.
[13] A. Bochkovskiy, C.-Y. Wang, en H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", CoRR, vol abs/2004.10934, 2020.
[14] C.-Y. Wang, I.-H. Yeh, en H.-Y. M. Liao, "You Only Learn One Representation: Unified Network for Multiple Tasks", arXiv preprint arXiv:2105. 04206, 2021.
[15] Proença, P. F., & Simões, P. (2020). TACO: Trash Annotations in Context for Litter Detection.
[16] Koskinopoulou, M., Raptopoulos, F., Papadopoulos, G., Mavrakis, N., & Maniadakis, M. (2021). Robotic Waste Sorting Technology: Toward a Vision-Based Categorization System for the Industrial Robotic Separation of Recyclable Waste. IEEE Robotics Automation Magazine, 28(2), 50–60. https://doi.org/10.1109/MRA.2021.3066040
[17] Zhang, H., Chang, H., Ma, B., Wang, N., & Chen, X. (2020, August). Dynamic R-CNN: Towards high quality object detection via dynamic training. In European conference on computer vision (pp. 260-275). Springer, Cham.
[18] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., ... & Lin, D. (2019). MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 346–361. doi: 10.1007/978-3-319-10578-9_23.
[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, Path Aggregation Network for Instance Segmentation. arXiv, 2018. doi: 10.48550/ARXIV.1803.01534.
[21] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, You Only Learn One Representation: Unified Network for Multiple Tasks. arXiv, 2021. doi: 10.48550/ARXIV.2105.04206.

Appendix A: YoloR prediction and labeling samples on the test set of `ZeroWaste-f` of with the best_overall model trained on `ZeroWaste-f` over 300 epochs

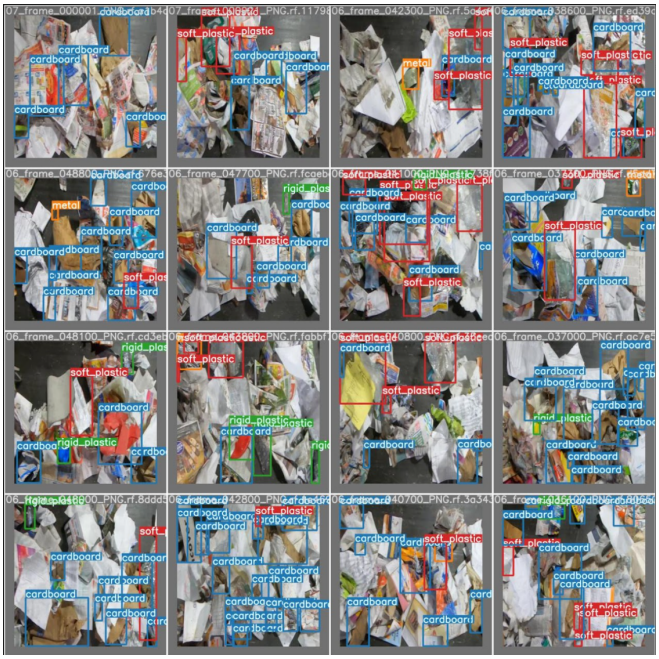

Figure XXX: test_batch2_labels.jpg

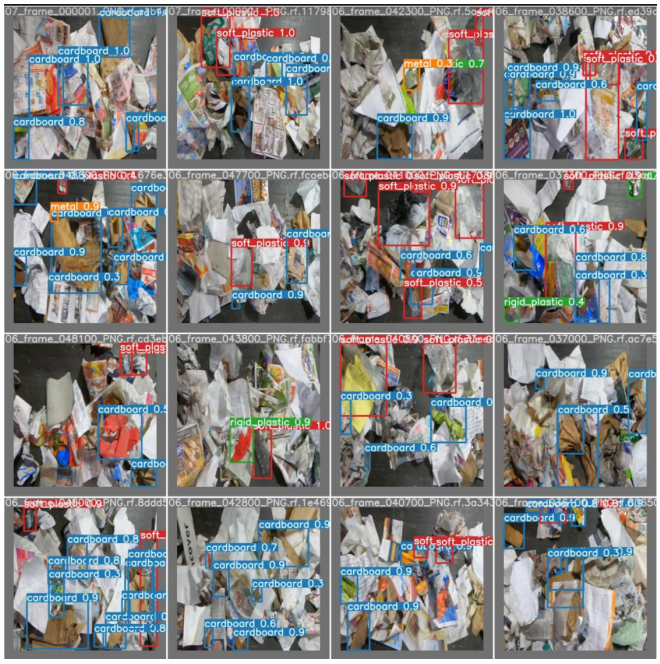

Figure XXX: test_batch2_pred.jpg

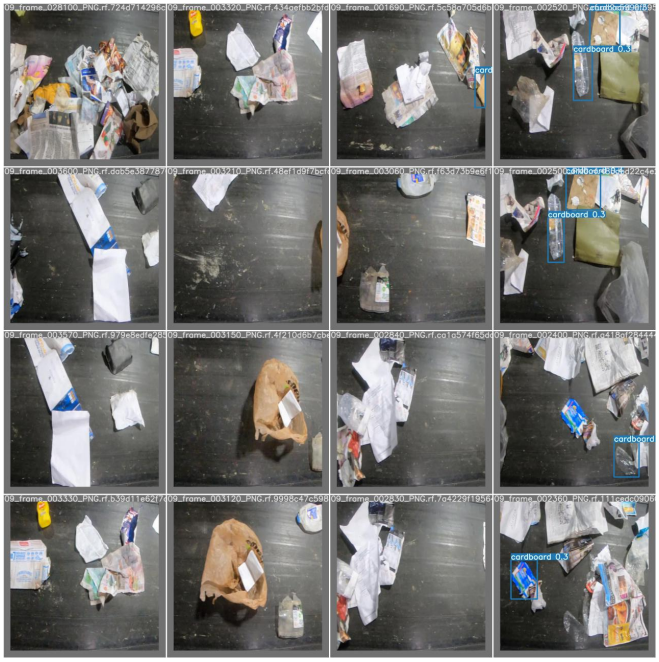

Figure XXX: test_batch2_pred.jpg

Appendix B: YoloV4 prediction and labeling

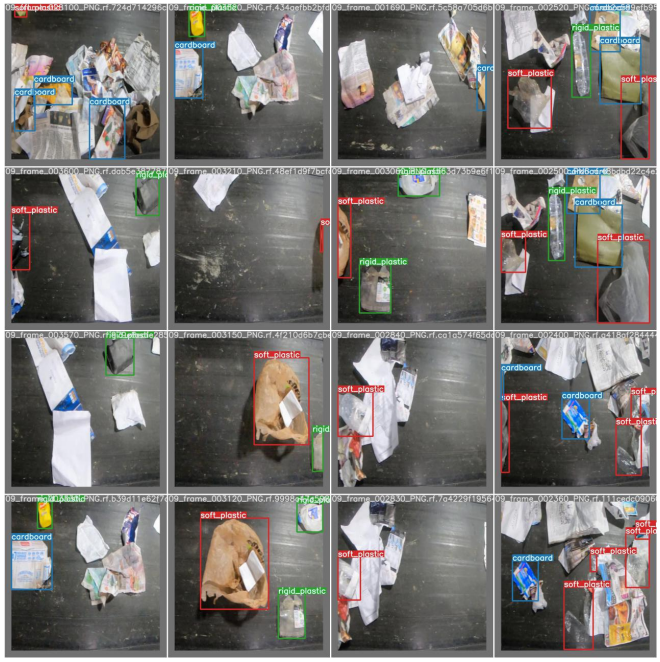

Figure XXX: test_batch2_labels.jpg