WASTE DETECTION & CLASSIFICATION USING DEEP LEARNING TECHNIQUES

George Liu
Student# 1006849341
georgedev.liu@mail.utoronto.ca

Kathy Lin
Student# 1007620166
kath.lin@mail.utoronto.ca

Viet Minh Nguyen
Student# 1007503989
nvmichael.nguyen@mail.utoronto.ca

Chris Shih
Student# 1008298272
chris.shih@mail.utoronto.ca

ABSTRACT

This report will outline the final details of the "Waste Detection & Classification Using Deep Learning Techniques" project. It will include a description and motivation for the project, data processing performed for training the deep learning models, along with testing results and further discussions.

—-Total Pages: 9

1 INTRODUCTION

With the rapid modernization and growth of the world's population, waste and pollutant generation has increased dramatically. The World Bank (Bank, 2022) estimates that the world's annual waste generation is poised to break 3.88 billion tonnes in 2050 - a 73% increase from 2020. Recycling has always been a great option to help us beat this statistic, but with a shockingly low recycling rate of 27% for discarded glass and 8% for discarded plastic (Cho, 2020), there is a clear need for improvement. Cho (2020) mentions a contamination risk if waste is deposited into the wrong bin or sent to the wrong facility, preventing large batches of materials from being recycled.

This project will seek to minimize this risk by leveraging 2 deep-learning models for image detection and classification. The neural network will be able to process input images, reliably locate waste objects present, and then classify the waste into 1 of 10 classes — battery, biological, cardboard, clothes, glass, metal, paper, plastic, shoes, and trash (other non-recyclable materials). This information can easily be applied to support garbage collection tasks, while large batches of waste can now be separated and sent to their respective processing facilities for recycling.

On the other hand, deep learning, more specifically with Convolutional Neural Networks (CNNs), has been a popular choice for solving image detection and classification problems. (Krizhevsky et al., 2012) mention that CNNs have a tendency to make strong, yet accurate assumptions about the nature of images and are easy to train on large datasets, which can be attributed to having fewer parameters and connections. Additionally, with highly accurate and complex pre-built neural networks being so widely accessible, utilizing transfer learning becomes a possibility. All of these factors contributed to the decision to train a CNN for this task.

2 ILLUSTRATION / FIGURE

Our idea is to build 2 separate deep learning models, as illustrated in Figure 1. An image is used as input to both models, in which one will perform object detection to locate garbage object(s) in the image, and one will classify the garbage into one specific class for recycling (e.g., paper, glass, metal, plastic, etc.). The results from the 2 models can then be combined to provide useful information about garbage objects in the image.

As part of an example implementation of the model in reality, the trained model can be applied to portable devices that has a means of collecting object images (e.g., smartphones, VR glasses, etc.). The collected image is then input to the model, resulting in information for fast garbage detection and classification. This is especially useful for people performing garbage collection tasks, to raise their awareness on environmental protection, or sometimes even to give warnings about potentially dangerous objects for people with limited eyesight.

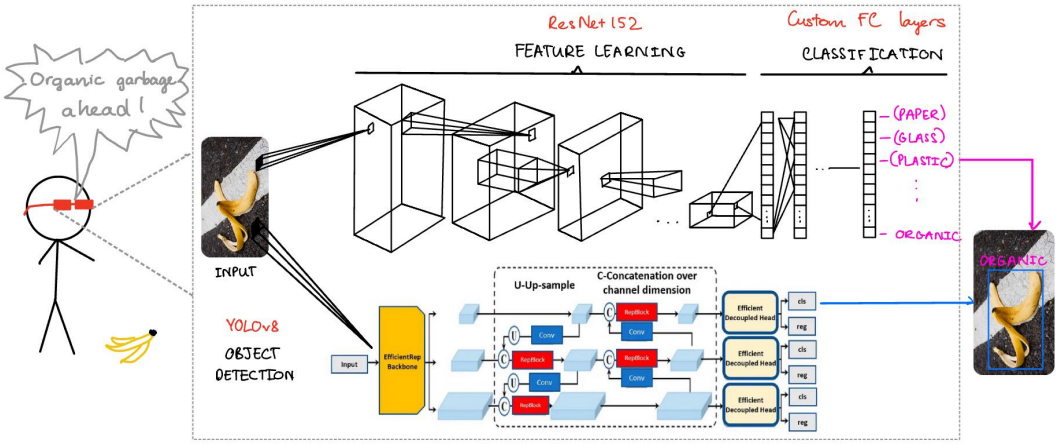

Figure 1: Project idea sketch

3 Background & Related Work

Image detection and classification are two of the most widely researched areas in the field of computer vision. With the recent advancements in hardware, frameworks, and scalability, naturally, deep learning models have become very popular. However, there are still traditional, non neural network-based models that we can use as baselines for our own model. Lin et al. (2011) developed a way of training large image datasets with support vector machine (SVM) classifiers by first leveraging parallel processing techniques and hundreds of mappers to extract features from images, then by utilizing an averaging stochastic gradient descent (ASGD) algorithm. The model achieves a 52.9% accuracy score on the ImageNet dataset — state-of-the-art at the time the paper was published, but completely outclassed by modern neural networks.

As for works on recycling and material sorting, Bircanoğlu et al. (2018) trained the DenseNet family of models (pre-trained on ImageNet) to obtain a 95% test accuracy on the TrashNet dataset. However, due to their focus on real-time implementation, they proposed their own model, RecycleNet, to accelerate inference times. This architecture altered the number of connections in the DenseNet model, offering a 46% inference time improvement on the CPU. As for hardware topics, Li & Grammenos (2022) developed a prototype for a ”smart recycling bin”, leveraging embedded systems (Jetson Nano, K210) and sensors to classify and segment waste at the time of disposal. Sorting materials is also a prominent topic in robotics — Koskinopoulou et al. (2021) proposes a robotic material categorization system trained on the Mask R-CNN architecture, utilizing cameras and sensors to segment, identify, and retrieve waste. Although this project will not be encompassing image segmentation, this remains a relevant topic in regard to the real-world implementation of our image classification CNN.

4 Data Processing

The data for the project consists of two types of datasets: one focuses on object detection while the other focuses on classification.

4.1 Object Detection Data Sources

To perform the object detection task, we chose the Trash Annotations in Context (TACO) dataset (Proença & Simões, 2020), which contains trash images found in various environments (such as clean, indoors, pavement, etc.) and has a good diversity of classes (around 60 classes of various types of trash). Selected examples are provided in Figure 2.

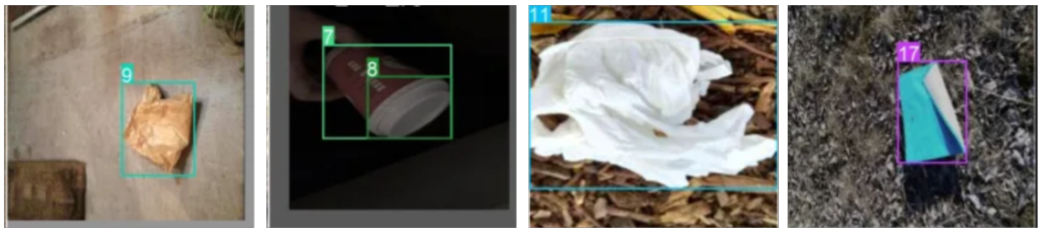

Figure 2: Selected examples from the TACO dataset

The dataset contains tight bounding boxes, class labels, and segmentations in the Common Objects in Context (COCO) format (Lin et al., 2014). Since the Detecto model (Bi, 2020) uses .xml format and the YOLOv7 and YOLOv8 model requires annotations in .txt annotations, we wrote custom code to process the annotations.json and emit the corresponding formats.

4.2 10-Class Classification Data Sources

For classification, we merge two datasets in order to obtain enough data across 10 classes of battery, biological, cardboard, clothes, glass, metal, paper, plastic, shoes, and trash. Each dataset contains images of different types of garbage along with their labels (Table 1). The datasets chosen were selected for the variety of image resolutions and different class distributions to ensure a varied and robust dataset for classification.

4.2.1 De-duplication & Dataset Merging

To merge the datasets, we combined the white-glass, brown-glass, and green-glass categories in Mohamed (2021) into a single glass category and then combined the corresponding of each category into a master dataset. To ensure there are no duplicates, we use the Linux utility fdupes to identify exact matches and remove 2,377 duplicate files as shown in Table 1.

Table 1: Number of examples in each class for each classification dataset

	Raw			Duplicates Removed		
	Mohamed (2021)	Chang (2018)	Total	Mohamed (2021)	Chang (2018)	Total
Paper	1,050	594	1,644	594	0	594
Glass	2,011	501	2,512	488	0	488
Trash	697	137	834	0	0	0
Plastic	865	482	1,347	478	2	480
Shoes	1,977	0	1,977	0	0	0
Metal	769	410	1,179	410	1	411
Cardboard	891	403	1,294	403	0	403
Battery	945	0	945	0	0	0
Biological	985	0	985	0	0	0
Clothes	5,325	0	5,325	1	0	1
Total	15,515	2,527	18,042	2,374	3	2,377

4.2.2 Data Augmentation

Since our dataset has a poor class distribution as shown in Figure 3 despite merging data from multiple data sources, data augmentation as well as trimming the largest classes was necessary to balance the dataset.

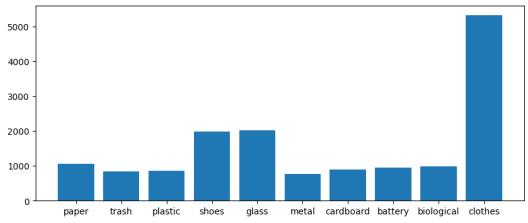

Figure 3: Class distribution prior to data augmentation

To improve the class distribution of the dataset and ensure model robustness, the team augmented the dataset. We applied random cropping, color jittering, rotations, blur, and exposures as shown in Figure 4. After augmentation and trimming the largest classes, each class has 2,000 training samples.

Figure 4: Examples of augmented images

5 Baseline Model

Object detection is a really complicated task so the baseline model was built only for the classifier. The baseline model chosen for the classifier is a traditional Support Vector Machine (SVM) machine learning algorithm, utilizing previously extracted image features to classify images. This algorithm utilizes the Scikit-Learn library, which provides simple APIs for a wide range of machine learning algorithms, including SVMs.

Due to the large size of the dataset and the lack of parallelization and GPU support in Scikit-Learn, the SVM was trained on a subset of our data, with only 75 images for each of the 10 classes (battery, biological, cardboard, clothes, shoes, glass, paper, plastic, metal, trash). To further improve training times, the image data was compressed to 150x150x3, instead of the 224x224x3 size that was used to train our final model. The image features were extracted by compressing the 3D image matrices into a single 1D array of pixel values.

Utilizing the Pandas library, these arrays were mapped to their corresponding labels, split into training and test sets, and processed by the SVM classifier. The baseline model achieved an accuracy of 46%, which is better than a random process, which should theoretically result in a 10% accuracy. The confusion matrix for the testing data on our 10 classes is shown below (Figure 5):

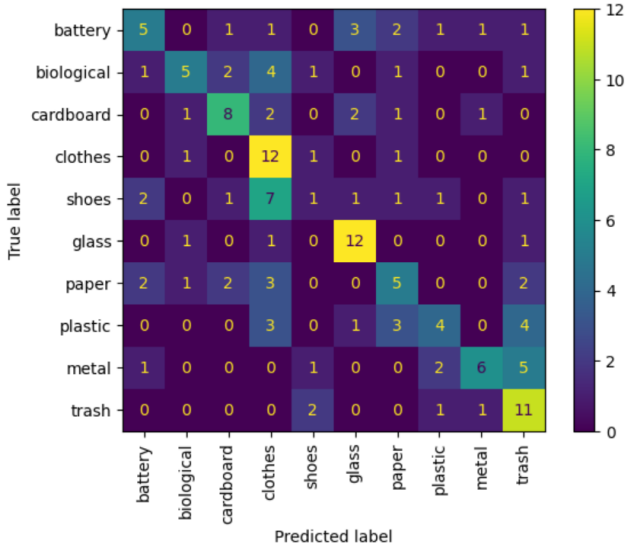

Figure 5: Confusion matrix for the SVM baseline model

6 ARCHITECTURE

Our team uses two separate models to detect (Detector) and classify (Classifier) waste objects in an image. Even though there are datasets available to train one model to do both tasks, we decided to try this new approach to have more control over the data processing and model construction.

To save training time and to achieve better results, we explored various pre-trained architectures for the Detector, all of which were built based on multiple CNN layers and trained on large datasets. Therefore, our task was to pre-process our custom garbage datasets and use thse to fine-tune these models (as mentioned in Section 4). The models tested includes AlexNet (Krizhevsky et al., 2012), Detecto (Bi, 2020), and You Only Look Once (YOLO) v7.

However, based on the behaviour of these models, we finally chose YOLOv8 architecture (Jacob Solawetz, 2023). YOLOv8 is the state-of-the-art pre-trained model for object detection. Even though the details of the model are not published, there are researches showing that it contains multiple CNN layers, along with Anchor free detection technique that effectively reduces the number of box predictions and increases training efficiency (Figure 6). Moreover, the model was trained on Common Objects in Context (COCO) dataset (Lin et al., 2014), which is diverse and well augmentated with techniques like mosaic augmentation (Dwyer, 2020), which helps generalize the task well on test data.

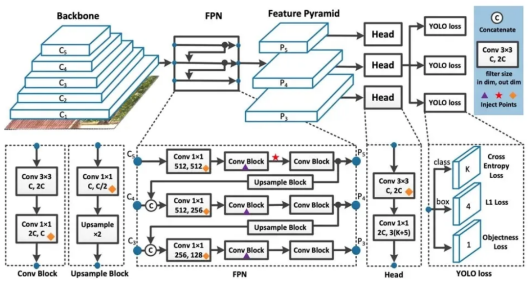

Figure 6: YOLOv8 architecture (Ali, 2023)

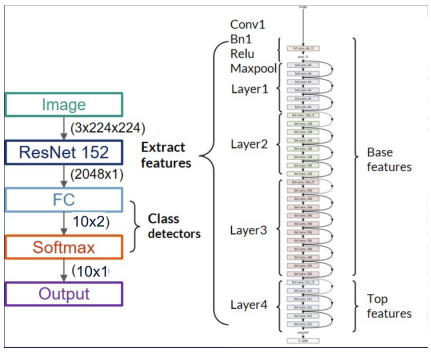

Figure 7: ResNet152 architecture (Shi, 2020)

As for waste classification, we tried both ResNet18 and ResNet152 models for feature extraction, and ResNet152 was used for the final model due to better overall generalization (Figure 7). For this task, we pass the waste images with dimension $3 \times 224 \times 224$ into ResNet152 with default weights and pre-trained layers frozen, since our dataset contains garbage images, which are generic and quite similar to ImageNet-1k dataset which ResNet152 was trained on (Face, 2020). We modified the last fully connected layer of ResNet152 to have output dimension of 10 in order to match the number of garbage classes we want to classify. The prediction produced by the last fully connected layer was then compared to the truth label (one-hot encoded representation of each classes) using Cross Entropy Loss function that works well with multi-class classification.

In the end, we display the bounding box outputted by the detection model, and get the material class outputted by the classification model.

7 Results

Several quantitative and qualitative measures are used to evaluate our model.

7.1 Quantitative Results from Training and Validation

We will discuss quantitative results from training and validation for the detector and the classifier in the next two sections.

7.1.1 Detection Model

For the detection model, the model-generated bounding boxes and the labelled ones do not need to exactly match, and rather we want to measure how close they are, so we chose loss as our metric. The detection model with YOLOv8 approximately achieved training loss of 0.66 and validation loss of 0.97 by training with batch size of 16 and learning rate of 0.001 for 40 epochs (Figure 8). The decreasing trend in both training and validation losses indicates that our model is effectively learning to detect waste objects.

7.1.2 Classification Model

For the classification model, we want to know whether the model is able to classify correctly, so accuracy was chosen as our metric. ResNet152 achieved training accuracy of 97% and validation accuracy of 94% by training with batch size of 64 and learning rate of 0.001 for 10 epochs (Figure 9). Training accuracy shows improvement over epochs, indicating effective learning. However, the validation curve plateaus around epoch 6, suggesting potential memorization of the training images, which might reduce the model's performance for new images.

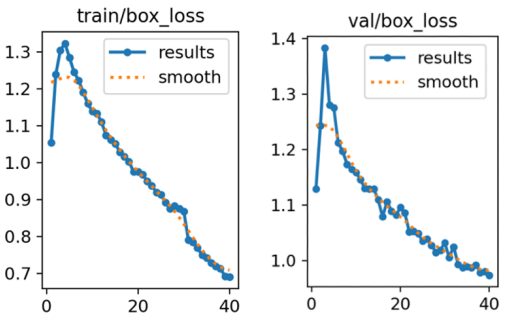

Figure 8: Training and Validation loss for the detection model

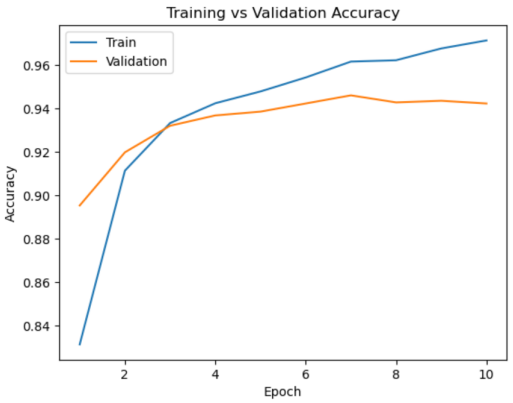

Figure 9: Training and validation accuracy for the classification model

7.2 Evaluate model on new data

To test our model with new data, we collected 275 images in total for the 10 classes that are closer to real life waste than the training datasets. See Table 2 for the number of images for each class. Since we only have the class labels, we did not get quantitative results for the detection model, but the qualitative results for the detection model and both the quantitative and qualitative results for the classification model are discussed below.

Table 2: The number of images for each class in the collected data

Battery	Biological	Cardboard	Clothes	Glass	Metal	Paper	Plastic	Shoes	Trash
33	20	18	24	25	32	32	31	32	28

7.2.1 Qualitative Results for Detection Model

For the detection model, clear plastic images showed interesting results. The rightmost plastic bottle in Figure 10 was detected correctly. However, for the other two images, the model detected a wrong object (the label on the plastic package) or nothing. We can see that the detection model is having difficulty finding the bounding box for clear plastic objects.

Figure 10: Successful plastic bottle detection (left), misdetection of the label instead of the clear plastic (middle), and no detection (right)

7.2.2 Quantitative Results on Classification Model

For classification, ResNet152 achieved the accuracy of 60%, which is 34% lower than validation accuracy. This is difference might be caused by the fact that our collected images are closer to real life images, while the training datasets resemble online images more closely. Figure 11 shows the accuracy for each class with ResNet152 using the collected images, with the shoes class achieving the highest accuracy of 99% and the metal class achieving the lowest accuracy of 87%. The accuracy for each class is higher than the overall accuracy as the collected data is not balanced as seen in Table 2.

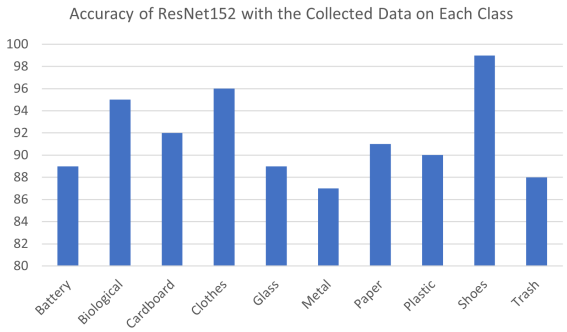

Figure 11: Testing accuracy for each class with the collected images on ResNet152

7.2.3 Qualitative Results for Classification Model

The model was able to classify a plastic bottle that looks similar to a glass bottle in the left of the Figure 12 correctly. However, the model failed to classify tissue paper on the right of Figure 12 as trash. The model classified this tissue paper to paper class which indicates that the model was able to figure out the material the waste is made of but was not able to tell whether the waste is recyclable or not. The fact that trash class has the second lowest accuracy (Figure 11) also indicates the model is having difficulty figuring out whether the waste is recyclable or not.

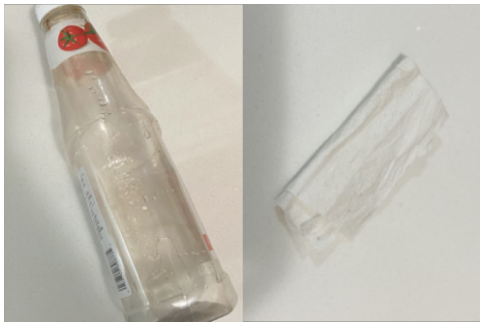

Figure 12: A plastic image resembles glass classified as plastic and a trash image classified as paper

8 Discussion

Our final image classification model outperformed the baseline model by around 15%. The ResNet152 model, with its skip connections and extremely deep architecture, allowed us to learn intricate features and results from our dataset, demonstrated by its high training and validation accuracy. However, the CNN's improvement from the baseline was surprisingly small, which could have been attributed to the datasets that were used to test the models. Due to the lack of GPU support for Scikit-Learn, the baseline was tested only on small subsets of our original dataset, while the CNN was tested on custom, high resolution images that were collected by members of the team.

The ambiguity of certain classes may also have attributed to the unexpectedly low test performance. For example, a dirty tissue could fall under the trash category, due to the low likelihood that it can be salvaged and recycled. However, it could also fall under the paper category. Figure 11 shows a low accuracy score in the trash category, compared to a nearly perfect accuracy score in the shoes category. It can be stated that shoes are a very well-defined category, while trash can come in many forms, some of which may overlap with other classes. In essence, it would be extremely difficult for any dataset to accurately represent the various classes of waste, and this ambiguity can definitely be seen in our final model. To improve our model, we would need to alter our dataset to include well-defined classes, which could entail creating more classes and expanding our dataset with more diverse images of waste.

As for the object detection model, we can see from Section 7.2.1 that our YoloV8 model struggles when the piece of waste blends into its background (it detects no waste in the image). This could be due to, once again, the nature of the dataset, which may not have a sufficient number of images similar to those displayed in Figure 12. However, this could be remedied by training the model for a larger number of epochs.

As seen in Figure 8, validation box loss does not seem to have plateaued after our 40 training epochs, meaning that there could be room for improvement. However, with 40 epochs taking over 1 hour to train, even with GPU support, the feasibility of longer training was greatly diminished. With more time and compute resources, we could train YoloV8 for up to 100-200 epochs, greatly improving our accuracy. It would also be interesting to see how our model would perform on images with multiple bounding boxes present. In our use case, only one piece of waste would be processed at a time, which was reflected in our testing data.

9 Ethical Considerations

9.1 Environmental Issues

The main ethical issue of the system is that it can unintentionally guide users to violate established garbage sorting rules, which could worsen the environment and recycling rates. This is primarily because most of the images we use have a white background, which is not the case in real-world scenarios. This also raises questions about users' environmental responsibilities.

9.2 Privacy

Another ethical consideration is privacy. Privacy concerns might arise due to the datasets we use, which are collected by taking pictures of people's garbage and scraping the web, potentially capturing their personal information. In addition, altering the model to perform unintended tasks, such as detecting names or addresses on labels, can pose a significant privacy risk by enabling the model to collect personal information during inference.

10 Project Difficulty / Quality

This challenging nature of the project stems from two factors: the nature of the classification problem at hand, along with the learning of new model architectures for object detection, which was not covered in lecture.

For the first factor, we needed to create a large dataset (20,000 images) that was sufficiently augmented enough in order to accurately predict the varying visual representations of waste in the real world. With ambiguous classes and large numbers of training samples, our team had to try out multiple CNN transfer learning models, some of which were extremely deep and complicated. Hyperparameter tuning for these models was challenging, especially due to the time required to train them. Even our final model still struggled with predicting the test images that we hand-collected, despite a 94% validation accuracy and a near-perfect training accuracy. The results were testament to the fact that our classification problem was much more difficult than what we had expected, based on our previous knowledge in the course labs. The dataset we had collected was simply not sufficient.

As for object detection, we had to perform extensive research since the topic was not covered in the course. Some of the methods we tried included tweaking AlexNet to provide bounding boxes, but we settled on specialized CNN models like YoloV8 for the task. Training YoloV8 was already a challenge—we needed to organize our images and bounding boxes into the YOLO format, using .yaml files to specify their locations, number of classes, and more. Many of these large data operations caused our computers to max out their RAM and crash, prompting us to learn and try new methods throughout. With how deep and complex the YoloV8 architecture was, we also spent a considerable amount of time on training the model. Additionally, interpreting the results of training was challenging. We were not familiar with mAP (mean average precision) scores for object detection, meaning that our team had to perform research to understand how our model was performing throughout each epoch.

In conclusion, this project provided our team with an intellectually stimulating task that allowed us to branch out of the regular content being taught in lecture, with many challenges to overcome along the way.

11 Link to Colab Notebook

We used Colab Notebook to store all codes for this project, which can be found at `https://colab.research.google.com/drive/1IU5ZWWouHFPTV29eOgz7BSQ_6fBYUvdD?usp=sharing`.

References

Syed Zahid Ali. Principles of yolov8, 2023. URL `https://medium.com/@syedzahidali969/principles-of-yolov8-6a90564e16c3`. Accessed on De-

cember 1, 2023.

The World Bank. Solid waste management, 2022. URL https://www.worldbank.org/en/topic/urbandevelopment/brief/solid-waste-management#:~:text=In%202020%2C%20the%20world%20was,3.88%20billion%20tonnes%20in%202050. Accessed on October 11, 2023.

Alan Bi. Build a custom-trained object detection model with 5 lines of code, 2020. URL https://hackernoon.com/build-a-custom-trained-object-detection-model-with-5-lines-of-code-y08n33vi. Accessed on October 12, 2023.

Cenk Bircanoğlu, Melvin Selim Atay, Fuat Beser, Ozgun Genc, and Merve Ayyuce Kizrak. Recyclenet: Intelligent waste sorting using deep neural networks. 07 2018. doi: 10.1109/INISTA.2018.8466276.

C Chang. Garbage classification, 2018. URL https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification/data. Accessed on October 13, 2023.

Renee Cho. Recycling in the u.s. is broken. how do we fix it?, 2020. URL https://news.climate.columbia.edu/2020/03/13/fix-recycling-america/. Accessed on October 11, 2023.

Brad Dwyer. Advanced augmentations in roboflow, 2020. URL https://blog.roboflow.com/advanced-augmentations/. Accessed on December 1, 2023.

Hugging Face. Resnet-152 v1.5, 2020. URL https://huggingface.co/microsoft/resnet-152#:~:text=ResNet%2D152%20v1.,-5&text=ResNet%20model%20pre%2Dtrained%20on,Recognition%20by%20He%20et%20al. Accessed on December 1, 2023.

Francesco Jacob Solawetz. What is yolov8? the ultimate guide., 2023. URL https://blog.roboflow.com/whats-new-in-yolov8/. Accessed on December 1, 2023.

Maria Koskinopoulou, Fredy Raptopoulos, George Papadopoulos, Nikitas Mavrakis, and Michail Maniadakis. Robotic waste sorting technology: Toward a vision-based categorization system for the industrial robotic separation of recyclable waste. *IEEE Robotics & Automation Magazine*, PP: 2–12, 04 2021. doi: 10.1109/MRA.2021.3066040.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25, pp. 1. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Xueying Li and Ryan Grammenos. A Smart Recycling Bin Using Waste Image Classification At The Edge. *arXiv e-prints*, art. arXiv:2210.00448, October 2022. doi: 10.48550/arXiv.2210.00448.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: Fast feature extraction and svm training. In *CVPR 2011*, pp. 1689–1696, 2011. doi: 10.1109/CVPR.2011.5995477.

Mostafa Mohamed. Garbage classification (12 classes), 2021. URL https://www.kaggle.com/datasets/mostafaabla/garbage-classification/data. Accessed on October 13, 2023.

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

Jing Shi. Actor-action video classification csc 249/449 spring 2020 challenge report, 2020. URL https://www.arxiv-vanity.com/papers/2008.00141/. Accessed on December 1, 2023.