WASTE SEGREGATION USING CNN AND ALEXNET MODELS

Athavan Therunavukarasu
Student# 1005363143
athavan.therunavukarasu@mail.utoronto.ca

Shadman Kaif
Student# 1005303137
shadman.kaif@mail.utoronto.ca

Abdurrafay Khan
Student# 1005215517
abdurrafay.khan@mail.utoronto.ca

Nafio Miah
Student# 1005235522
nafio.miah@mail.utoronto.ca

ABSTRACT

In this report, Group 27 reiterates our project rationale, highlights existing work in the field, explains our data processing, explains our architectures, explains our baseline model, summarizes our results, evaluates our model on new data, and discuss takeaways, ethical dilemmas and our project's difficulty level. —-Total Pages: 8

1 INTRODUCTION

The rise of global temperature in the past decade has been a concern for scientists as its implications and effects can be catastrophic to mankind in the near future [1]. Decreasing the amount of waste at landfills by recycling and ensuring proper waste disposal in households can help mitigate the effects of global warming [2]. Hence, our project's objective is to design a system that aids in and encourages the correct disposal of waste by helping users correctly categorize their waste.

Within the city of Toronto, applications such as TOwaste exist that offer a waste wizard that assists users in classifying their waste [3]. Similar applications exist for different regions, but they are often limited to the contents of their specific databases. These often get outdated, require manual updates, and are not as helpful when identifying how to dispose of products with varying material composition. Hence, we offer a solution where users can take a picture of their waste item, and be directed to the correct method of disposal for the waste.

Our model aims to eliminate the need for manual database updates by utilizing the predictive capabilities of Deep Learning. Convolutional Neural Networks (CNNs) are known for solving image classification problems like ours. Using this would be highly efficient, resolving the need for constant updates to existing databases as our trained model will eventually be able to classify images that have not been seen before.

2 ILLUSTRATION

Figure 1 offers a high-level overview of our system. As we explored 2 main architectures in our project, the Deep Learning Neural Network model represented either our own CNN architecture design, or a standard AlexNet architecture design. We describe these architectures in detail in the Architecture section of this report. The model's input is a 3x224x224 image of a waste item, while the model's output is a disposal bin category. The output of the Deep Learning model gets sent to the Disposal Bin classifier to perform the mapping between waste item and disposal bin.

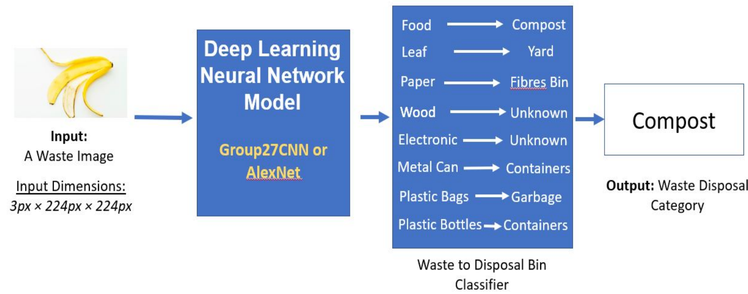

Figure 1: High Level Overview of Model's Input and Output

3 Background and Related Work

The most popular techniques for determining the correct bin for waste are from city resources. Paper notices and the city website are examples of city resources that are used by residents for waste disposal [4]. The resources often contain a list of common waste items and the correct bin to dispose of them. With the adoption of technology by cities, more online tools are created to aid residents with proper waste disposal. The Waste Wizard and the TOwaste App are examples of this in Toronto [5]. These tools allow users to enter the item they are disposing of, and the tools respond back with the appropriate bin to dispose in. Similar tools are available in other cities as well. For example, New York has an online tool that can be found on the Department of Sanitation website that allows residents to enter the disposal item and the website outputs the proper procedure to dispose of the item [6].

Although artificial intelligence is not a common method today for waste classification, there have been developments in the last couple of years to incorporate artificial intelligence in the process. Examples of this are from a paper titled "An Automatic Garbage Classification System Based on Deep Learning" [7] and a design called TrashBot [8]. Both designs are similar and feature a solution that integrates artificial intelligence and hardware to create a self-separating recycling bin. The recycling bin first uses cameras and sensors to scan the waste item. This information is then used by the machine learning model to make the classification. Once the classification is made, the bin selects which compartment the waste goes into. Both designs are also designed so that there are only two classifications, recyclables and non-recyclables.

4 Data Processing

We used the Waste Segregation Image Dataset from Kaggle. The dataset consists of 2 primary categories: biodegradable and non-biodegradable. The biodegradable category consists of the Food, Leaf, Paper, and Wood classes, while the non-biodegradable category consists of Electronics, Metal Cans, Plastic Bags and Plastic Bottle classes. Our model will map these 8 classes into the various methods of disposal such as the Green Compost Bin, the Containers bin, and Garbage.

This dataset was imbalanced as the Food class constituted the majority of the dataset. We presume that this was due to the nature of variability in food's appearance. In addition, Google Colab's GPU limitations limited us on adding training data from many other datasets such as the Trashnet dataset. Those other datasets were then used for testing. We used 9197 images for training, and 3083 images for validation. The remaining images were used for testing. Two images required removal as it caused errors during model training.

Before feeding data into our model, some processing was required. As the images varied in size, we resized all the images to be 224x224x3. A resolution of 224x224 was chosen to ensure that the image contains enough detail to be identifiable while not being too fine to the extent that it results in an inefficient model. Furthermore, the dataset also contained many cartoon images. We believe cartoon images are not the best representation of real-world waste items. However, they can be used to extract high-level features. Hence, we limited their quantity.

To explore the effect of the class imbalance, we created a secondary dataset consisting of duplicated images to normalize the classes. We initially thought of increasing the sizes of the non-food classes, thinking the model would perform better with more data. However, we recognized that this would result in longer training times and we were restricted by Google Colab's GPU limitations. Hence, we had to decrease the size of our Food class, while increasing the non-food classes to move towards a more normalized distribution.

5 ARCHITECTURE

We explored two main neural network architecture models. Figure 2 illustrates the first of the two models which is our own CNN architecture, while Figure 3 illustrates our second model which follows the standard AlexNet Architecture.

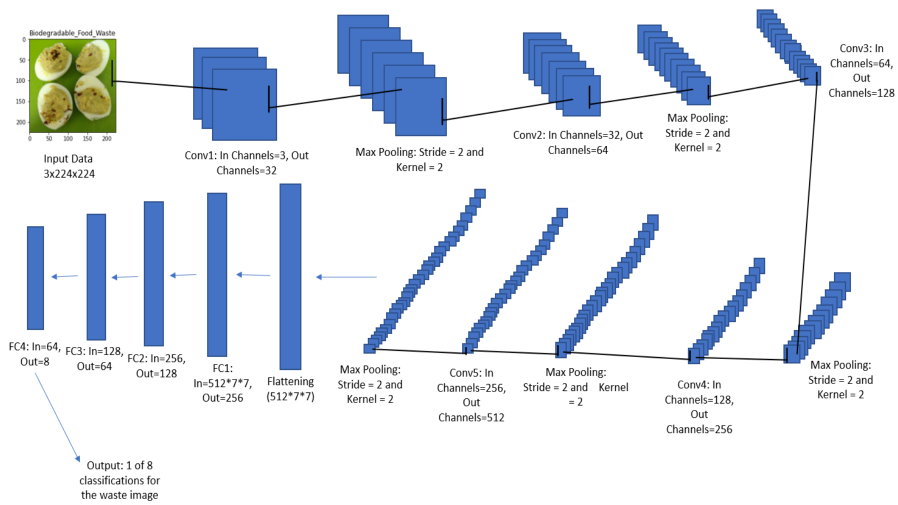

Figure 2: CNN Architecture

As illustrated in Figure 2, our own CNN architecture contains 5 convolution layers, 5 max pooling layers of stride 2 and padding 2, and 4 fully connected layers with ReLU activations. Each convolution layer has a 3x3 kernel filter with stride and padding of 1, respectively.

As we trained our own CNN model to maximize its performance, we thought we could do better. With AlexNet being a pre-trained model that would be much more efficient in terms of performance, we decided to use an AlexNet architecture as well [9].

As illustrated in Figure 3, the AlexNet model has 5 convolutional layers, and 3 fully connected layers. Our first convolution has an 11x11 filter size, the second convolution has a 5x5 filter size, and the 3rd, 4th, and 5th convolution layers have 3x3 filter sizes. We also use 3x3 max pooling with stride 2 after the first, second and fifth convolution. The fully connected hidden layers have 4046 neurons and the output layer has only 8 neurons as we have 8 different classifications.

6 BASELINE MODEL

The team implemented the random forest classifier algorithm as the baseline model that is used to compare against the CNN and AlexNet model. This algorithm is a simple machine learning model created using multiple decision trees. Random subsets of the training dataset are used to create each unique tree. Once the random forest has been built, validation/testing can be carried out to ensure that the algorithm works as intended. Every input that enters the random forest passes through every decision tree. Each tree will then produce a class, and the random forest prediction will be based on the class with the highest number of tallies. The random forest is an effective starting point for classification issues since the low correlation between the models (trees) allows it to ideally settle on

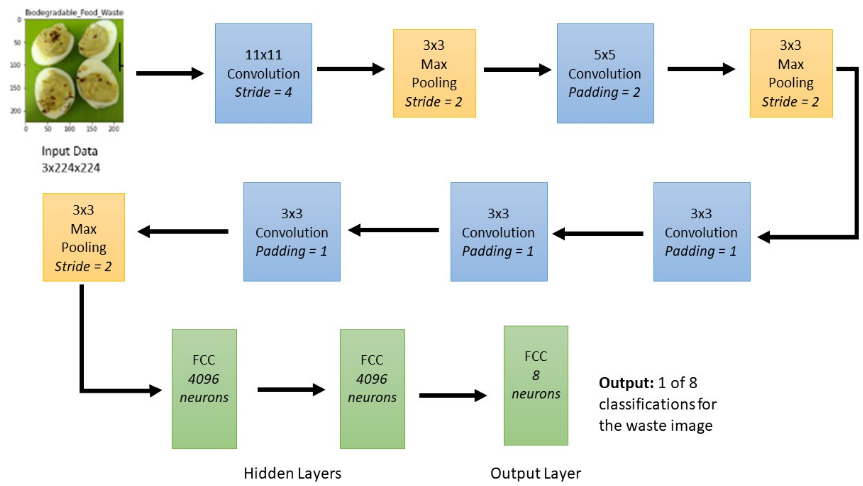

Figure 3: AlexNet Architecture

the right class. This algorithm also works well with large datasets because the random subset done in the beginning stages allows the data to be partitioned.

The RandomForestClassifer modules located in the scikit-learn python package were used to implement the Random Forest. The first step was to load the train, validation and testing datasets. The images in the datasets were then converted from 3x244x244(RGB) image to a single tensor for the classifier to use. The training set was used to train the classifier while the validation/testing set was used for the results. The number of trees is a hyper-parameter that the RandomForestClassifier uses. The increase in the number of trees will create a more accurate classifier as the effect of one tree on the classification reduces which will reduce the chance of bias. The team tested the algorithm using various numbers of trees and discovered that the validation accuracy started to plateau at around 40 trees. 50 trees were used for the final baseline model.

The baseline model achieved a testing accuracy of 81.375% with 50 trees. Therefore, the baseline goal for the CNN and AlexNet model is to achieve a testing accuracy that exceeds 81.375%.

7 Quantitative and Qualitative Results

To measure how our models perform, we used training, validation, and test accuracy as our main metrics. We also found the ratio of correctly predicted classifications for every class. Upon training the graphs, we displayed the train and validation accuracy for every epoch as it shows how our models perform incrementally. Initially, we trained our CNN model with 15 epochs without data duplication and the results are illustrated in Figure 4. The accuracies are calculated as an average over its last 5 trained epochs. For the curve without data duplication, there is fluctuation for the validation curve, which resulted in us performing data augmentation to mitigate the fluctuation and improve steadiness. After data duplication, we notice more plateau for both the training and validation curves, as well as less jagged fluctuation for the training loss. We achieved approximately 4% higher training accuracy, but lost 8.87% validation accuracy, shown in Table 1.

Table 1: **Training and Validation Accuracy for Different Models**

Model	CNN Model without Data Duplication	CNN Model with Data Duplication	AlexNet Model without Data Duplication	AlexNet Model with Data Duplication
Training Accuracy	92.32%	96.31%	90.77%	98.70%

Validation Accuracy	83.69%	74.82%	85.46%	76.74%

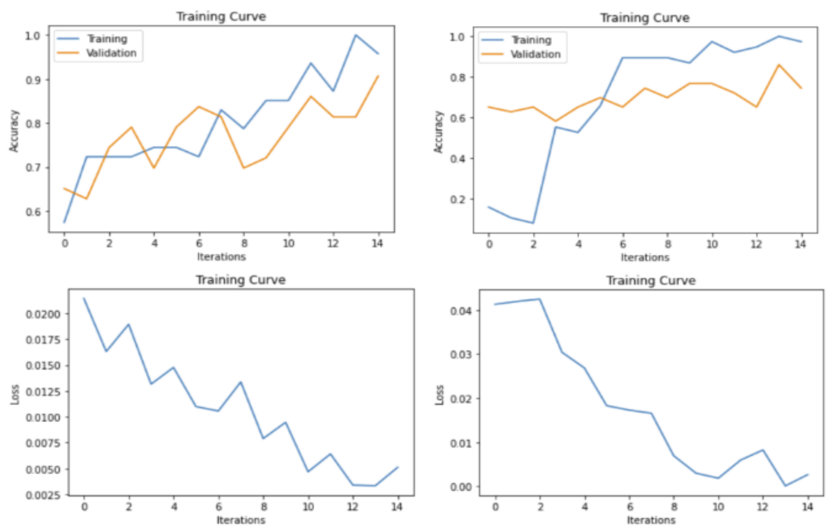

Figure 4: Training and Validation Accuracy Curves, and Training Loss Curves for CNN Model without Data Duplication(Left), with Data Duplication(Right)

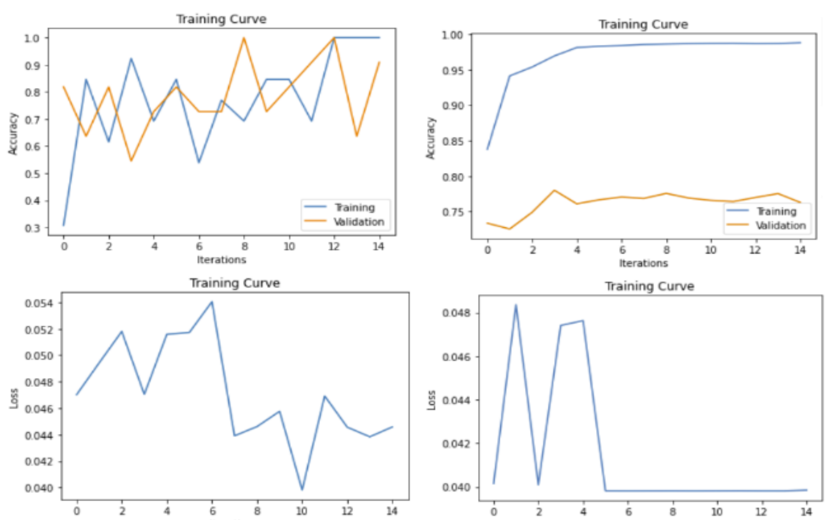

Figure 5: Training and Validation Accuracy Curves, and Training Loss Curves for AlexNet Model without Data Duplication(Left), with Data Duplication(Right)

We also trained two instances of our AlexNet model - one with data duplication and one without. Similar to our CNN model results, we noticed improved steadiness on both training and validation curves after implementing data duplication, see Figure 5. We also gained almost 8% in training accuracy; the trade-off however was that we lost 8.72% in validation accuracy shown in Table 1. It is interesting to see how data duplication increased training accuracy for both our models, yet sacrificed almost an identical 9% in validation accuracy.

After obtaining training and validation accuracy, we found the test accuracy for the models which had the highest validation accuracy: CNN and AlexNet models without data duplication. For the CNN and AlexNet models we attained 85.5% and 94.1% on unseen test data respectively. These results were satisfactory as it demonstrates good classification performance on test data that comes from a different source than our initial training and validation data. We further wanted to see the ratio of correct classification for every class, as that would give us a better idea of which classes obtained the most incorrect predictions.

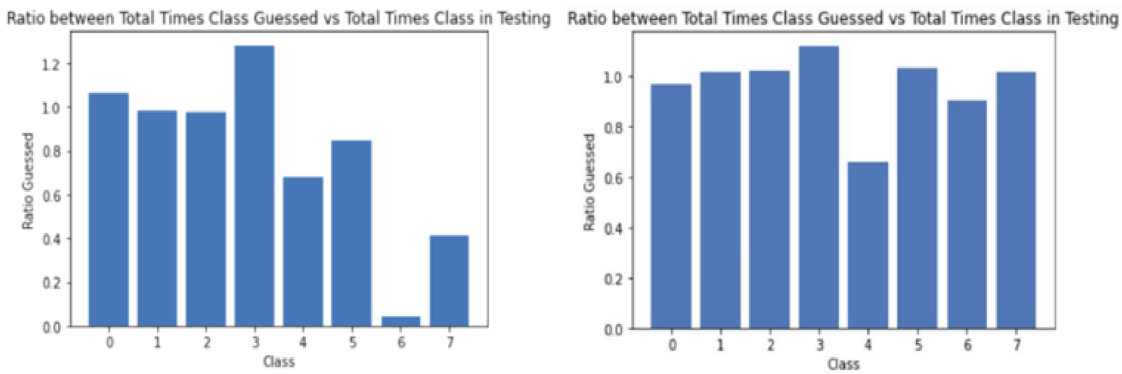

Figure 6: Ratio between the Total Times Classes Guessed vs the Total Times Classes Appear in Test Data for CNN Model (Left) and AlexNet Model (Right)

Per the ratios, we notice most of the classes almost achieved a correct ratio of 1.0, see Figure 6. The CNN model predicted 2636 of 3084 test images correctly and the AlexNet model predicted 2903 of 3084 test images correctly. However, the 4th class (Biodegradable Electronic Waste), had a low ratio for correct predictions. Upon investigating this, we discovered that there were only 234 images of this class in training data and 46 images of this class in validation data, and since these ratios were for the models without data duplication, it makes sense why the 4th class had fewer correct predictions than the others. Further, we looked at the test data and noticed that the range of electronic waste was vast compared to the limited training and validation examples we had.

8 Evaluate model on new data

It is essential to establish good results on never before seen testing data given our model. This is why we wanted to ensure data variety and used the TrashNet dataset as our testing data while using the Kaggle dataset for our training and validation data. Using different datasets for different purposes ensures that our model isn't just memorizing data if it achieves a high test accuracy. We had 3084 images in the test data from TrashNet and achieved 85.5% and 94.1% test accuracy from the CNN and AlexNet models without data duplication respectively. Moreover, due to the vast amount of waste in our daily lives, we also personally created a collection of waste images from every class to verify our models' high test accuracies. We created a dataset of 24 images from our day-to-day lives, selectively choosing 3 images into each class as it allows us to obtain results for every class. Upon testing with the CNN model, we achieved 20/24 correct classifications; our model had 83.33% accuracy on our personal waste images. Two original images that we used for classifications respectively are in Figure 7. A sample output of a correct classification made for non-biodegradable plastic bag waste and a correct classification made for biodegradable leaf waste is shown in Figure 8.

For the other classes, plastic bottles and metal cans classify to the container's bin. Wood waste and electronic waste classify as unknown by the model. Paper waste classifies into the fibers bin. Food waste classifies to the compost. The classes which reported the least accuracy amongst our 24 image dataset were: electronics, metal cans and plastic bags, shown in Figure 9. This is because they had the least amount of training and validation data from the model to learn from. Our model makes it feasible and easy for a user to upload an image onto Google Drive and obtain a bin classification for the waste.

Figure 7: Our Personal Images of a Leaf and a Plastic Bag

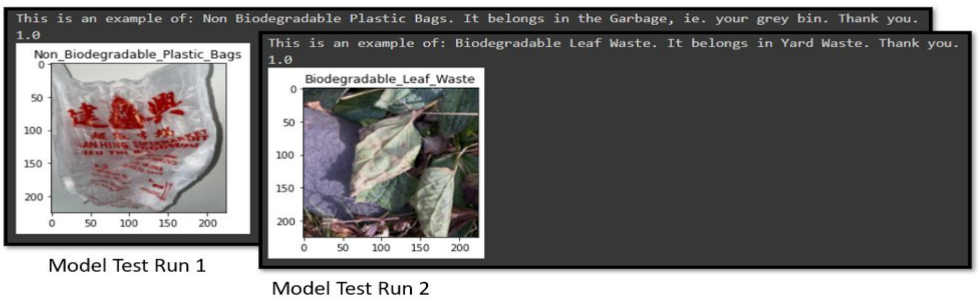

Figure 8: Model Output for Biodegradable Leaf Waste and Non Biodegradable Plastic Bag

Using our own images to test the model's output reinforced our test results, and gives us confidence that the model can successfully classify unseen data with high accuracy.

9 Discussion

One of the measures of success for our neural networks was to compare it to the test accuracy of our baseline model which was 81.375%. The test accuracy achieved by the CNN and AlexNet models were 85.5% and 94.1% respectively. This indicates our neural networks are performing better than the baseline; therefore, the team believes the models are performing well.

Another measure of success for the AlexNet model was for the test accuracy to be higher than that of the CNN model. The team expected this as the AlexNet model uses a deeper architecture compared to our CNN model. Since the AlexNet model was around 10% higher in accuracy than the CNN model, this is an indicator that the AlexNet model is performing well.

A difference between the AlexNet model and the CNN model is the time it took to train the model. Training was significantly shorter for the AlexNet than the CNN as the AlexNet model took around 20 minutes while the CNN model took around 3 hours. Given more time to improve the project, the team would have enhanced the AlexNet model instead of the CNN model due to its higher efficiency and accuracy.

Although we consider the two neural networks were a success, we also acknowledge some of their shortcomings. It was observed that the models are more likely to predict some classes compared to others. Classes such as Food and Leaf are more likely to be chosen than classes like Electronics and Plastic Bags. One reason for this could be due to the features of the classes. Classes like Leaf could potentially have features that are more recognizable by the model than some of the other classes. Another reason for the difference is because of the dataset. The training data used was more biased in the Food class when compared to the others. To normalize the dataset, the team duplicated images

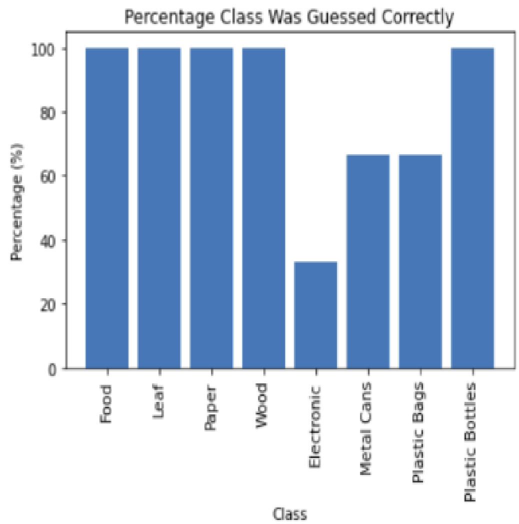

Figure 9: Percent Each Class was Predicted Correctly for our 24 Image Dataset

for classes with less data until the classes had an equal amount of data. However, this led to over-fitting. When comparing the biased model vs the normalized dataset, the biased model performed better in terms of accuracy and balanced prediction per class. However, there was still generalization from the model that needs to be improved.

Given more time, a potential next step would be to perform data augmentation on the datasets. Instead of duplicating images, slightly modifying copies of images from classes with less data may reduce the over-fitting. This is because the new modified copies will be unique to the model while also having similar features to the original image. The team would implement this using the help of the Python module called transforms.Compose.

10 Ethical Considerations

Our project aims to enhance waste disposal applications that exist currently. These applications may need the user's permission to use their device's camera and access their geographic locations to know of the disposal methods of their location. Our project deals solely with the classification of waste item images, hence, gathering of user information that could possibly result in privacy breaches is not required by the application.

In addition, our model cannot claim to successfully classify every waste item due to the impracticality of training on every possible waste item. Training is limited to the images found in the training dataset. For example, our model struggles to correctly classify the disposal of stuffed animals as it is not trained for that task by including a number of stuffed animal images in the training dataset. The model is also only trained to classify images into the 8 categories described by the Waste Segregation Image Dataset. Therefore, misclassifications are possible, and we cannot guarantee that the model can always give a correct classification. Misclassifications may negatively impact the environment which is an ethical concern that we strived to mitigate by maximizing our model's performance.

11 Link to Github or Colab Notebook

Here is the link to the Colab Notebook and Baseline Model Colab Notebook:
Primary Model Colab Notebook, Base Model Colab Notebook

References

[1] S. Kumar and S. K. Himanshu, "Effect of global warming on mankind -a review," Nov 2012.

[2] O. A. Lamma, "The impact of recycling in preserving the environment - researchgate," Dec 2021.

[3] C. of Toronto, "Towaste app," Aug 2022.

[4] C. of Toronto, "Recycling, organics amp; garbage," Aug 2022.

[5] C. of Toronto, "Towaste app," Aug 2022.

[6] N. Sanitation, "Nyc.gov."

[7] G. Li, Z. Kang, J. Yang, and Z. Zhang, "An automatic garbage classification system based on deep learning," *IEEE Xplore*, p. 140019–140029, Jul 2020.

[8] "Trashbot: The smart recycling bin that sorts at the point of disposal," Nov 2022.

[9] S. Sultan and Y. Bekeneva, "A comparative analysis of a designed cnn and alexnet for image classification on small datasets," *Intelligent Distributed Computing XIV*, p. 441–446, 2022.