Audrey Sellers, Jaishankar Govindaraj, Pengru Lin

Automated Waste Sorting System Using Deep Learning: RealWaste Project Report

Problem Statement & Motivation

Image classification, an area that fascinates all the members, is one reason we took this Neural Network class. The beauty of having a computer to classify different images to different groups, coupled with the group's focus on sustainability and a greener future had lured us to pursue a topic in waste classification for this project.

Virtually, every resident, organization, and human activity in the U.S. generates some type of waste. Once generated, wastes must be managed through reuse, recycling, storage, treatment, energy recovery, and/or disposal or other environmental releases. An effective waste sorting system is important for efficient waste management and recycling efforts. Traditional approaches, such as visual inspection, weighing and volume measurement, and manual sorting, have been widely used but suffer from subjectivity, scalability, and labor requirements. However, manual sorting processes are labor-intensive, time-consuming, and prone to errors. In contrast, machine learning approaches, particularly Convolutional Neural Networks (CNN), have emerged as powerful deep learning models for waste detection and classification. By leveraging the RealWaste dataset, we aim to develop a deep learning-based waste sorting system that can accurately classify different waste materials to improve waste sorting efficiency and reduce environmental impact.

Dataset

We came across the UCI Machine Learning Repository, Wollongong City Council. The dataset, RealWaste, contains colored waste images with 4752 instances, with no missing values, and with 9 distinct classes labeled: Cardboard, Food Organics, Glass, Metal, Miscellaneous Trash, Paper, Plastic, Textile Trash, and Vegetation. The images were 524x524, meaning the features were extensive. This is exactly the kind of dataset we want to explore and put what we learned into action, so we pursued this quest without hesitation. Below is the link to this dataset: https://archive.ics.uci.edu/dataset/908/realwaste

Complexity and Handling:

Since this RealWaste dataset is huge, we utilized Boston University's SCC to successfully download and unzip the files, which saved us time and memory from downloading it locally. To handle the complexity of our dataset, we as a group initially decided to work on the project solely on SCC. However, we realized that there were issues with accessing the notebook as well as waiting for availabilities under time constraints. While we still wanted to work on the project on SCC, we also uploaded the file to a shared Google Folder to prevent any further issues with SCC. After plotting a bar chart of the counts of different classes, we realized that class imbalances are another problem to handle. Since some classes dramatically have more instances compared to others, we were afraid of training the model well to predict certain classes better over others. Thanks to Professor Mohannad for providing the group with his insights, we decided to handle this imbalance problem by assigning different weights to the classes. Specifically, the fewer instances of the class, the higher the weight for it.

Analysis Methodology:

KNN:

We opted for a K-Nearest Neighbors (KNN) model with 3 neighbors due to its simplicity and effectiveness, particularly as a baseline model. To accommodate KNN's requirement of one-dimensional arrays, we flattened the image data.

However, the KNN model returned disappointing results with an accuracy of only 22.2%. This low accuracy was reflected across most classes, with generally poor precision, recall, and F1 scores.

Notably, Class 2 exhibited a relatively high recall of 83%, indicating it could identify 83% of actual instances correctly. However, its low precision at 16% suggested that many images not belonging to Class 2 were incorrectly labeled as such.

These discrepancies in performance across different classes underscore the need for a more balanced dataset or specialized techniques tailored to handle specific class characteristics. Further optimization or alternative approaches may be necessary to improve the model's accuracy and robustness.

CNN:

We built a CNN model from scratch with Convolutional Layers (Conv2D) for extracting essential features from the input data and chose to do Max Pooling (MaxPool2D) to reduce the spatial dimensions to focus on important information from the images and aid in computational efficiency. We also incorporated ReLU Activation that way the model can handle non-linearity which we needed for identifying complex patterns and improving feature learning. To address the challenge of class imbalance, Cross Entropy Loss with weighted classes (from pre-processing) was used so that imbalance would not hinder our model further. Stochastic Gradient Descent (SGD) optimization was chosen for its computational efficiency because we needed to handle the complexity of the dataset even though we had a lot of computational limitations.

The 54.5% test accuracy shows that our model is performing very averagely, but with limited computational resources, this was the best architecture for a CNN to run. Our model is performing about 40% better than a random guess would (11%). Our worst run had an accuracy of 30.29%, so this increase is substantial and valuable.

ResNet: We utilized ResNet50, a pre-trained deep learning model, to extract high-level features such as edges, corners, and textures from the data. These features were then inputted into a KNN classifier due to the limited size of the dataset. By leveraging ResNet's ability to capture complex features, the KNN classifier served as a fast and efficient classification model. The resulting confusion matrix showed that the majority of predictions were correct, with a high accuracy rate of 85.6%. This hybrid method demonstrates the effectiveness of combining deep learning with traditional algorithms to achieve significant improvements in classification tasks

Real World Applications

Our real-world applications are mainly related to sorting waste so manual sorting and littering are limited. For waste management facilities, recycling centers can implement this technology into a sorting system to increase customer retention and general recycling rates. Thus, rather than customers going to a recycling facility, and separating their waste by plastic, glass, etc, the customer would dump their waste into a bin that has an automated sorting system. This technology could also be incorporated into waste bins in public settings like streets, malls, and stores, embedding into IoT devices to sort on demand. Although most places have recycling, trash (inorganic waste), and organic waste bins, many people do not appropriately use them, so the waste material is still unsorted. This model would reduce the amount of displaced waste in landfills and recycling facilities.

Limitation and Challenges

Lack of computational resources was the main challenge. The limited amount of free GPU made it extremely difficult to build complex models with multiple layers. For example, we wanted to build a deep-layered CNN to capture data complexity, but the session crashed multiple times, forcing us to construct a model with a simple architecture. To try to mitigate this issue for the next model, we purchased Google Colab Pro and were able to run it, but very timely.

Our dataset also had under 5000 instances, so our training dataset was very small. This means that our models had limited data to train on, potentially making it difficult for the models to generalize to real-world data. This small dataset can also impact our model accuracy as our dataset has very complex patterns. Highly complex data can make it hard for the model to learn patterns with a limited amount of images.

GitHub Link: https://github.com/asellers13/BA865-Group10.git

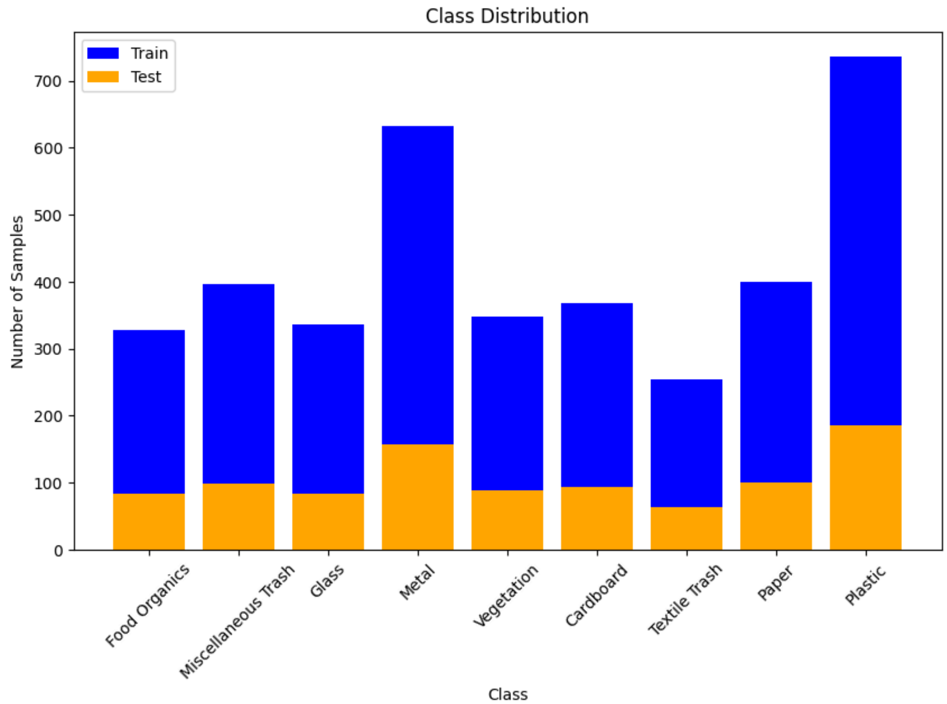
Contribution:

Name	**Coding**	**Theory**
Pengru Lin	Preprocessing	Report, Analysis in colab/SCC & Presentation
Jaishankar Govindaraj	KNN & ResNet	Report, Analysis in colab/SCC & Presentation
Audrey Sellers	CNN	Report , Analysis in colab/SCC, & Presentation

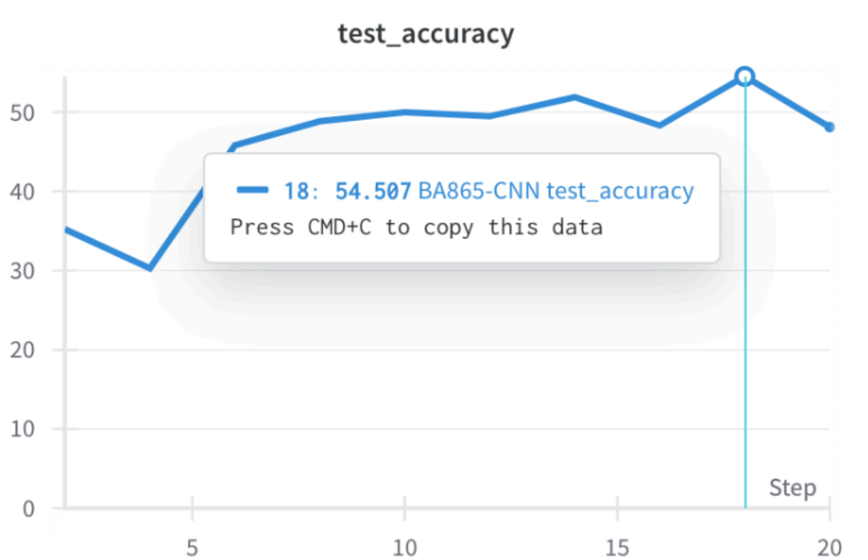
Audrey Sellers, Jaishankar Govindaraj, Pengru Lin

Appendix:

Class Imbalance:



CNN Test Accuracy over 10 Epochs:



Audrey Sellers, Jaishankar Govindaraj, Pengru Lin

ResNet Confusion Matrix: