

Group 14 Progress Report: Municipal Waste Image Classifier

Sunny Yao, Aswin Kuganesan, Yousef Shahin

{yaoh24, kugana7, shahiy3}@mcmaster.ca

1 Introduction

The rapid increase in municipal solid waste has become a major global issue, contributing to landfill overflow, environmental pollution, and the depletion of natural resources. Effective waste management plays a vital role in addressing these challenges by promoting recycling, conserving resources, and reducing the overall environmental footprint. However, traditional methods of waste sorting are often manual, time-consuming, and error-prone, limiting the efficiency and scalability of recycling systems.

This report presents the development of a Municipal Waste Image Classification system that applies deep learning techniques to automate the identification and categorization of waste materials. The system is designed to classify common waste types such as plastic, paper, glass, metal, and organic matter based on image data. By automating the classification process, the project aims to improve recycling efficiency, streamline waste management operations, and support environmental sustainability efforts.

The main challenge of this project lies in the variability and complexity of real-world waste images. Factors such as lighting conditions, background noise, and material degradation make accurate classification difficult. Overcoming these challenges requires robust machine learning models capable of handling diverse input data while maintaining high accuracy and efficiency. The success of this project would demonstrate the potential of artificial intelligence to transform waste management and support more sustainable, environmentally responsible cities.

2 Related Work

Image classification for waste management has gained significant attention in recent years, with various studies leveraging deep learning architec-

tures to improve recycling efficiency. Convolutional Neural Networks (CNNs) have proven especially effective for this task due to their ability to automatically learn hierarchical feature representations from raw image data.

Several studies have demonstrated the potential of CNNs for automated waste sorting. Wang et al. (2019) developed a deep learning-based waste classification system using transfer learning with pre-trained models such as VGG16 and ResNet50, achieving accuracies exceeding 90% on a multi-class dataset. Their work showed that fine-tuning deep architectures on domain-specific waste imagery can substantially enhance performance across diverse material types including plastic, paper, and metal.

Earlier, Yang and Thung (2016) constructed one of the first comprehensive waste image datasets, known as TrashNet, and implemented a custom CNN model for recyclability classification. They identified key challenges in differentiating visually similar materials, such as glass and plastic, and emphasized the importance of data augmentation and preprocessing for improving model robustness.

Building on this foundation, Awe et al. (2020) proposed an ensemble CNN approach that combines the outputs of multiple architectures to improve overall accuracy and generalization. Their ensemble method effectively mitigated misclassification between visually overlapping categories by leveraging complementary feature extraction from different models.

To enhance computational efficiency, particularly for real-time applications, Kumar et al. (2021) introduced a lightweight CNN model optimized for deployment on edge devices. Their framework balanced model complexity with inference speed, enabling efficient on-device waste classification without requiring high-end hardware.

More recently, Alqahtani et al. (2022) integrated deep learning with Internet of Things (IoT) tech-

nologies to develop an intelligent waste classification system for smart cities. Their model demonstrated the feasibility of combining AI-driven image recognition with connected infrastructure to support sustainable urban waste management.

Our approach builds upon these foundational works by implementing a custom CNN architecture that progressively extracts features through multiple convolutional blocks. It incorporates global average pooling to minimize overfitting and dropout for regularization. While our dataset is currently limited in size, our model design follows proven principles from the literature and is optimized for six-category municipal waste classification.

3 Dataset

The dataset used is the Garbage Images Dataset from Kaggle. This dataset was chosen because of the large number of images it provides for the model, and since the dataset contains different types of garbage in folders, which allows the model to get trained on the different types of waste. The dataset is separated into cardboard, glass, metal, paper, plastic and trash. Several preprocessing operations were employed, which involved traversing each subfolder in the GarbageDataset class and collecting the paths for all images along with their corresponding class labels.

A list of tuples was created to store the path and label for all images that had a valid image format of jpg, jpeg or png, and the images were all loaded in RGB format. This ensures that all images has three color channels even if the original images are in a different format, which is important because the Convolutional Neural Network expects a constant number of channels. If the channel dimensions are inconsistent, the training will be prone to error.

Using PyTorch’s torchvision library, the transforms module resized the images to 224 by 224, which is ideal for image processing for the Convolutional Neural Network while also maintaining image quality. The transforms module also converted the images to a tensor, which normalizes the images to a [0.0, 1.0] range. This ensures that the gradient computation is stable, since larger pixel values will be normalized. This also converts the images from the height, width, channel format to a channel, height, width format, since this format is more ideal for the Convolutional Neural Network.

Thirty percent of the dataset is then chosen using random sampling, since thirty percent of the dataset

is sufficient. Random sampling is used to ensure that all garbage categories are equally represented for training. An 80 percent training and 20 percent testing split was used to evaluate the model, and stratification is applied to ensure that the same class distribution is used for both the training and testing datasets.

Finally, data loaders are created for the training and testing datasets to allow for efficient image processing and loading, with batch sizes of 32 and with the training dataset being randomized for each epoch. The dataset was already annotated since the folders included the garbage type, so matching the images to their class label did not require any manual annotation. Each garbage type was assigned to an integer value, which was mapped to each image path. Since the image label can be easily converted to an integer value, additional annotation was not required.

4 Features

The dataset used in this project consists of labeled images of municipal waste categorized into classes such as plastic, paper, glass, metal, and organic waste. Each image is organized within a directory structure, where each subfolder represents one waste category. A custom GarbageDataset class was developed to efficiently load the data by reading image paths and their corresponding labels, returning (image, label) pairs suitable for use in PyTorch data loaders.

Before being fed into the model, all images undergo a series of preprocessing steps using torchvision.transforms. These include resizing the images to a uniform size, converting them into tensors, and normalizing pixel values to ensure consistent scale and distribution across the dataset. This preprocessing helps improve training stability and ensures that the model can handle variations in lighting, background, and image quality.

The model itself is a Convolutional Neural Network, which is particularly well-suited for image classification tasks. Unlike traditional approaches that rely on manual feature engineering, the CNN automatically learns to extract hierarchical visual features directly from raw image data. Early convolutional layers capture low-level features such as edges, colors, and textures, while deeper layers identify higher-level patterns like shapes and object structures. The network then uses these learned embeddings to classify each image into its corre-

sponding waste category.

All feature extraction and classification are performed within this single end-to-end neural network pipeline. By leveraging the CNN's ability to learn rich spatial representations, the system avoids manual feature selection and instead learns robust, generalizable features directly from data. This approach enhances classification accuracy and makes the model more adaptable to real-world waste sorting applications, where visual variability is common.

5 Implementation

Our waste classification model is implemented as a PyTorch neural network comprising four convolutional blocks followed by a classification head. The architecture is designed to progressively extract increasingly complex features from input images while maintaining computational efficiency.

Network Architecture:

The model accepts RGB images of size 224×224 pixels (3 input channels) and processes them through four sequential convolutional blocks:

Block 1 consists of two 3×3 convolutional layers with 64 filters each, followed by ReLU activations and 2×2 max pooling. This block captures low-level features such as edges and simple textures. The use of padding = 1 preserves spatial dimensions before pooling.

Block 2 doubles the feature depth to 128 channels through two convolutional layers, again followed by ReLU activations and max pooling. This block learns mid-level representations that combine basic shapes and textures typical of different waste materials.

Block 3 increases feature depth to 256 channels with a single 3×3 convolution and max pooling, enabling the network to detect higher-order visual patterns related to object composition and structure.

Block 4 expands the feature space to 512 channels through a 3×3 convolutional layer followed by ReLU activation and 2×2 max pooling. This final block consolidates high-level abstractions and provides rich, discriminative features for classification. Including this block improves the model's capacity to handle the visual diversity in waste imagery without excessive overfitting.

Following the convolutional stages, we employ Global Average Pooling (AdaptiveAvgPool2d) rather than standard flattening. This reduces the number of parameters substantially by averaging

each feature map into a single scalar, improving generalization on limited data.

The classification head consists of:

- A flattening layer to convert 2D features to 1D
- A fully connected layer reducing 256 features to 256 hidden units with ReLU activation
- A dropout layer (p=0.5) for regularization during training
- A final linear layer mapping to 6 output classes (cardboard, glass, metal, paper, plastic, trash)

Training Configuration:

The model is trained using the Adam optimizer with a learning rate of 0.001 and Cross-Entropy loss function, which is appropriate for multi-class classification. We employ stratified train-test splitting (80/20) to ensure balanced class representation, with a subset of 1% of the full dataset (139 samples) used for computational efficiency during development.

Data preprocessing includes resizing all images to 224×224 pixels and conversion to tensors. The batch size is set to 32 for both training and testing. Our training loop implements 15 epochs with validation after each epoch to monitor performance.

Performance Characteristics:

The implementation uses PyTorch 2.9.0 and is designed for easy extension to GPU acceleration, though current development is CPU-based. The modular architecture allows for straightforward modifications such as adjusting the number of filters, adding batch normalization, or implementing different pooling strategies.

6 Results and Evaluation

The dataset was divided using an 80/20 split, with 80 percent being used for training and 20 percent being used for testing. Thirty percent of the dataset was used, and the model was trained for 30 epochs using cross-entropy loss to determine the difference between the predicted class and the true label. Here is a simplified version of the cross-entropy loss function used by PyTorch:

$$L = - \sum_{i=1}^C y_i \log(\text{softmax}(x_i))$$

The cross-entropy loss function used by PyTorch implements a softmax function for the predicted

class, and then multiplies the value by the true label for the class using the true label vector. An Adam optimizer was used with a learning rate of 0.001, which is an algorithm that works similarly to stochastic gradient descent and updates the weights of the model using a dynamic learning rate. The algorithm starts with a learning rate of 0.001, but uses different learning rates as the algorithm determines what the ideal learning rate is. Due to the size of the dataset and the simplicity of the model, the runtime to train and test the model was long.

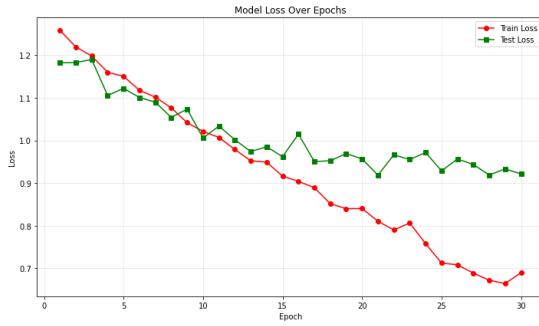


Figure 1: Model loss over 30 epochs for training and testing datasets

Using this methodology, the training loss for the first epoch was 1.25839, and the accuracy was 52.17 percent. The testing loss was 1.18205 for the first epoch, and the accuracy was 54.86 percent. This demonstrated that the model in the first epoch worked fairly well at determining the garbage type, but there was still a lot of room for improvement.

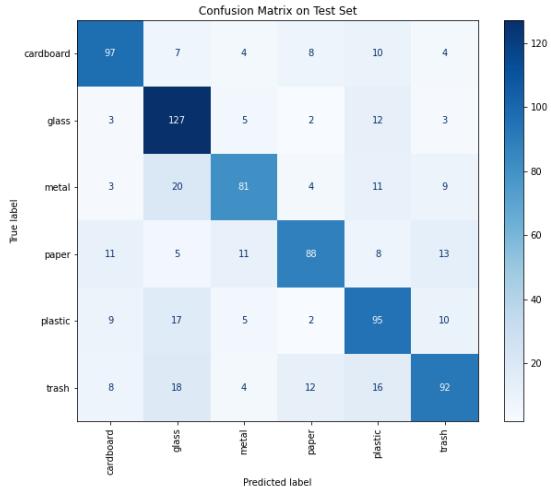


Figure 2: Confusion matrix for the model with 5 categories of waste

As we continued to train the model and as the number of epochs increased, the training and testing loss decreased, and the training and testing ac-

curacy increased. This demonstrated that the model was improving at classifying garbage and was not overfitting to the dataset. This is because the increase in accuracy was steady, and the accuracy values were reasonable. By epoch 30, the training loss was 0.69006, and the training accuracy was 75.65 percent. The testing loss was 0.92142, and the testing accuracy was 70.60 percent.

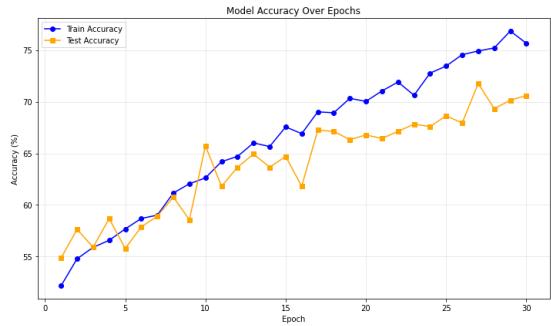


Figure 3: Accuracy over 30 epochs for training and testing datasets

This demonstrates that the model is able to classify the waste fairly accurately, but also demonstrates that the model can still improve. This is because the model will fail to classify images approximately 30 percent of the time. This model is a great baseline for future models. Future changes can include adjusting the architecture and changing the hyperparameters. Using the baseline accuracy of 70 percent, we can also try different architectures and choose to implement architectures that surpass the 70 percent accuracy threshold.

7 Feedback and Plans

For the remainder of the project, our primary focus will be on improving the accuracy, robustness, and generalization ability of our CNN-based waste classification model. Based on the TA's feedback, we will enhance our report by including a framework graph of the CNN architecture to better illustrate our model design. Additionally, we plan to perform feature visualizations to analyze what the CNN has learned and determine whether it is capturing meaningful patterns from the input data. We will also expand our experimental results by including metrics such as training and validation loss, accuracy curves, and other performance comparisons. Furthermore, we will explore other CNN-based models to study their techniques and determine if any can be incorporated into our approach.

Team Contributions

Yousef Shahin: Completed section 2, 5, references, and created the initial model

Aswin Kuganesan: Complete sections 3, 6, citations and equations

Sunny Yao: Completed sections 1, 4, 7, created the confusion matrix generalization and randomized sampling.

References

- H. Alqahtani, M. S. Hossain, and G. Muhammad. 2022. Deep learning-based intelligent waste classification system for smart cities. *Sustainable Cities and Society*, 80:103753.
- O. Awe, S. Mengistu, and C. Ezema. 2020. An ensemble convolutional neural network approach for waste classification. In *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, pages 295–300.
- R. Kumar, A. Singh, and P. Verma. 2021. Lightweight cnn model for real-time waste classification on edge devices. *IEEE Internet of Things Journal*, 8(14):11572–11580.
- W. Wang, Y. Zhang, and X. Li. 2019. Deep learning-based waste classification using transfer learning with vgg16 and resnet50. *IEEE Access*, 7:178966–178975.
- M. Yang and G. Thung. 2016. Classification of trash for recyclability status. CS229 Project Report, Stanford University. Available: <https://github.com/garythung/trashnet>.