# Waste Classification Using CNNs

**James Chen**
Undergraduate Student
Department of Computer Science
University of Wisconsin–Madison
Team 17
tchen424@wisc.edu

**Leobardo Guardado Jr.**
Undergraduate Student
Department of Computer Science
University of Wisconsin–Madison
Team 17
guardadojr@wisc.edu

**Joel Myers**
Undergraduate Student
Department of Computer Science
University of Wisconsin–Madison
Team 17
jcmyers3@wisc.edu

**Emilio Oliveros**
Undergraduate Student
Department of Computer Science
University of Wisconsin–Madison
Team 17
eoliveros@wisc.edu

## 1 Abstract

We explore the potential of deep learning for automating waste classification using the RealWaste dataset, which consists of 4,752 labeled images across nine material categories. We evaluate six pre-trained convolutional neural network (CNN) architecture through a unified training pipeline featuring model-specific preprocessing, feature extraction training, and fine-tuning. Our results show that five out of six of our models outperform the best model in the original paper with Wide ResNet101-2 achieving 92.88% accuracy. Additionally, applying a data augmentation strategy on DenseNet121 pushes the original accuracy up by 1%. These findings highlight the strong potential of deep learning in the field, demonstrating that even simple approaches can deliver impressive results.

## 2 Introduction

As environmental awareness grows, recycling is increasingly recognized for its economic and ecological benefits. However, waste classification remains a major challenge, traditionally relying on manual inspection, measurement, and sorting—tasks that are both labor-intensive and time-consuming. Fortunately, advancements in neural networks have made image classification a common application for machine learning (ML) models, particularly Convolutional Neural Networks (CNNs). This presents an opportunity to develop more efficient and scalable approaches to waste classification.

The application of deep learning models to waste classification is a relatively new research area compared to other existing ML applications. Younis et al. [2] compared six recent studies with various datasets and models used for waste classification. While these studies demonstrated promising results, the field remains in its early stages, with many challenges yet to be addressed. Given the limited research time devoted to this topic, there is considerable potential for further exploration.

In this project, we have chosen to leverage the RealWaste dataset, which was used in one of the most recent studies on waste classification [1]. This dataset is one of the most diverse and comprehensive in the field, containing 4,752 images across nine distinct waste categories. Its diversity ensures a wide representation of different waste types, making it well-suited for training and evaluating deep learning models. By utilizing this dataset, we aim to build a robust classification model that can generalize effectively across various waste materials, ultimately contributing to the development of more efficient and scalable waste management solutions.

Preprint. Under review.

## 3 Data

### 3.1 Dataset

This dataset, sourced from the UC Irvine Machine Learning Repository, consists of 4,752 JPEG images depicting various types of waste. Each image is labeled into one of nine categories: Cardboard, Food Organics, Glass, Metal, Miscellaneous Trash, Paper, Plastic, Textile Trash, and Vegetation.

`https://archive.ics.uci.edu/dataset/908/realwaste`

RealWaste features color images of waste items captured at the point of reception in a landfill environment. The images are provided at a resolution of $524 \times 524$ pixels, with labels indicating the predominant material type.

### 3.2 Preprocessing

To prepare the image data for training and evaluation, we applied model-specific preprocessing pipelines using PyTorch's transforms module. Images were resized based on the input requirements of each model, normalized using ImageNet mean and standard deviation values, and converted to tensors. For testing sets, images were center cropped before normalization.

The dataset was initially loaded without transformations using ImageFolder, then randomly split into 80% training and 20% test subsets. Since random_split does not apply transformations, the resulting subsets were wrapped in a custom class to apply the appropriate transformations during training and testing. Data loaders were then constructed with shuffling enabled for the training set and disabled for the test set.

### 3.3 Augmentation

To further improve model performance, various data augmentations were applied using PyTorch's transforms module. Techniques such as Random Resized Crop, Horizontal Flip, Rotation, Color Jitter, Random Erasing, Random Perspective, and Gaussian Blur were tested through multiple training iterations.

After extensive experimentation, the most effective augmentation pipeline — shown in Figure 1 — included a random resized crop, a random horizontal flip, a random rotation within 20 degrees, and color jitter. This configuration, when used with a DenseNet model, resulted in a 1% increase in accuracy compared to training without augmentations.

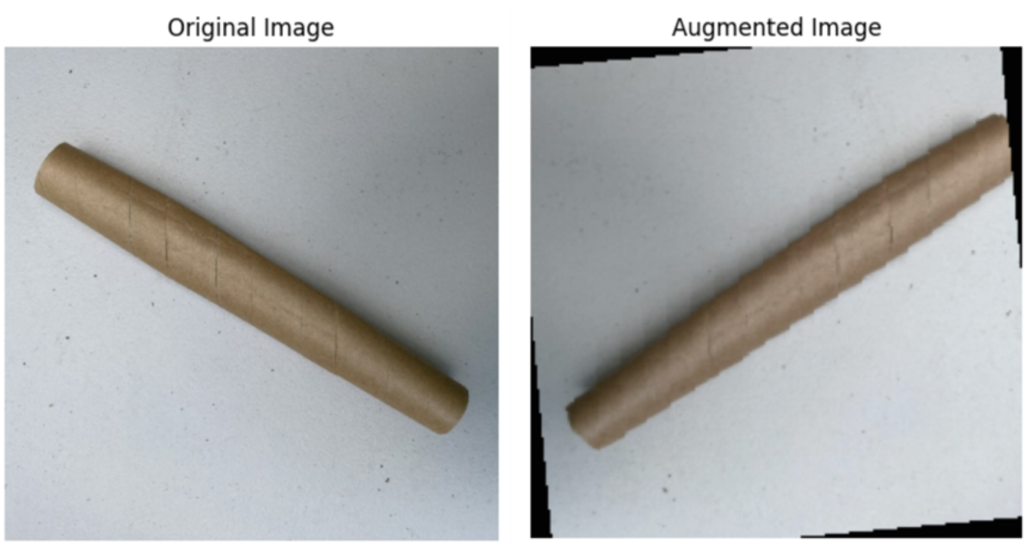


Figure 1: Comparison between original image and augmented image.

# 4 Method

We designed a training pipeline consisting of data preprocessing, data augmentation, and model training to ensure consistency across different models. The specifics of each component are detailed in the following subsections. To evaluate the effectiveness of our pipeline compared to the original RealWaste paper [1], we selected six pre-trained models from the torchvision.models library to undergo the same training process.

## 4.1 Pre-trained Model Selection

Model selection was guided by architectural diversity. Of the six models used, four were also selected and explained in the RealWaste paper [1]:

- **VGG-16** was chosen for its relatively shallow architecture.
- **DenseNet121** offers depth through dense connections.
- **Inception V3** introduces grouped layers for improved feature extraction.
- **MobileNet V2** provides a lightweight model suitable for resource-constrained environments.

Since the torchvision.models library does not include InceptionResNet V2, which was used in the original paper, we added two additional models to increase architectural diversity:

- **Wide ResNet101_2** was included for its enhanced capacity, achieved by widening the network's layers.
- **ConvNeXt Base** was selected for its modern design, incorporating transformer-inspired elements into a convolutional framework.

## 4.2 Data Preprocessing & Augmentation

Data preprocessing was applied across all models to ensure consistency. However, data augmentation was tested only with DenseNet121, which served as our baseline model. Further details can be found in the previous section.

## 4.3 Model Training

All pre-trained models underwent 8 or 10 epochs of feature extraction training, followed by 10 epochs of full fine-tuning. During feature extraction, all model parameters were frozen except for the top classifier layer. Before fine-tuning, all layers were unfrozen to allow full model training. Feature extraction was conducted with a learning rate of 0.01, while fine-tuning used a reduced learning rate of 0.001 to prevent unstable oscillations. Both training phases employed the SGD optimizer with a momentum of 0.9 and a weight decay of 0.0005, which contributed to training stability and reduced the risk of overfitting, and cross entropy loss is selected as the loss function. After each training phase, the training history is recorded and reviewed to ensure a smooth and consistent learning trend.

## 4.4 Evaluation

After training, performance metrics, including accuracy, F1 score, precision, and recall (both macro and micro averages), were reported, along with a confusion matrix to provide a detailed evaluation of model predictions.

# 5 Results

All of the following results were computed from a 20% held out testing set of RealWaste images. Top-1 accuracy and macro-averaged precision recall and F1-scores are reported for each model along with detailed per-class performance of each model.

Table 1: Macro-averaged performance metrics for all fine-tuned models and baseline. Highest performance in each metric bolded.

| Model | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| **RealWaste Paper** | | | | |
| InceptionV3 | 89.19% | 91.34% | 87.73% | 90.25% |
| **Our Implementations** | | | | |
| DenseNet121 (baseline) | 91.14% | 91.40% | 91.46% | 91.38% |
| DenseNet121 (augmentation) | 92.05% | 91.99% | 92.36% | 92.09% |
| InceptionV3 | 91.67% | 92.10% | 91.22% | 91.62% |
| ConvNeXt | 92.22% | 92.61% | 92.07% | 92.29% |
| MobileNetV2 | 91.62% | 91.69% | 91.56% | 91.54% |
| VGG16 | 86.65% | 86.52% | 86.92% | 86.65% |
| Wide_ResNet | **92.88%** | **92.69%** | **92.61%** | **92.53%** |

### 5.1 Aggregate Model Performance

The results in Table 1 show that overall Wide ResNet achieved the best performance in every metric (accuracy, precision, recall, F1 score) compared to all other models. All of the fine-tuned models achieved a testing accuracy greater than 86%, with only VGG16 having a testing accuracy lower than 90%. Every fine-tuned model outperformed (in terms of accuracy) the best performing model (InceptionV3) from the original paper introducing the RealWaste dataset except VGG16. Data augmentation on DenseNet121 resulted in about a 1% improvement in testing accuracy.

### 5.2 Per-Class Performance

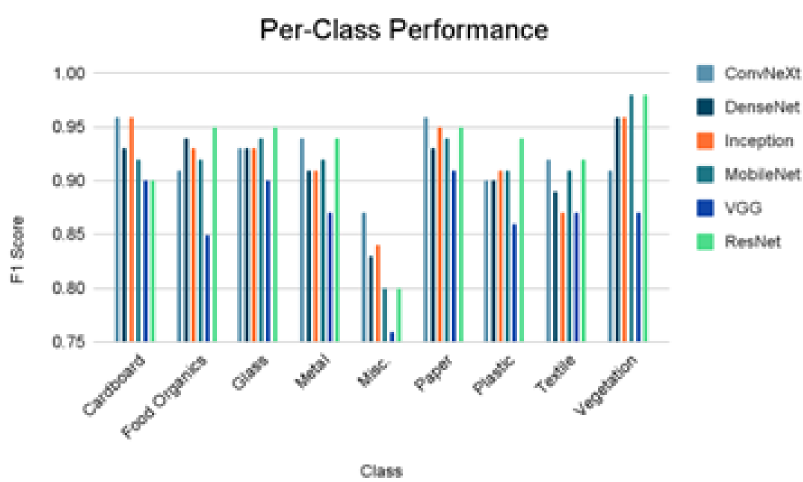


Figure 2: Comparison of F1-scores by class for 6 fine tuned model architectures. Each cluster contains 6 bars, 1 per model

Figure 2 shows that all models performed worst on the Miscellaneous trash class. Vegetation was the most easily identifiable class. The best model, ResNet, relatively underperformed on the classes of cardboard and miscellaneous trash. The second best performing model, ConvNeXt, relatively underperformed on plastic and vegetation. VGG16 had significantly lower F1 scores across the board.

### 5.3 Confusion Matrix of Best Model

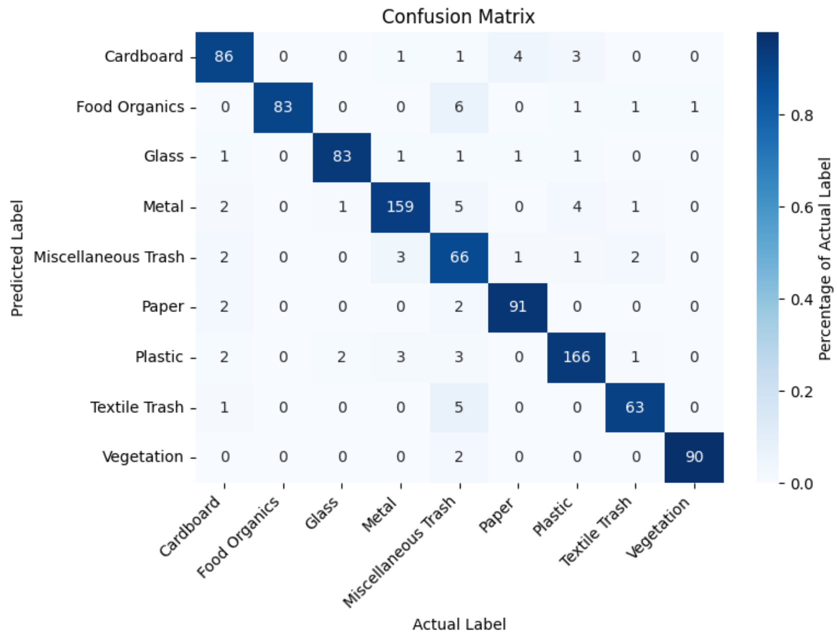


Figure 3: Confusion matrix for fine-tuned Wide ResNet101_2. Computed from testing split.

Figure 3 shows that the most common misclassifications for the Wide ResNet101_2 model were images that were actually miscellaneous trash being identified as other classes (mostly textile trash and food organics). Beside miscellaneous trash misidentification, the largest errors were paper misidentified and cardboard (4 instances) and plastic misidentified and metal (4 instances).

## 6 Discussion

DenseNet121 reached 91.14% accuracy on RealWaste (Precision = 91.4%, Recall = 91.46%, F1 = 91.38%). A simple augmentation suite—random resized crop, flip, rotation, and color-jitter—pushed its accuracy up by 1%, confirming that richer sample variety improves generalization and should lift other models, too.

### 6.1 How the models stack up

| Model | What tips the scales |
|---|---|
| Wide ResNet101_2 | Extra-wide residual paths capture subtle textures. |
| ConvNeXt-Base | Modernized ResNet blocks, strong feature flow. |
| Inception V3 | Multi-scale filters edge past DenseNet. |
| MobileNet V2 | Depth-wise convolutions give speed with little loss. |
| DenseNet121 | Dense links remain competitive but trail the top tier. |
| VGG-16 | Shallow plain stacks miss complex cues. |

Table 2: Summary of model characteristics and performance insights.

The trend is clear when comparing table 1 and 2: depth, width, and smarter blocks unlock gains—Wide ResNet's broad residual channels and ConvNeXt's refined design extract detail that simpler nets overlook. Lightweight MobileNet shows how far efficiency tricks can go, nearly matching DenseNet with a fraction of the parameters. VGG-16, by contrast, struggles against the dataset's visual diversity.

### 6.2 Where every model stumbles

Categories that share colour and shape—notably *Miscellaneous Trash*, *Cardboard*, *Plastic*, and *Metal*—remain the primary sources of error. Increasing sample density and refining labels for these classes looks to be the most promising route for improvement.

A practical route is to integrate *TrashNet*, a publicly available waste-image collection, with RealWaste [1]. TrashNet offers thousands of additional images spanning overlapping material types. By merging the two datasets (after reconciling class definitions), we can (i) expand training coverage for under-represented or ambiguous categories, (ii) and provide the networks with a richer range of intraclass variation. We anticipate that this enlarged dataset will enhance recall—particularly for *Miscellaneous Trash*—and elevate overall accuracy across all models.

### 6.3 Limitations and Future Work

- **Epochs:** We trained for 10+10 epochs for Inception V3 and DenseNet121, and 8+10 epochs for other models. Future studies should increase the number of epochs to ensure more stable training and better convergence.
- **Data Splitting:** A non-stratified split was used during preprocessing, which may have resulted in unequal class proportions across training and test sets.
- **Test Dataset:** The test set was created before RealWaste curation, containing 10% of images from each class selected at random. While diverse, future work could explore alternative splitting strategies.
- **Hyperparameter Tuning:** All models were trained with the same hyperparameters, differing from the original paper's approach. Future studies should examine whether consistent tuning or model-specific tuning yields better performance.
- **Code Oversight:** All model parameters were unfrozen during fine-tuning, contrary to the plan to unfreeze only the final layers. However, this did not cause overfitting, as test accuracy remained stable. Targeted fine-tuning should be explored in future work.
- **Training Time:** Training time was not recorded for each model. Future research should consider the trade-off between training efficiency and performance, particularly for real-world deployment scenarios.
- **Statistical Uncertainty:** Future studies should report 95% confidence intervals (or use bootstrap tests) to distinguish between close model performances (e.g., MobileNet V2 vs. DenseNet121).

Addressing these points, and adding more data for the confusing classes, should push accuracy higher while clarifying which architecture truly leads.

## References


[1] Sam Single, Saeid Iranmanesh, and Raad Raad. Realwaste: A novel real-life data set for landfill waste classification using deep learning. *Information*, 14(12), 2023.

[2] Hussein Younis and Mahmoud Obaid. Performance comparison of pretrained deep learning models for landfill waste classification. *International Journal of Advanced Computer Science & Applications*, 15(11), 2024.