

Contribution

Jaekwang Shin: EDA, Data Preprocessing, Data Augmentation

Emily Zhang: Model Evaluate Function, Decision Tree Model, CNN Model 2, Deepdive of misclassification

Victor Wang: CNN Model 1, Simple CNN model

Github: https://github.com/paloaltojae1/datasci207_section5_team3

Deep Learning Model for Automated Waste Classification

Emily Zhang, Victor Wang, Jaekwang Shin

DATASCI207

UC Berkeley School of Information



Agenda

- 1 Research Question
- 2 Data Exploration and Preprocessing
- 3 Baseline - Decision Tree
- 4 Performance - CNN
- 5 Results and Conclusion



Project Goal

Goal: Develop a deep learning model using the **RealWaste dataset** (4,700+ labeled images).

Motivation: Automated classification can reduce contamination, improve recycling rates, and support scalable waste management solutions.

Expected Outcomes:

- Improved sorting efficiency
- Reduction in manual labor and sorting errors

Research Questions

Leading Question: How effectively can CNNs classify real-world landfill waste images into correct material categories?

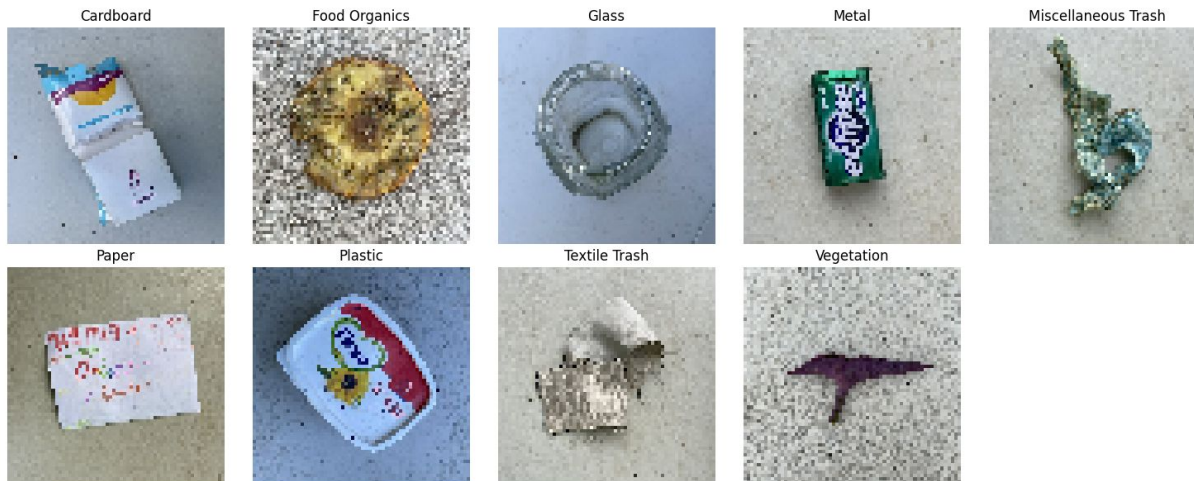
Sub-questions:

- What strategies (**data augmentation**, **advanced architectures**) can enhance accuracy over a baseline model?
- How significant are improvements over simple baseline approaches (e.g., Decision Trees, Majority Class Predictor)?

Objective: Identify the most effective modeling strategies to automate and optimize waste classification in real-world scenarios.

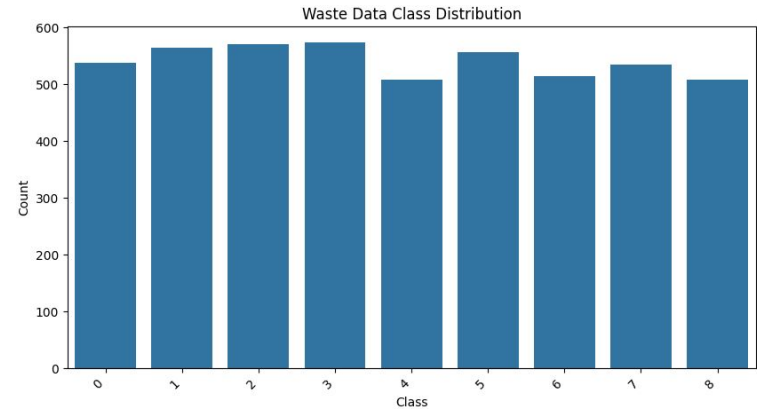
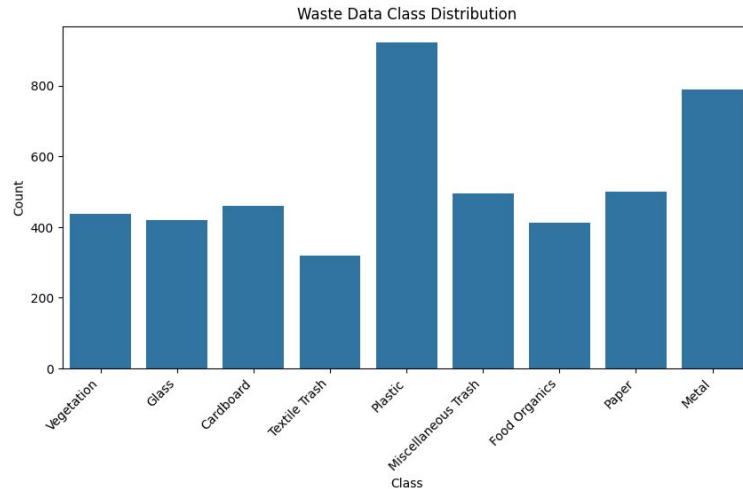
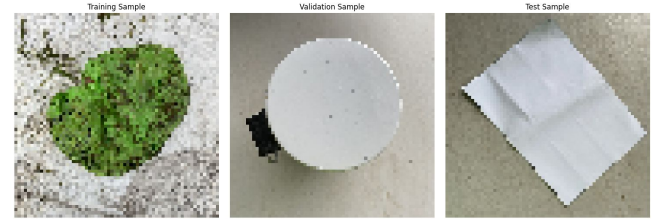
Data Exploration

- ~4,700 labeled images
- 9 waste categories (e.g., plastic, glass, metal)
- Images taken in real-world conditions (different lighting, angles, backgrounds)



Data Preprocessing and Augmentation

- Normalization: Scaled pixel values
- Resizing: Made all images the same size
- Augmentation : rotation, flips, and brightness changes
- Uniform sampling



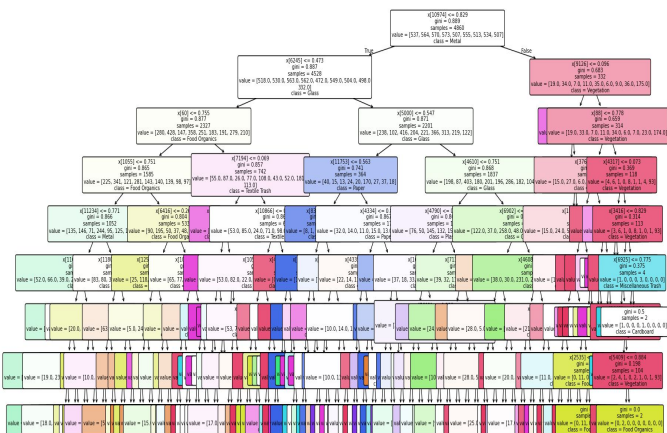
Why Decision Tree

- **Minimal Preprocessing Required**
Can directly accept flattened image pixel data ($64 \times 64 \times 3$) without extensive preprocessing or feature extraction.
- **Interpretability and Transparency**
Provides clear decision rules and pathways, making it easy to interpret initial results and troubleshoot.
- **Quick Training and Evaluation**
Enables fast baseline establishment and rapid feedback loops.
- **Effective Benchmarking Tool**
Offers an accessible baseline to easily compare the relative improvement provided by more advanced models.

How is the Decision Tree Model Performance

- **Hyperparameters:**

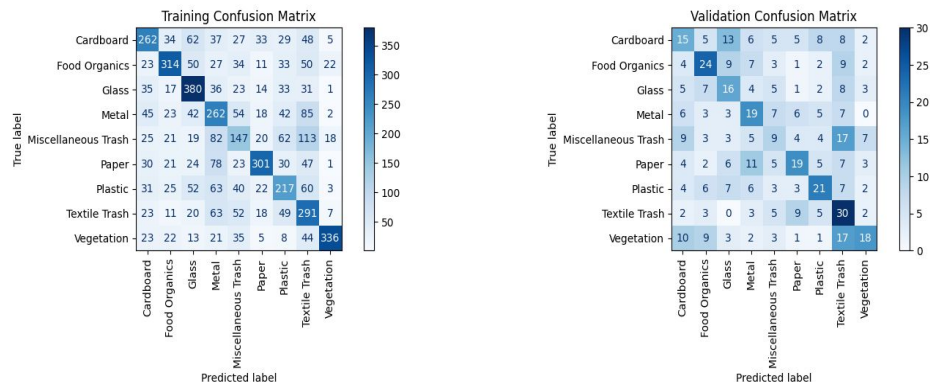
```
max_depths = 8, min_samples_leaf = 2
```



- **Performance**

| Datasets\Metrics | Accuracy | Precision | Recall |
|------------------|----------|-----------|--------|
| Training Data | 51.6% | 53.5% | 51.5% |
| Validation Data | 31.7% | 32.5% | 31.9% |

- **Confusion Matrix**



Limitation of Decision Tree Model

- **Poor Performance on Image Data**

Struggles to capture spatial relationships and visual patterns across pixels.

- **High Dimensionality Issues**

Flattened image inputs lead to sparse and noisy representations.

- **No Feature Hierarchy**

Lacks ability to learn layered visual abstractions (e.g., edges → shapes → objects).

- **Overfitting Risk**

Decision trees can easily overfit to pixel-level noise without regularization.

- **Limited Scalability**

Not well-suited for scaling to large or diverse image datasets like RealWaste.

Why CNN

- **Positional Invariance**

CNNs can recognize patterns regardless of where they appear in the image.

- **Learns Visual Patterns**

Automatically extracts low-level features (like edges, textures) and combines them into high-level concepts.

- **Robust to Real-World Variability**

Handles variation in lighting, angles, and background noise present in the RealWaste dataset.

- **Improves Over Baseline Models**

Outperforms decision trees by learning features directly from pixel data, without manual engineering.

- **Suited for Image-Based Classification Tasks**

Designed for vision problems like waste categorization, where spatial structure matters.

Preliminary CNN Model Structure

Hyperparameters: ADAM optimizer with learning rate of 0.01. Trained for 2 epochs

Model: "sequential_2"

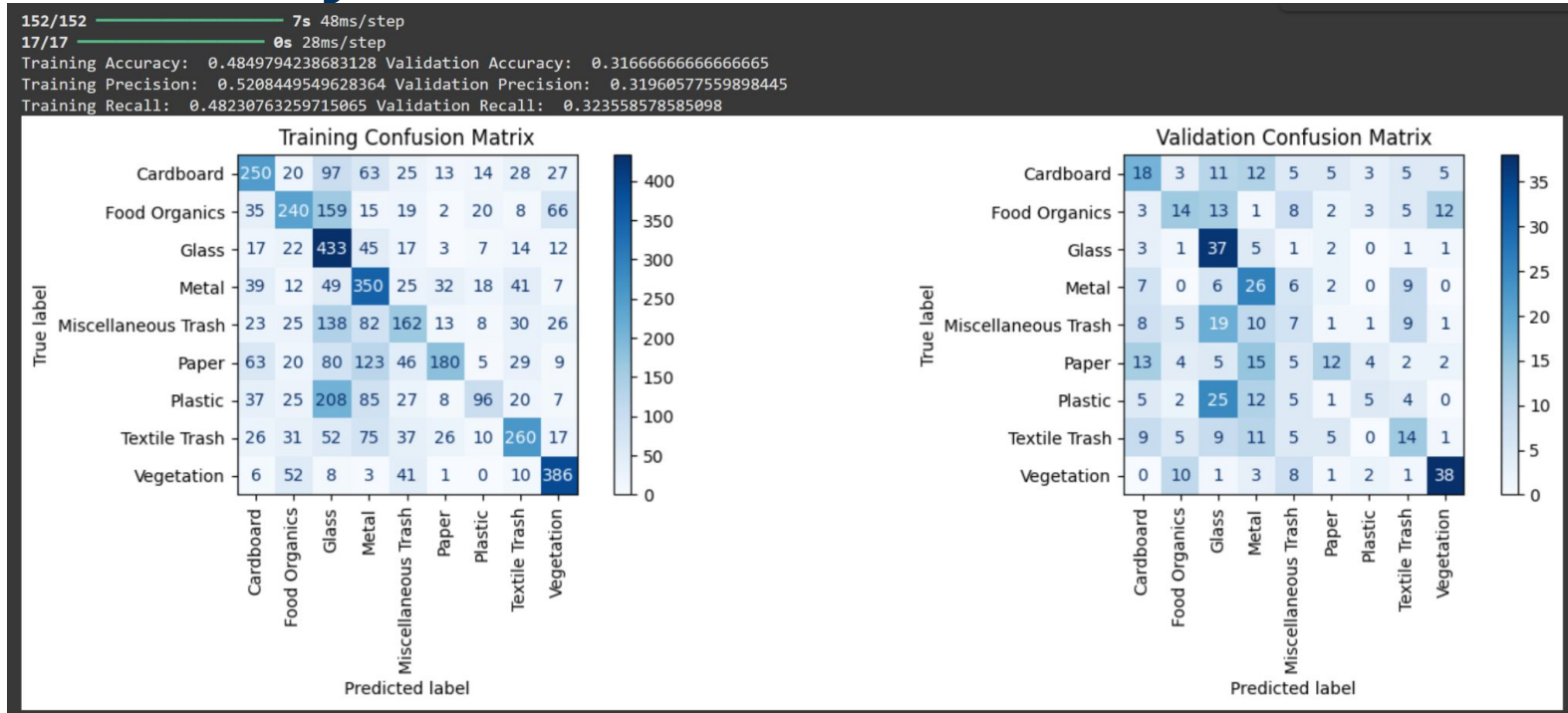
| Layer (type) | Output Shape | Param # |
|--------------------------------|--------------------|---------|
| conv2d_6 (Conv2D) | (None, 64, 64, 24) | 1,176 |
| max_pooling2d_6 (MaxPooling2D) | (None, 32, 32, 24) | 0 |
| dropout (Dropout) | (None, 32, 32, 24) | 0 |
| flatten_2 (Flatten) | (None, 24576) | 0 |
| dense_3 (Dense) | (None, 9) | 221,193 |

Total params: 222,369 (868.63 KB)

Trainable params: 222,369 (868.63 KB)

Non-trainable params: 0 (0.00 B)

Preliminary CNN Performance



- Overfitting observed on validation set
- Common confusions:

Paper ↔ Cardboard // Metal ↔ Glass // Plastic ↔ Misc Trash / Textile // Vegetation ↔ Cardboard // Plastic ↔ Glass

Limitation of Preliminary CNN Model

- **Shallow Architecture**

Limited number of convolutional layers restricted feature extraction capacity.

- **Low Input Resolution (64×64)**

Missed fine-grained visual details important for differentiating similar waste types.

- **No Regularization**

Lack of dropout or weight decay made the model prone to overfitting.

- **Static Learning Rate**

A fixed, high learning rate caused unstable convergence and poor generalization.

- **Underutilized Optimizer Settings**

Default optimizer configurations weren't well-tuned for this task.

Final CNN Model Setup

Model Architecture

- Add more CNN layers - ~50%
- VGG16 Inspired- Too big
- ResNet - ~65%

Tried multiple Image Resolutions

- $64 * 64$ - 65%
- $128 * 128$ - 70%
- $256 * 256$ - 70%

Learning Rate

- Static
 - 0.01 - Fast overfitting
 - 0.001 - 65%
 - 0.0001 - 70%
- Dynamic
 - 10x decrease in training rate - 75%

Optimizer

- Adam/AdamW/Nadam

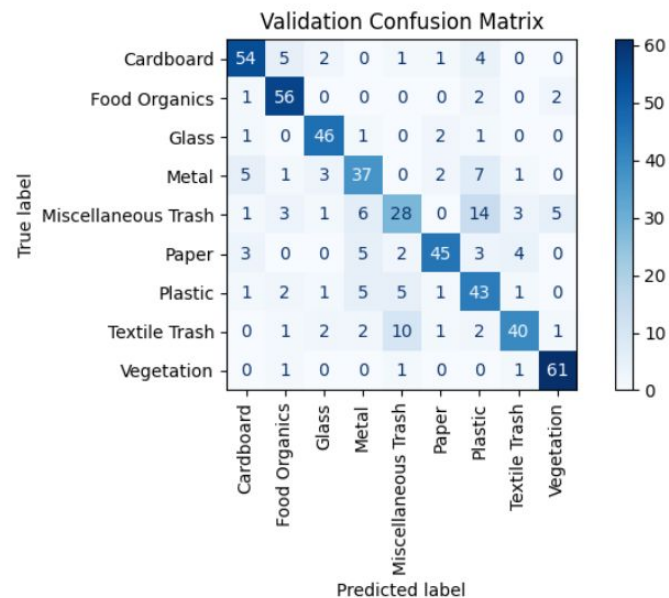
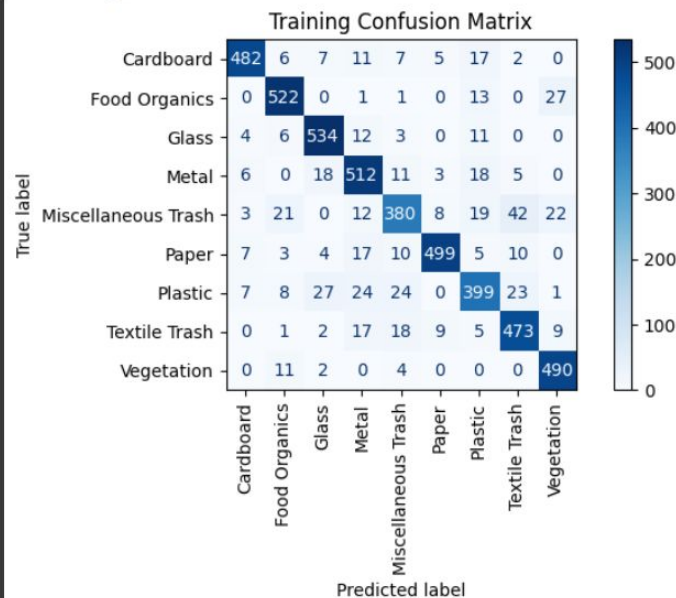
Final CNN Model Performance

Depth = 20

Training Accuracy: 0.8829218106995885 Validation Accuracy: 0.7592592592592593

Training Precision: 0.8822237515077148 Validation Precision: 0.7597589648991859

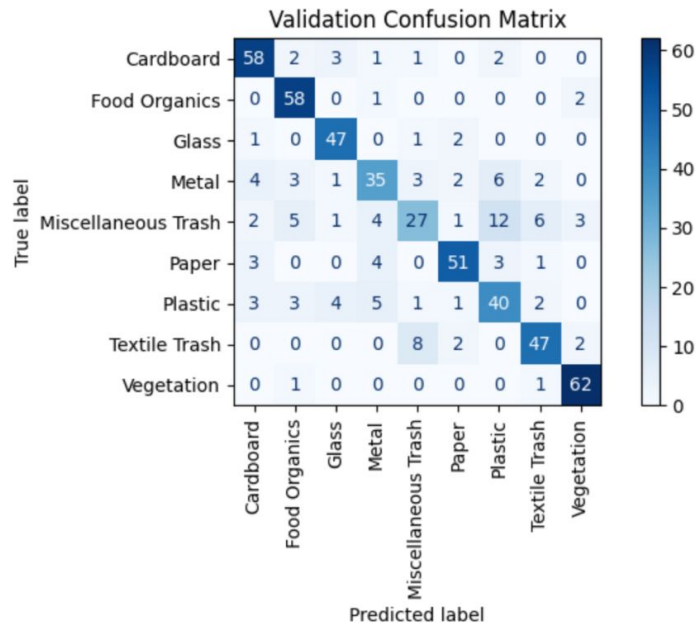
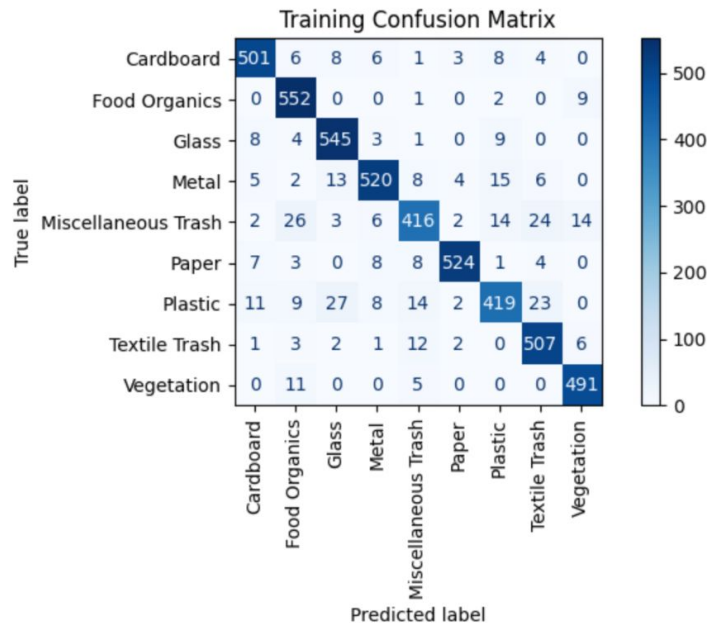
Training Recall: 0.8813463243847859 Validation Recall: 0.759045056915496



Final CNN Model Performance

Depth = 38

Training Accuracy: 0.9207818930041153 Validation Accuracy: 0.7870370370370371
Training Precision: 0.9207315068984274 Validation Precision: 0.7794186191520763
Training Recall: 0.9194033473380476 Validation Recall: 0.7857322009489529



Final CNN Model Performance

- High accuracy & clean diagonals in training → model learned the patterns well
- Slight performance drop in validation → due to inter-class visual similarity, not overfitting
- Prediction accuracy varies among different classes

High Accuracy: Cardboard, Food Organics, and Vegetarian

Low Accuracy: Miscellaneous Trash vs Plastic, Metal vs Plastic. These categories share visual or material similarity



Summary and Future Improvements

Overall, the model is

- Well-regularized
- Strong across all major classes
- Especially strong in organic vs inorganic separation (e.g., Food Organics vs Plastic)
- Ready for deployment or fine-tuning

For Future, we could improve the prediction even better through:

- More Class-Specific Augmentation: E.g., rotate or distort plastic and glass samples differently, since they may appear similar from certain angles
- Multi-view or metadata if available: If available, use additional context like texture or edge features