Waste Segregation Project

CNN Image Classification Model

by Thi Anh Tuyet Tu

Problem Statement

Improper waste disposal contributes to environmental degradation, increased landfill waste and inefficient recycling processes. Manual sorting is labour-intensive, error-prone and costly. An AI-powered waste classification system addresses these challenges by streamlining waste segregation, reducing operational costs and improving recycling rates.

Objective

The objective of this project is to implement an effective waste material segregation system using convolutional neural networks (CNNs) that categorises waste into distinct groups. This process enhances recycling efficiency, minimises environmental pollution, and promotes sustainable waste management practices.

The key goals are:

- Accurately classify waste materials into categories like cardboard, glass, paper, and plastic.
- Improve waste segregation efficiency to support recycling and reduce landfill waste.
- Understand the properties of different waste materials to optimise sorting methods for sustainability.

ML Approach

This project applies deep learning techniques to enhance recycling efficiency, reduce environmental pollution and promote sustainable waste management practices. In this project, we will:

- Train and fine-tune a CNN model to classify waste such as cardboard, glass, paper and plastic
- Evaluate model accuracy using performance metrics such as precision, recall and F1-score
- Gain insights into the role of AI in sustainable environmental solutions

PART 1:

DATA ANALYSIS AND PREPROCESSING

Data Analysis

Examining dataset properties, visualizing images

Preprocessing

Preparing data for model training

Dataset Splitting

Creating training and validation sets

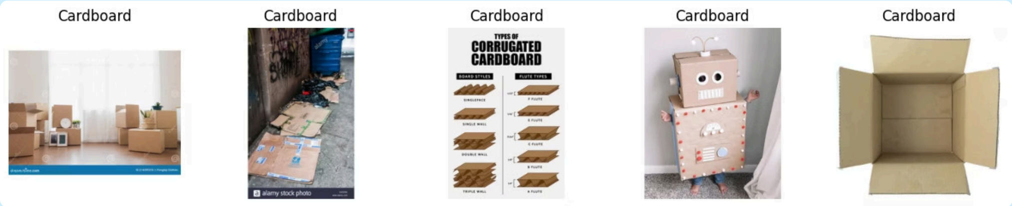
Raw Data Overview

Dataset Structure

The dataset consisted of a compressed zip file containing waste classification images organized into 7 distinct categories. After extraction, the data was structured as individual folders representing different waste types, with each folder containing corresponding images for that classification.

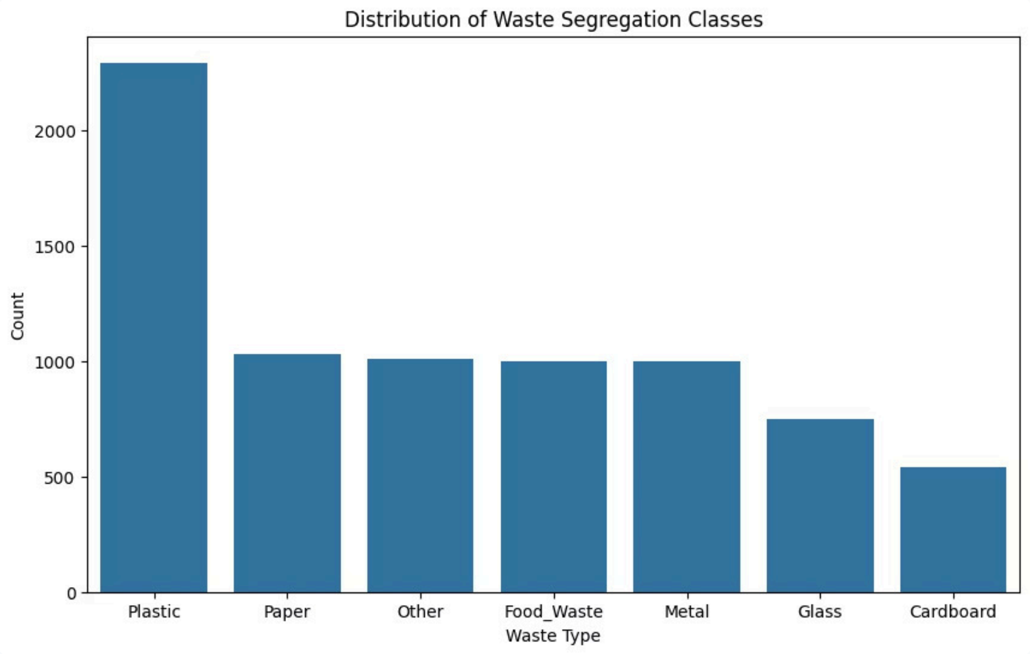
Organization Method

Images were organized in a hierarchical folder structure, with each waste category having its own dedicated directory.



Dataset Composition

Te dataset contains 7,625 total images distributed across 7 waste categories with significant class imbalance:



Label	Category	Count	Percentage
0	Plastic	2,295	30.1%
1	Paper	1,030	13.5%
2	Other	1,010	13.2%
3	Food_Waste	1,000	13.1%
4	Metal	1,000	13.1%
5	Glass	750	9.8%
6	Cardboard	540	7.1%

Image Properties

Format

PNG files

Original Dimensions

256×256 pixels

Channels

RGB (3-channel color images)

Pixel Value Range

0-255 (standard 8-bit)

Data Preprocessing Pipeline

Image Resizing

Maintained square aspect ratio and resized to 224×224 pixels, which is optimal for CNN architectures

Color Space Verification

Ensured all images are in RGB format

Label Encoding

Applied one-hot encoding using pandas get_dummies for multi-class classification

Pixel Normalization

Scaled pixel values to [0,1] range and converted to float32 for model compatibility

Train-Validation Split

70-30 split with stratification to preserve class distribution

Final Dataset Structure

5,337	**2,288**	**224x224**	**7**
Training Set	**Validation Set**	**Image Shape**	**Label Shape**
70% of total images	30% of total images	(224, 224, 3)	(7,) - one-hot encoded

PART 2:

MODEL ARCHITECTURE AND TRAINING ANALYSIS

Architecture Design

Creating neural network structure

Model Training

Optimizing parameters and weights

3

Model Evaluation

Creating training and validation sets

Initial Model Exploration (1-3 Layers)

Architecture Design:

- Convolutional Layers: 1-3 layers with (3×3) kernels
- Filter Progression: 32 → 64 → 128
- Pooling: MaxPooling (2×2) after each conv layer
- Regularization: Dropout (0.25 after conv, 0.5 after dense)
- Global Pooling: Global Average Pooling instead of Flatten to reduce parameters
- Output: Softmax activation for 7-class classification
- Optimizer: Adam with categorical crossentropy loss

Initial Model Performance Analysis

CONV LAYERS	TRAINING TIME (MINUTES)	PARAMETERS	FINAL VAL ACCURACY	FINAL VAL LOSS	BEST VAL ACCURACY	BEST VAL ACC EPOCH	FINAL TRAIN ACCURACY	FINAL TRAIN LOSS
1	2.90	6,023	32.78%	1.7085	32.91%	9	33.18% ✅	1.7373
2	**6.32**	**28,615**	**39.03%**	**1.6042**	**39.03%**	10	35.92% ✅	**1.6738**
3	**7.80**	**110,663**	**43.23%** 🏆	**1.5199**	**43.23%** 🏆	10	40.19% ✅	**1.6309**

Performance Improvement | Computational Cost | Best Performer | Healthy Pattern | Early Convergence

- **Performance Scaling Confirmed:** Clear positive correlation between depth and accuracy. Validation accuracy increases from 32.78% (0 layers) to 43.23% (2 layers) - a significant 10.45% improvemen
- **Training Efficiency Excellent:** Loss consistently decreases as layers are added, indicating better model learning. Training time and parameters grow substantially with each layer (37.8 mins and 1.1M params for 2 layers). 3-layer model achieves best performance with reasonable computational cost (2.7x time for 32% gain).
- **Healthy Learning Pattern:** Training and validation accuracies closely aligned,indicating minimal overfitting detected across all architectures
- **Early Convergence Signal:** All models reached peak accuracy at final epoch, indicating strong potential for improvement with more training epochs
- **Architecture Scaling Potential:** Consistent improvement pattern suggests deeper architectures (4-6 layers) will likely yield better results
- **Foundation Established:** Solid baseline performance provides clear direction for second model optimization

Extended Architecture Testing (4-6 Layers)

Deeper Architecture

- Test 4-6 convolutional layers to capitalize on scaling benefits

Filter Cap

Maximum 512 filters to prevent overfitting

Extended Training

Double the epochs (20+ epochs) since all models showed potential for continued learning

Based on foundation results, the second model will target with this configuration.

Expected Outcomes: Significant performance improvement while maintaining healthy training dynamics observed in foundation model.

- **Performance Target:** Aim for >50% accuracy based on observed scaling trajectory
- **Maintain Efficiency:** Monitor computational cost vs. performance trade-offs

Extended Model Performance Analysis

Conv Layers	Training Time (Minutes)	Parameters	Final Val Accuracy	Final Val Loss	Best Val Accuracy	Best Val Acc Epoch	Final Train Accuracy	Final Train Loss
1	3.68	6,023	34.53%	1.6890	34.75%	19	*33.69%*	1.7080
2	**6.28**	**28,615**	**38.11%**	**1.5962**	**41.65%**	19	*39.40%*	**1.6210**
3	**7.77**	**110,663**	**41.48%**	**1.5557**	**42.57%**	**13**	*39.91%*	**1.6063**
4	7.74	110,663	**43.49%**	**1.4838**	**44.67%**	**15**	*42.44%*	**1.5710**
5	**7.82**	110,663	**41.43%**	**1.4949**	**44.10%**	**17**	*40.73%*	**1.5974**
6	**7.89**	110,663	**44.01%**	**1.4874**	44.01%	**20**	*40.30%*	**1.6037**

Positive Trend Negative Trend Best Performer Underfitting Warning

- **Optimal Architecture Confirmed:** 4 convolutional layers achieves peak validation accuracy (44.67%)
- **Significant Performance Gain:** 29.4% relative improvement from 1-layer (34.53%) to 4-layer (44.67%)
- **Diminishing Returns Beyond 4 Layers:** Performance plateaus/declines at 5-6 layers despite same parameter count
- **Unusual Pattern:** Train accuracy < validation accuracy suggests underfitting rather than overfitting
- **Excellent Training Efficiency:** Only 2.1x time increase (3.68→7.89 mins) for 18x parameter growth
- **Strategic Direction:** Use 4-layer architecture as a base model, and further optimize this model

Final Model Optimization

Class Weight Implementation

Addressing class imbalance, help to improving accuracy with better prediction for minority classes

Callback Integration

Monitoring and optimizing training

Hyperparameter Tuning

Fine-tuning model parameters

Performance Evaluation

Measuring model improvements

Class Weight Implementation

Due to significant class imbalance, class weights were incorporated:

LABEL	COUNT	WEIGHT
Plastic	2,295	0.474738
Paper	1,030	1.057460
Other	1,010	1.078400
Food Waste	1,000	1.089184
Metal	1,000	1.089184
Glass	750	1.452245
Cardboard	540	2.017007

- **Class weights delivered a meaningful 2.3% improvement in validation accuracy (43.5% → 45.8%),** indicating better generalization and more balanced predictions across minority classes. However, this performance gain came at the cost of significant training instability, with highly volatile learning curves and erratic convergence patterns - most notably a dramatic drop to 19.3% validation accuracy at epoch 6. Training time also increased by 40% per step due to the weighted loss computations.
- While the baseline model without class weights showed smooth, consistent improvement, the class-weighted version exhibited unpredictable fluctuations throughout training. Despite these challenges, the final 45.8% validation accuracy represents the best performance achieved and suggests improved minority class recognition.
- The recommendation is to retain class weights for the proven performance benefit, but implement more aggressive callbacks (shorter patience for early stopping and learning rate reduction) to mitigate the training instability and ensure more reliable convergence to the improved performance baseline.

Callback Integration

Monitoring Metric Selection

All callbacks monitor val_accuracy as the primary metric since this directly reflects the model's generalization capability on unseen data, which is more reliable than training accuracy for preventing overfitting.

EarlyStopping Configuration

Set with patience=7 epochs to allow sufficient time for the model to recover from temporary performance dips while preventing unnecessary training once peak performance is reached. The restore_best_weights=True ensures the final model uses the best-performing weights rather than potentially inferior final epoch weights.

ModelCheckpoint Strategy

Configured to save only the best model based on validation accuracy, providing a safety net to preserve optimal performance even if training continues beyond the peak.

Learning Rate Reduction

Implemented with factor=0.7 to provide moderate learning rate reduction (30% decrease) when validation accuracy plateaus. This conservative factor allows gradual fine-tuning without overly aggressive learning rate drops that could halt learning prematurely. The patience=4 setting is shorter than EarlyStopping to enable learning rate adjustments before complete training termination, while min_lr=1e-6 prevents the learning rate from becoming too small to be effective.

Final Model

4

Convolutional Layers

Optimal depth for feature extraction

456,007

Total Parameters

Model complexity

30

Maximum Epochs

With early stopping

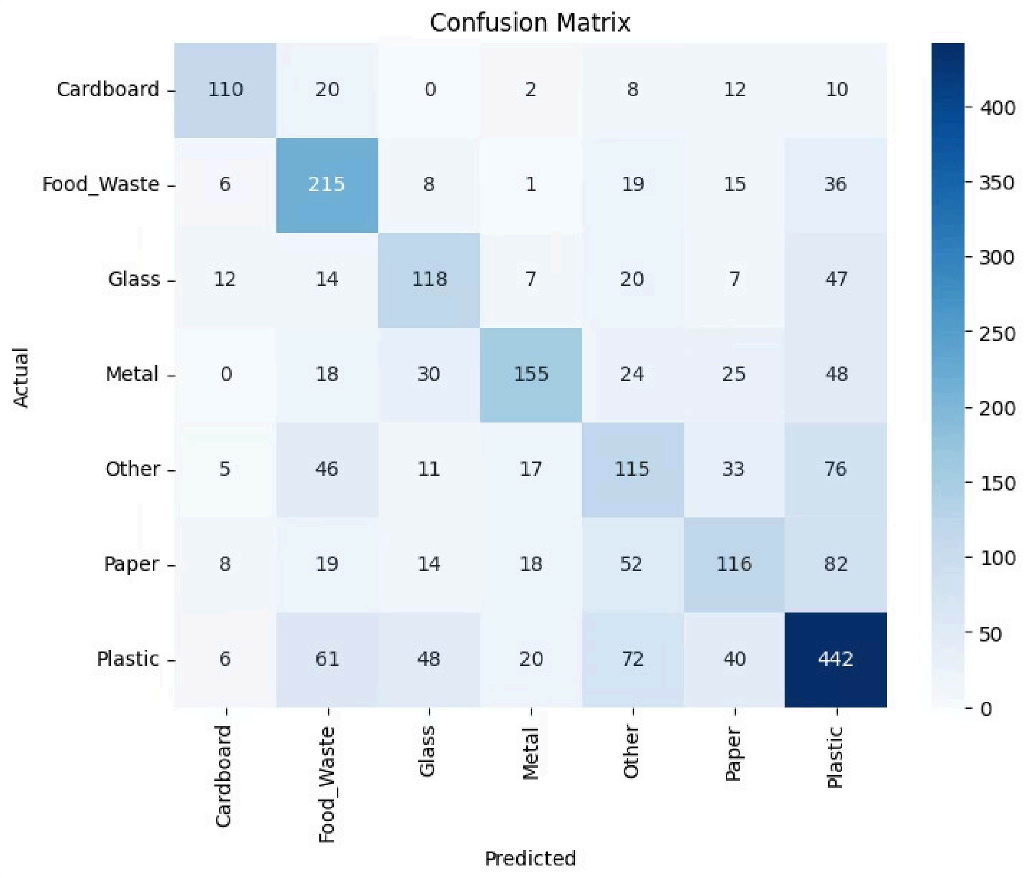
Final Model Performance

Evaluation Metrics:

- Accuracy: 55.5%
- Precision: 56.0%
- Recall: 55.5%
- F1 Score: 55.3%

Confusion Matrix Analysis:

The model shows strong performance for well-represented classes (Plastic, Paper) but struggles with minority classes (Glass, Cardboard). The confusion matrix reveals common misclassifications between similar waste types, particularly between Paper and Cardboard categories.



Key Conclusions

1

Optimal Architecture

4-layer CNN provides the best balance of performance and efficiency

Class Imbalance Impact

Class weights improve minority class recognition but introduce training instability

Regularization Necessity

Callbacks are essential for managing overfitting and training volatility

Performance Limitation

55.5% accuracy indicates room for improvement through data augmentation or transfer learning approaches