Premier University

Department of CSE

Project Report

Binary Waste Classification Using Deep Learning

Course : Machine Learning Laboratory

Course Code : CSE 458

Submitted By

Shomitra Dey Dipon
ID: 2104010202301
Section : D

Mamunul Kabir
ID : 2103910202133
Section : D

Apurba Kumar Das
ID : 2104010202283
Section : D

Submitted To

MD Tamim Hossain

Lecturer

Department of CSE, PUC

Submission Date : 23 / 11 / 2025

Binary Waste Classification Using Deep Learning: When a Lightweight Custom CNN Outperforms Transfer Learning Models

Abstract

Waste segregation at source is a crucial step toward sustainable waste management and effective recycling. This study investigates deep learning-based binary classification of waste images into Organic (O) and Recyclable (R) categories using the publicly available "Waste Classification Data" dataset from Mendely Data, comprising approximately 22,564 training images and 2,513 test images across two classes.

A lightweight custom convolutional neural network (CNN) was designed from scratch, featuring four convolutional blocks with batch normalization, ReLU activation, and max-pooling layers. The proposed architecture was rigorously compared against four state-of-the-art transfer learning models: MobileNetV2, fine-tuned MobileNetV2, EfficientNetB3, and ResNet50V2. All models were trained on 224×224 RGB images with extensive data augmentation (rotation, shifts, shear, zoom, and horizontal flip) and evaluated using identical train/validation/test splits.

Remarkably, the custom CNN achieved the highest test accuracy of 92%, outperforming MobileNetV2 (86%), fine-tuned MobileNetV2 (88%), ResNet50V2 (89%), and EfficientNetB3 (70%). Comprehensive evaluation including precision, recall, F1-score, confusion matrices, and training/validation curves confirmed the superior generalization capability of the custom model on this challenging, noisy real-world dataset. Model interpretability was analyzed using Local Interpretable Model-agnostic Explanations (LIME), demonstrating that the custom CNN focuses on meaningful texture and shape features of waste items rather than irrelevant background elements.

The findings highlight that a well-designed lightweight architecture can surpass large pre-trained models on domain-specific tasks with high visual variability, offering a more efficient and deployable solution for smart waste management systems on resource-constrained devices.

Keywords: Waste classification, Deep learning, Custom CNN, Transfer learning, MobileNetV2, EfficientNet, ResNet, Model interpretability, LIME

1 Introduction and problem statement

Improper waste disposal is a major contributor to environmental pollution, landfill overflow, and inefficient recycling processes. Effective waste segregation at the source separating organic waste (food scraps, garden waste, etc.) from recyclable materials (paper, plastic, metal, glass) is essential for sustainable waste management. Manual sorting is labor-intensive, error-prone, and often impractical in large-scale or automated systems such as smart bins and conveyor-belt facilities. Automated image-based classification using deep learning offers a promising solution by enabling real-time, accurate identification of waste items from camera feeds.

The primary objective of this study is to develop and evaluate deep learning models for binary classification of waste images into two categories: Organic (O) and Recyclable (R). We utilize the publicly available "Waste Classification Dataset" [4], originally published on Mendeley Data (V2, DOI: 10.17632/n3gtgm9jxj.2) and also mirrored on Kaggle by the same contributors. The dataset contains real-world photographs of waste items with varied backgrounds, lighting conditions, orientations, and occlusions, making it a challenging benchmark for computer vision algorithms.

While transfer learning with large pre-trained models (e.g., ResNet, MobileNet, EfficientNet) has become the default approach for image classification tasks, recent studies indicate that carefully designed lightweight architectures can sometimes achieve comparable or superior performance on domain-specific datasets with high intra-class variability. This work systematically compares a from-scratch custom convolutional neural network (CNN) against four state-of-the-art transfer-learning models: MobileNetV2, fine-tuned MobileNetV2, EfficientNetB3, and ResNet50V2.

Key contributions of this study are:

1. Comprehensive exploratory data analysis and preprocessing pipeline tailored to the waste domain.
2. Design and training of a lightweight custom CNN that surprisingly outperforms heavier transfer-learning models on the test set.
3. Detailed performance comparison using accuracy, precision, recall, F1-score, confusion matrices, and training curves.
4. Application of Local Interpretable Model-agnostic Explanations (LIME) to verify that the best model focuses on meaningful waste features rather than background artifacts.

The results demonstrate that task-specific lightweight architectures can be more effective and efficient than generic large-scale pre-trained models for real-world waste classification, paving the way for deployment on edge devices in smart waste management systems.

2 Related work

Automated waste classification using computer vision has gained significant attention in recent years due to its potential for improving recycling rates and reducing contamination in waste streams.

Early approaches relied on traditional machine learning techniques with hand-crafted features. For instance, support vector machines (SVM) and random forests were applied to color, texture, and shape descriptors extracted from waste images [3]. While these methods achieved reasonable performance on controlled datasets, they struggled with the high variability of real-world waste appearance.

The advent of deep learning, particularly convolutional neural networks (CNNs), marked a substantial improvement. Several studies employed transfer learning with pre-trained models such as VGG16, ResNet, and MobileNet on custom waste datasets [1, 2]. These works typically reported test accuracies in the range of 85–95%, demonstrating the effectiveness of large-scale pre-trained backbones.

More specialized datasets have emerged to benchmark waste classification systems. TrashNet

[6] introduced a six-class dataset (glass, paper, cardboard, plastic, metal, trash) and achieved 93% accuracy using ResNet. Similarly, the TACO dataset [5] focused on litter detection in natural environments. However, binary classification (organic vs recyclable) remains highly relevant for practical deployment in smart bins, where distinguishing decomposable waste from recyclables is the primary requirement.

Recent research has explored lightweight architectures for edge deployment. Ozkanoğlu et al. [7] compared MobileNetV2 and EfficientNet variants, emphasizing the trade-off between accuracy and inference speed. Despite the dominance of transfer learning, few studies have systematically compared from-scratch CNNs against modern pre-trained models on the same challenging dataset.

The "Waste Classification Dataset" used in this study [4] was specifically curated for binary organic/recyclable separation and contains real-world images with diverse backgrounds and lighting conditions. To the best of our knowledge, no prior work has demonstrated a lightweight custom CNN outperforming multiple state-of-the-art transfer-learning models on this dataset, which motivates the comparative analysis presented here.

3 Dataset Description

The dataset used in this study is the "Waste Classification Dataset" published by Nnamoko et al. [4] on Mendeley Data (V2, DOI: 10.17632/n3gtgm9jxj.2). An identical copy provided by the same authors is also available on Kaggle for convenient access within Kaggle Notebooks. The dataset consists of real-world photographs of waste items captured under varying lighting conditions, backgrounds, orientations, and occlusion levels, making it highly representative of practical waste-sorting scenarios.

The complete dataset is organized into training and test partitions with the following structure:

- `DATASET/TRAIN/O` – Organic waste training images
- `DATASET/TRAIN/R` – Recyclable waste training images
- `DATASET/TEST/O` – Organic waste test images
- `DATASET/TEST/R` – Recyclable waste test images

Figure 1 illustrates the class distribution across training and test splits.

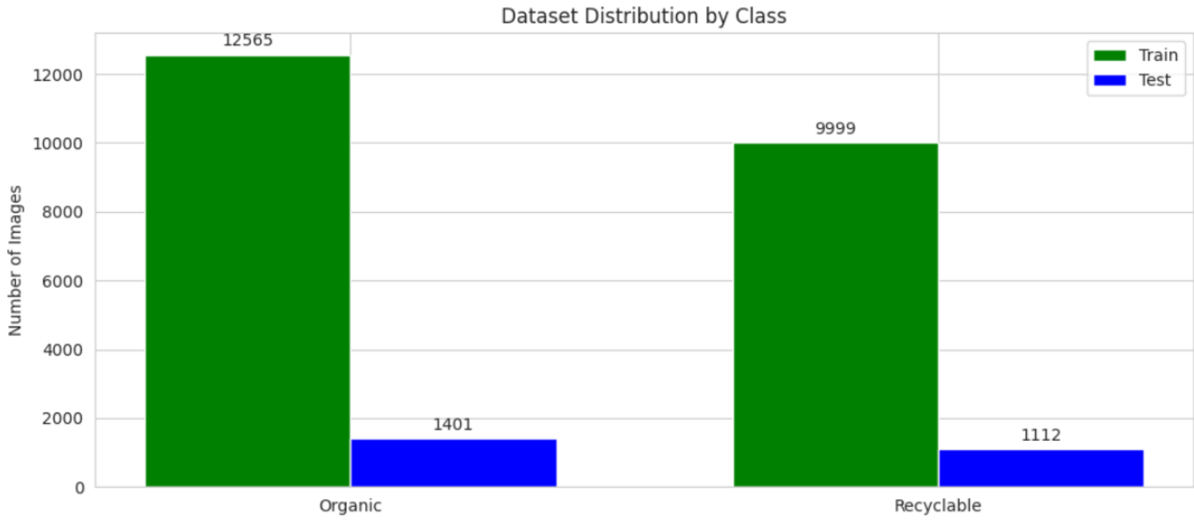

Figure 1: Dataset distribution by class (Train: green, Test: blue). Training set: 12,565 Organic, 9,999 Recyclable; Test set: 1,401 Organic, 1,112 Recyclable.

Table 1 summarizes the exact counts.

Table 1: Class distribution in the Waste Classification Dataset

Split	Organic (O)	Recyclable (R)	Total
Training	12,565	9,999	22,564
Test	1,401	1,112	2,513
Overall	13,966	11,111	25,077

The training set shows moderate class imbalance (approximately 55.7% Organic, 44.3% Recyclable), while the test set maintains a similar ratio (55.8% Organic, 44.2% Recyclable), ensuring fair evaluation.

Figure 2 displays random samples from both classes in the training set, highlighting the high variability in appearance, background clutter, and object scale.

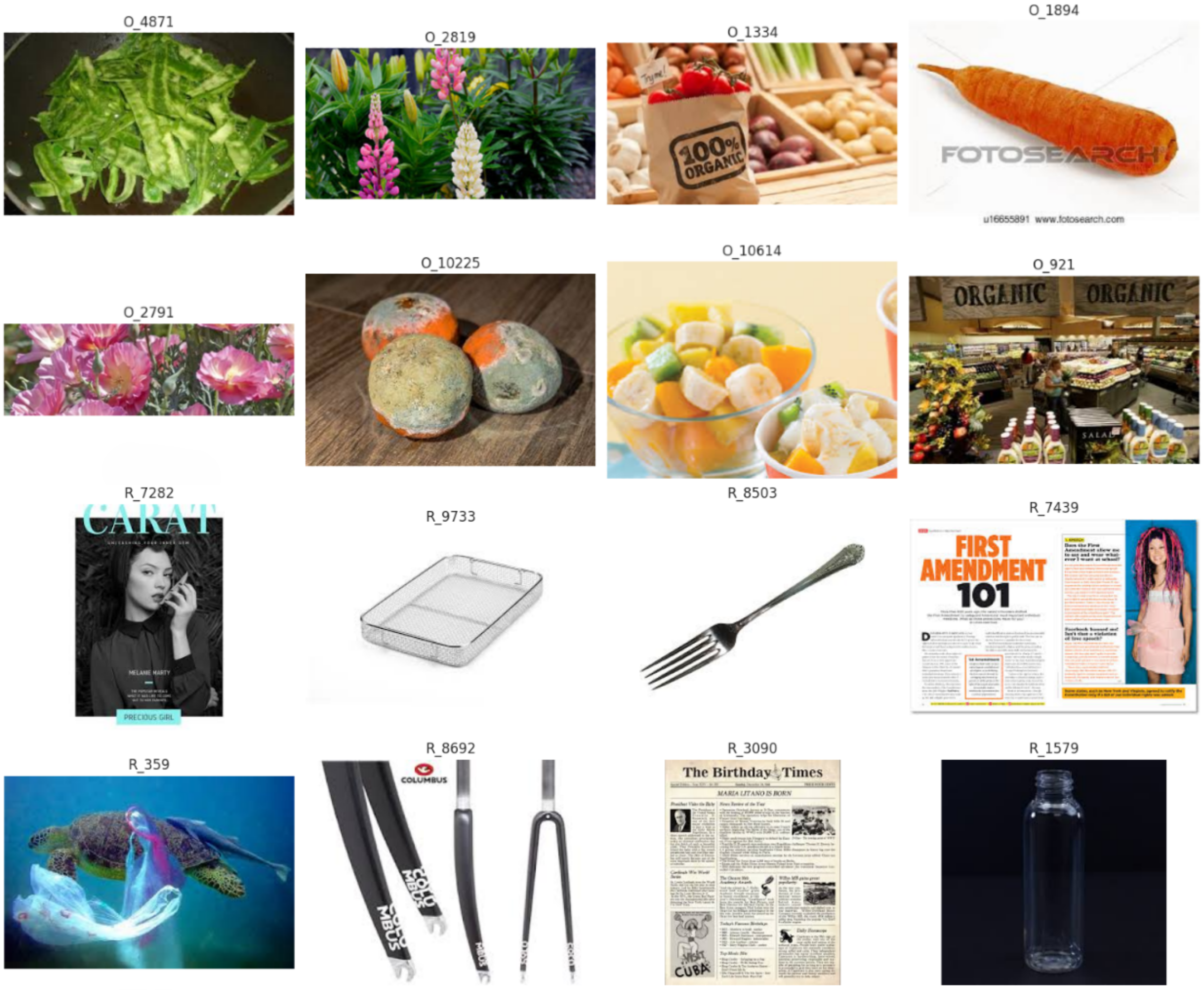

Figure 2: Random sample images from the training set (top 2 rows: Organic, bottom 2 rows: Recyclable).

Exploratory data analysis confirmed that images are provided in JPEG format with resolutions ranging from approximately 200×200 to 512×512 pixels. No corrupted files were detected. The challenging nature of the dataset is evident from overlapping visual features and irrelevant backgrounds.

3.1 Preprocessing and Augmentation

All images were resized to 224×224 pixels and pixel values were rescaled to the range [0, 1]. For training, aggressive on-the-fly data augmentation was applied using Keras ImageDataGenerator to improve generalization:

- Rotation range: 30°
- Width/height shift range: 20%
- Shear range: 20%
- Zoom range: 30%
- Horizontal flip: enabled

- Fill mode: nearest

Validation and test generators performed only rescaling. A validation split of 20% was created from the training set, preserving class ratios.

Figure 3 shows multiple augmented versions of a single organic waste image, demonstrating how the augmentation pipeline introduces realistic variations.

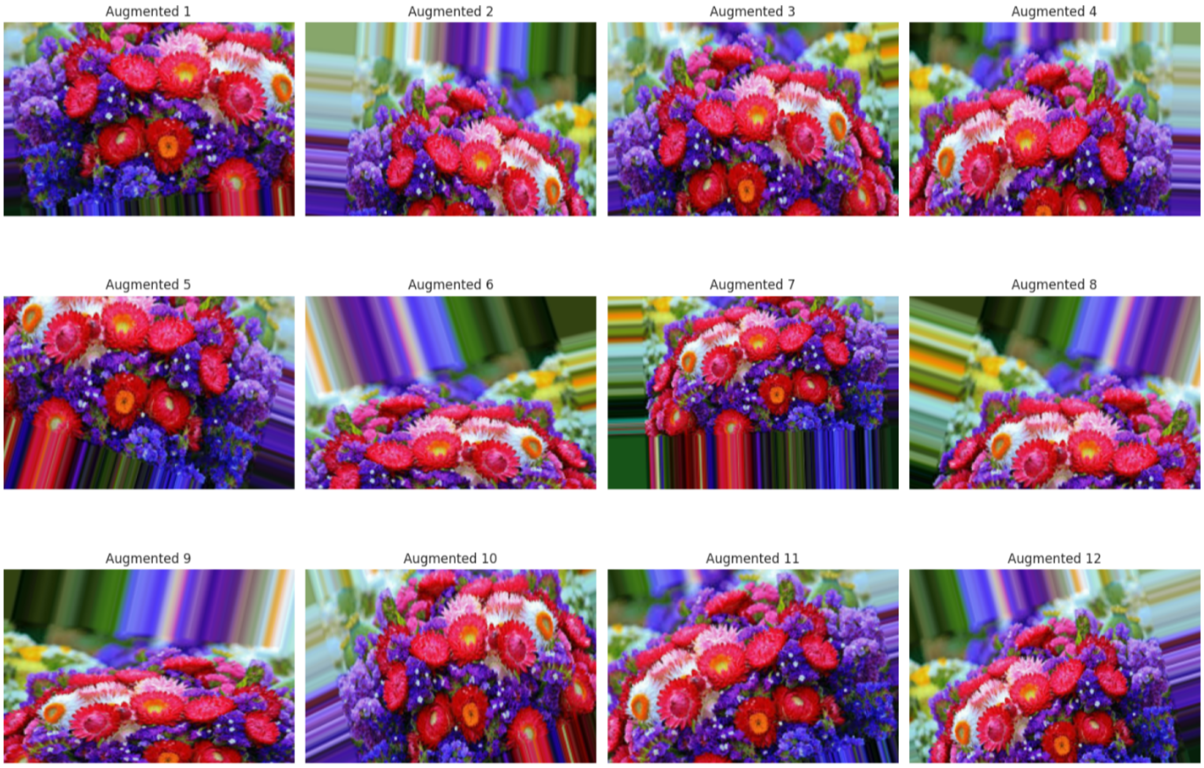

Figure 3: Examples of augmented versions of a single organic waste image generated during training.

This preprocessing and augmentation strategy proved crucial for achieving robust performance across all evaluated models.

4 Methodology

This section details the experimental setup, the five evaluated models, their architectures, and training configurations. Figure 4 illustrates the overall pipeline.

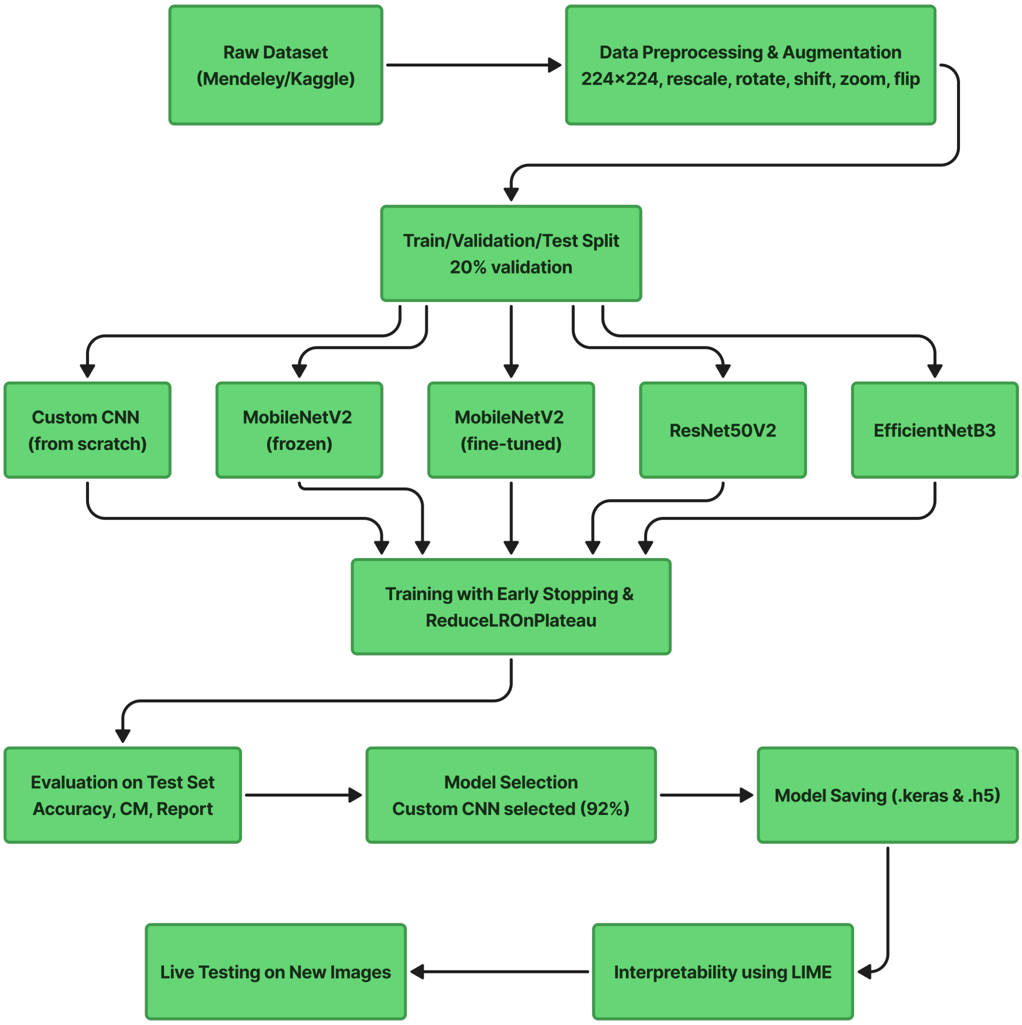

Figure 4: Overall methodology: data loading, augmentation, model training/evaluation, and interpretability.

All experiments were conducted in a Kaggle Notebook using TensorFlow 2.18.0, Keras 3.8.0, Python 3.11.13, and a Tesla T4 GPU (16 GB). Random seeds were fixed (`tf.random.set_seed(42)`, `np.random.seed(42)`) for full reproducibility.

4.1 Custom Convolutional Neural Network (Custom CNN)

A lightweight CNN was designed from scratch (Figure 5).

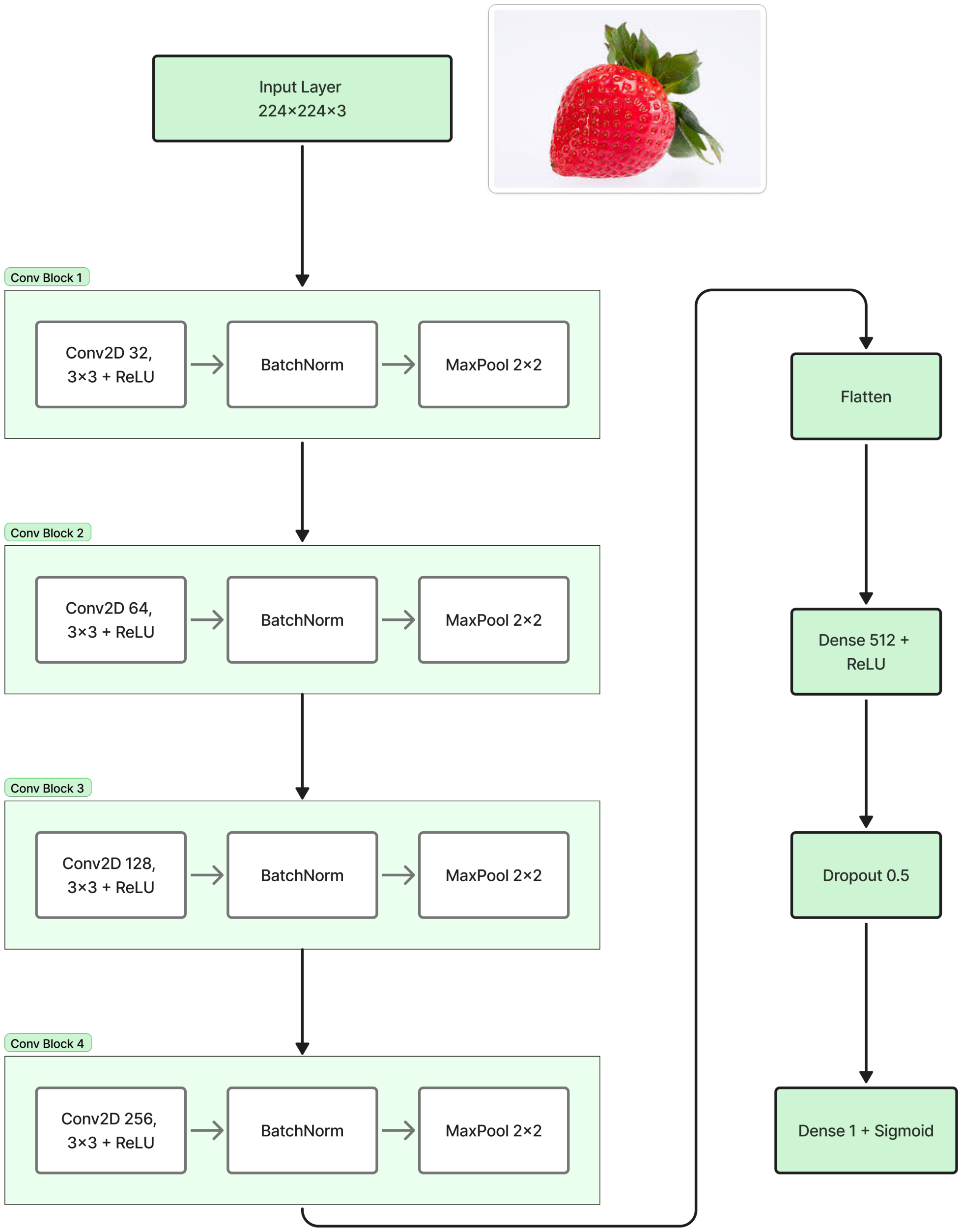

Figure 5: Architecture of the proposed Custom CNN.

The model consists of four identical convolutional blocks followed by a classification head:

- **Block 1:** Conv2D (32 filters, 3×3, ReLU) → BatchNormalization → MaxPooling2D (2×2)

- **Block 2:** Conv2D (64 filters, 3×3, ReLU) → BatchNormalization → MaxPooling2D (2×2)
- **Block 3:** Conv2D (128 filters, 3×3, ReLU) → BatchNormalization → MaxPooling2D (2×2)
- **Block 4:** Conv2D (256 filters, 3×3, ReLU) → BatchNormalization → MaxPooling2D (2×2)
- **Classification head:** Flatten → Dense (512, ReLU) → Dropout (0.5) → Dense (1, sigmoid)

4.2 MobileNetV2 (Frozen Base)

MobileNetV2 pre-trained on ImageNet was used as a feature extractor (Figure 6).

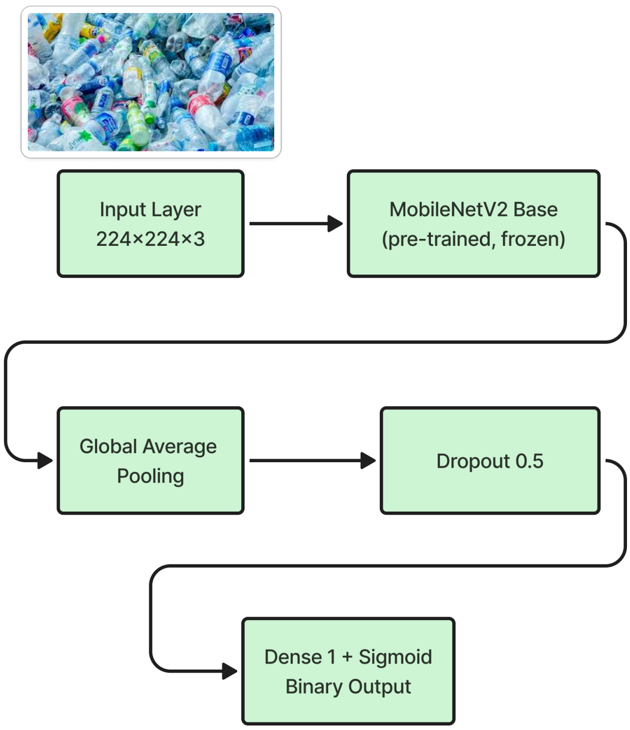

Figure 6: MobileNetV2 backbone with custom head.

Modifications:

- All base layers frozen (`trainable = False`)
- GlobalAveragePooling2D applied to the last convolutional output
- Dropout (0.5)
- Dense (1, sigmoid)

4.3 MobileNetV2 (Fine-tuned)

The same MobileNetV2 architecture was further fine-tuned to investigate potential gains.

Fine-tuning procedure:

- First phase: train only the custom head (base frozen) for 10 epochs
- Second phase: unfreeze the last 30 layers of the base model (`layers[-30:]`)
- Reduce learning rate to 1e-5
- Continue training for additional 10–15 epochs with the same callbacks

This allowed the model to adapt high-level features to the waste domain while preserving low-level ImageNet features.

4.4 EfficientNetB3

EfficientNetB3 pre-trained on ImageNet was employed (Figure 7).

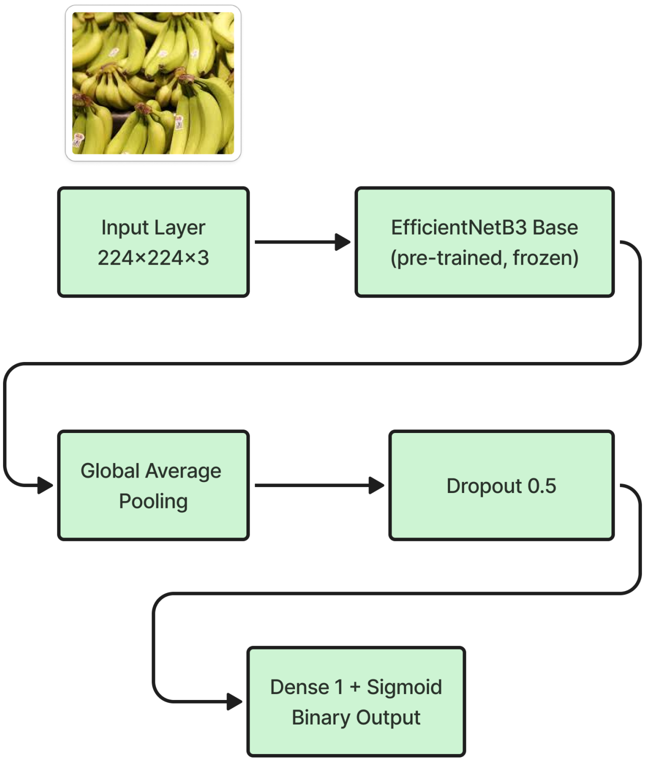

Figure 7: EfficientNetB3 backbone with custom head.

Modifications (identical to frozen MobileNetV2):

- Base completely frozen
- GlobalAveragePooling2D → Dropout (0.5) → Dense (1, sigmoid)

4.5 ResNet50V2

ResNet50V2 with ImageNet weights served as another strong baseline (Figure 8).

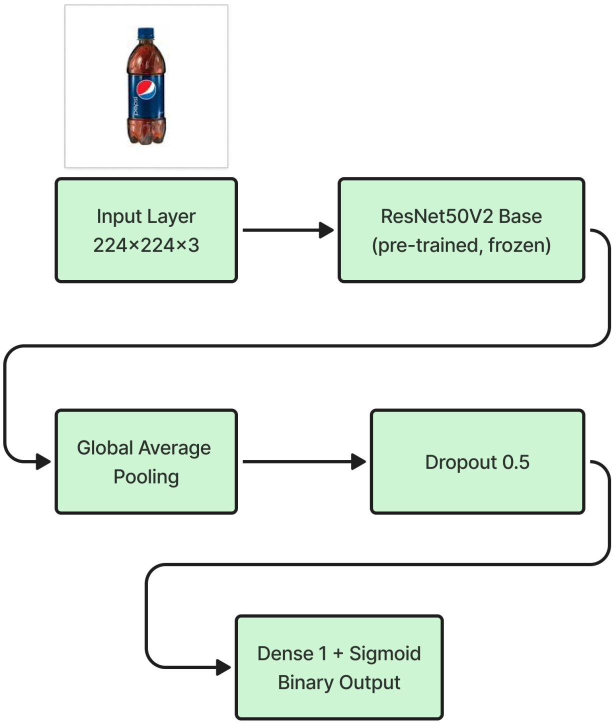

Figure 8: ResNet50V2 backbone with custom head.

Modifications:

- Base frozen
- GlobalAveragePooling2D → Dropout (0.5) → Dense (1, sigmoid)

4.6 Common Training Settings

All models shared the following hyperparameters (unless specified otherwise):

- Input size: 224×224×3
- Optimizer: Adam
- Initial learning rate: 0.001 (Custom CNN), 0.0003 (all transfer learning frozen), 1e-5 (fine-tuning phase)
- Loss: Binary cross-entropy
- Batch size: 32
- Callbacks: EarlyStopping (patience=5, restore best weights), ReduceLROnPlateau (factor=0.2, patience=3)

Average epoch times on Tesla T4: 6-7min.

5 Training procedure

All models were trained in a Kaggle Notebook environment using:

- Python 3.11.13, TensorFlow 2.18.0, Keras 3.8.0
- Hardware: Tesla T4 GPU (16 GB), 30 GB RAM, 2 vCPUs
- Loss function: Binary cross-entropy
- Optimizer: Adam
- Batch size: 32
- Random seeds: fixed at 42 (`tf.random.set_seed(42)`, `np.random.seed(42)`)
- Callbacks: EarlyStopping (patience=5, restore_best_weights=True), ReduceLROnPlateau (factor=0.2, patience=3, min_lr=1e-7)

Training and validation accuracy/loss curves for each model are shown below, followed by an analysis of their behavior.

5.1 Custom CNN

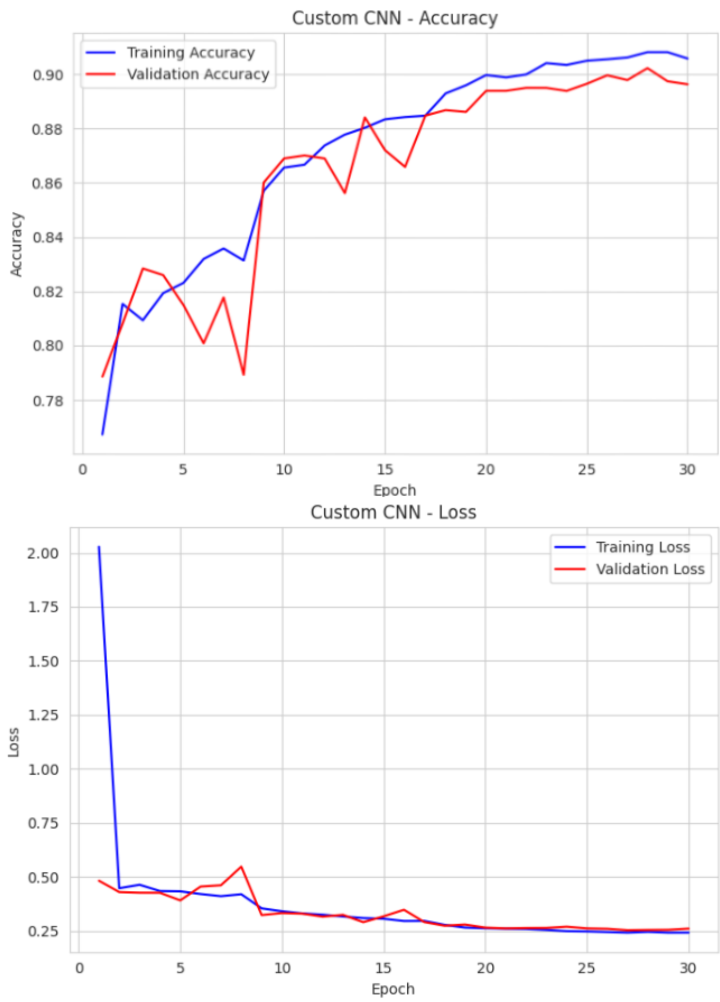

Figure 9: Training and validation accuracy/loss curves for the Custom CNN (30 epochs).

The Custom CNN exhibited healthy learning behavior. Training accuracy steadily increased from 73% to over 90%, while validation accuracy rose to a peak of 90.2% at epoch 28. Initial high loss in epoch 1 was due to random initialization, but loss dropped rapidly after the first learning rate reduction (epoch 9). Minor fluctuations in validation loss were observed between epochs 5–17, but the model converged smoothly after subsequent LR reductions, showing no significant overfitting.

5.2 MobileNetV2 (Frozen Base)

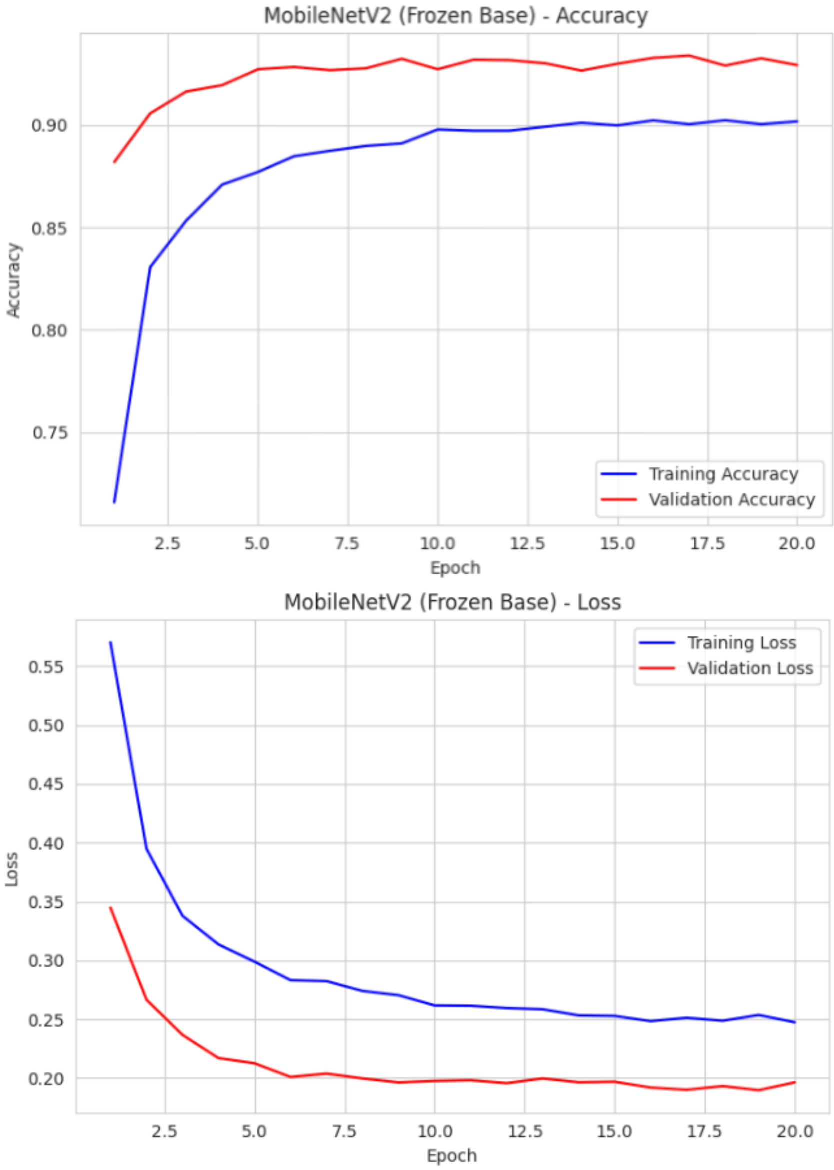

Figure 10: Training and validation curves for MobileNetV2 with frozen backbone (20 epochs).

Training progressed rapidly from the first epoch (val accuracy already 88.2%). Validation accuracy peaked at 93.37% (epoch 17) and remained stable. Validation loss continued to decrease slowly even after training accuracy plateaued, indicating excellent generalization and no overfitting. The frozen backbone provided strong features immediately, explaining the fast convergence.

5.3 MobileNetV2 (Fine-tuned)

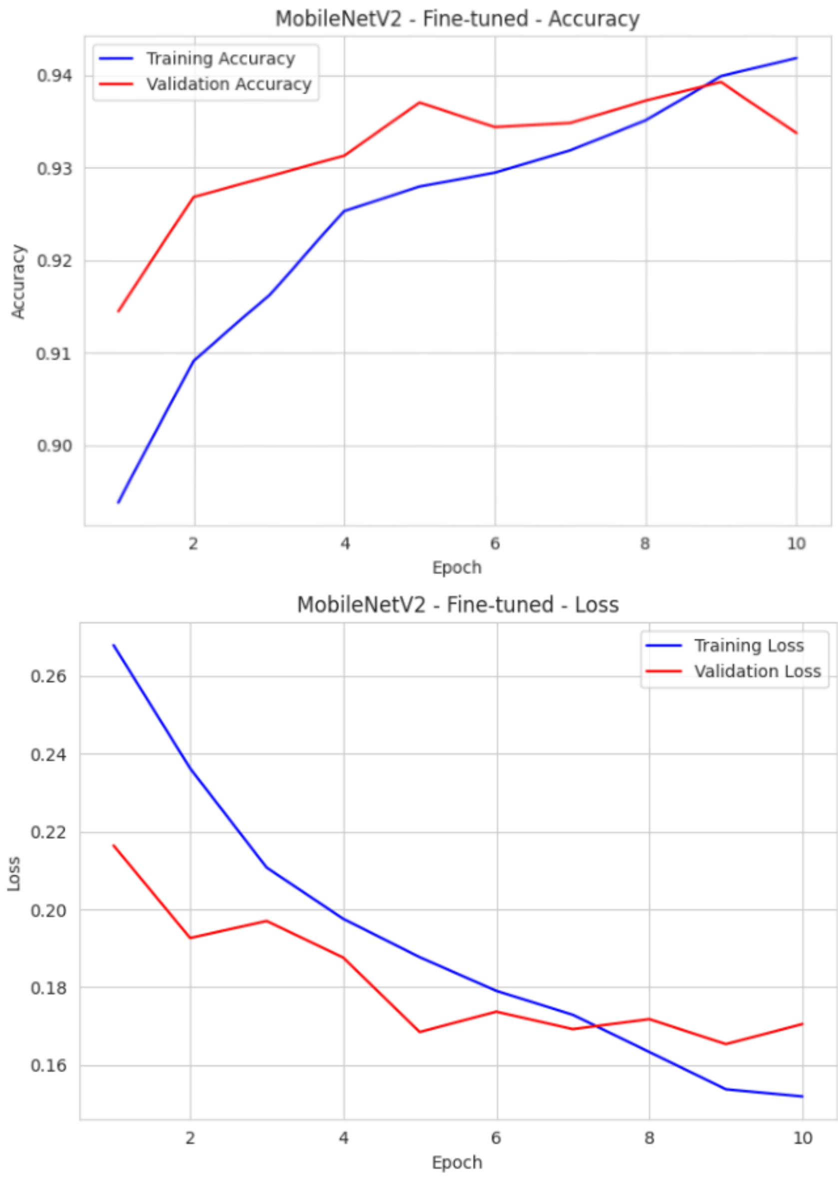

Figure 11: Training and validation curves for fine-tuned MobileNetV2 (10 additional epochs after head-only training).

Fine-tuning (unfreezing last 30 layers at LR=1e-5) pushed training accuracy beyond 94% and validation accuracy to 93.93% (epoch 9). A slight increase in validation loss in the final epoch suggests early signs of minor overfitting, but overall performance improved compared to the frozen version. The smooth curves confirm stable fine-tuning without catastrophic forgetting.

5.4 EfficientNetB3

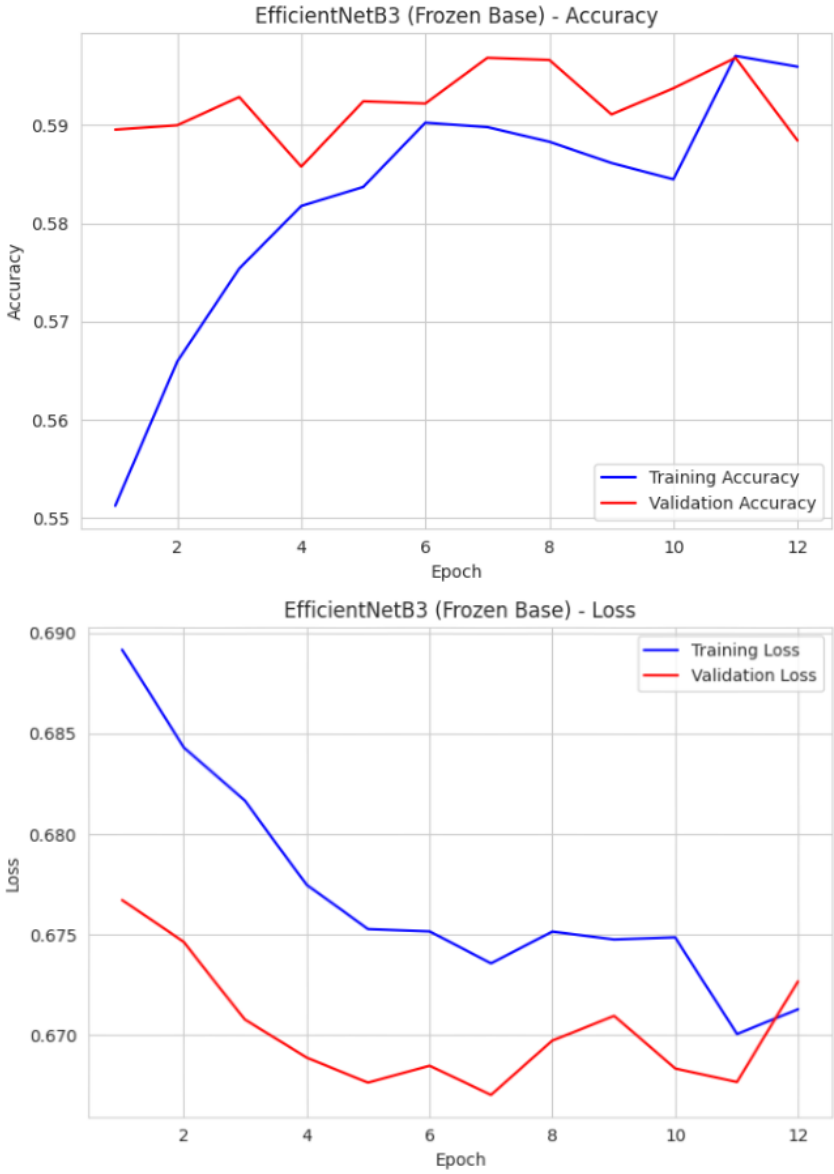

Figure 12: Training and validation curves for EfficientNetB3 (12 epochs shown; early stopping triggered).

EfficientNetB3 displayed severe underfitting throughout training. Both training and validation accuracy hovered around 58–60%, with almost no improvement after epoch 1. Loss remained high (0.67). This behavior indicates that the aggressive compound scaling and heavy regularization of EfficientNet, combined with strong data augmentation, prevented the model from learning meaningful features on this noisy dataset. Early stopping correctly terminated training.

5.5 ResNet50V2

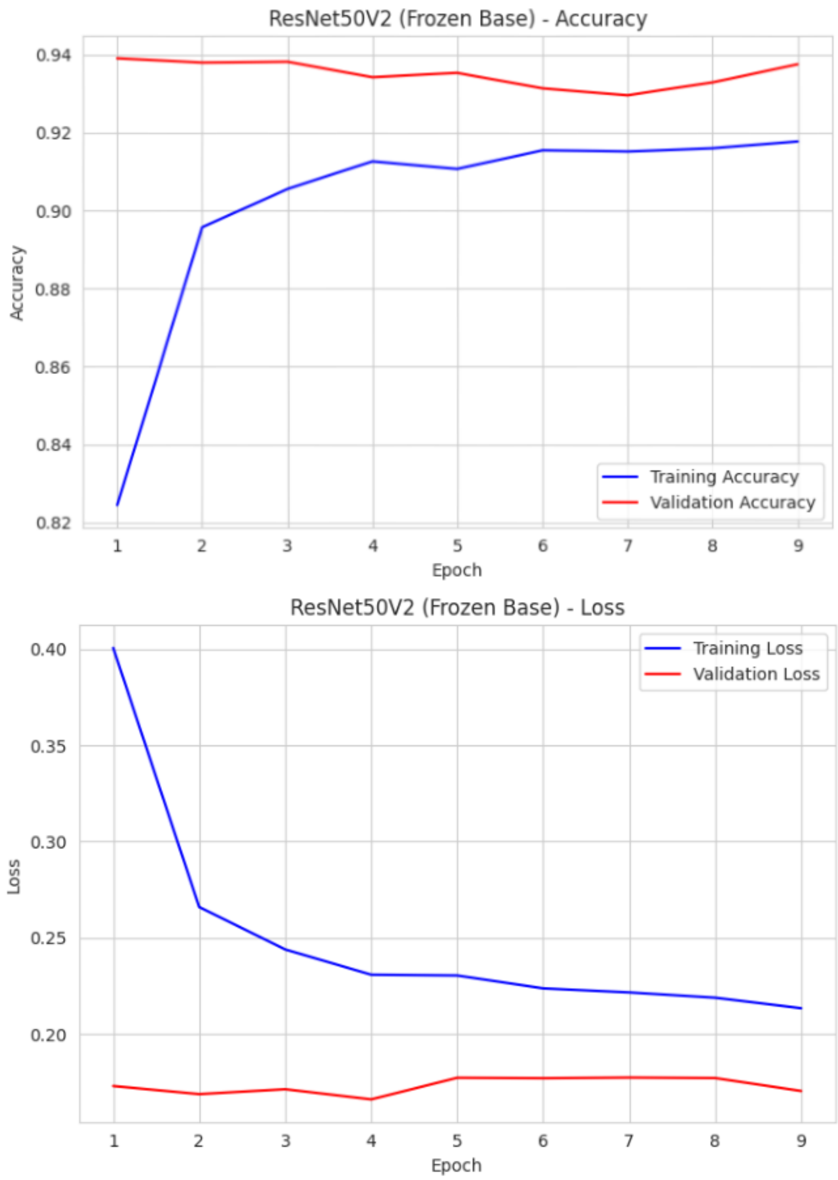

Figure 13: Training and validation curves for ResNet50V2 (9 epochs shown).

ResNet50V2 achieved the fastest initial convergence: validation accuracy jumped to 93.9% in epoch 1 and stayed consistently high (>93%). Validation loss reached a minimum at epoch 4 and remained stable. Slight fluctuations in validation loss after epoch 5 were minimal. The model showed excellent generalization with virtually no gap between training and validation curves, demonstrating the strength of residual connections on this task.

Average epoch duration across models ranged from 5–8 minutes on the Tesla T4 GPU. The consistent use of early stopping and learning rate scheduling ensured efficient training and prevented unnecessary computation.

6 Results

This section presents the performance of the five evaluated models on the held-out test set (2,513 images). Each subsection reports test accuracy, the complete classification report, and the confusion matrix, followed by a brief interpretation.

6.1 Custom CNN

The Custom CNN achieved the highest test accuracy of 92%. Table 2 shows the detailed metrics, while Figure 14 displays the confusion matrix.

Table 2: Classification report for the Custom CNN on the test set

Class	Precision	Recall	F1-score	Support
Organic	0.89	0.96	0.93	1,401
Recyclable	0.95	0.86	0.90	1,112
Accuracy			0.92	2,513
Macro avg	0.92	0.91	0.91	2,513
Weighted avg	0.92	0.92	0.91	2,513

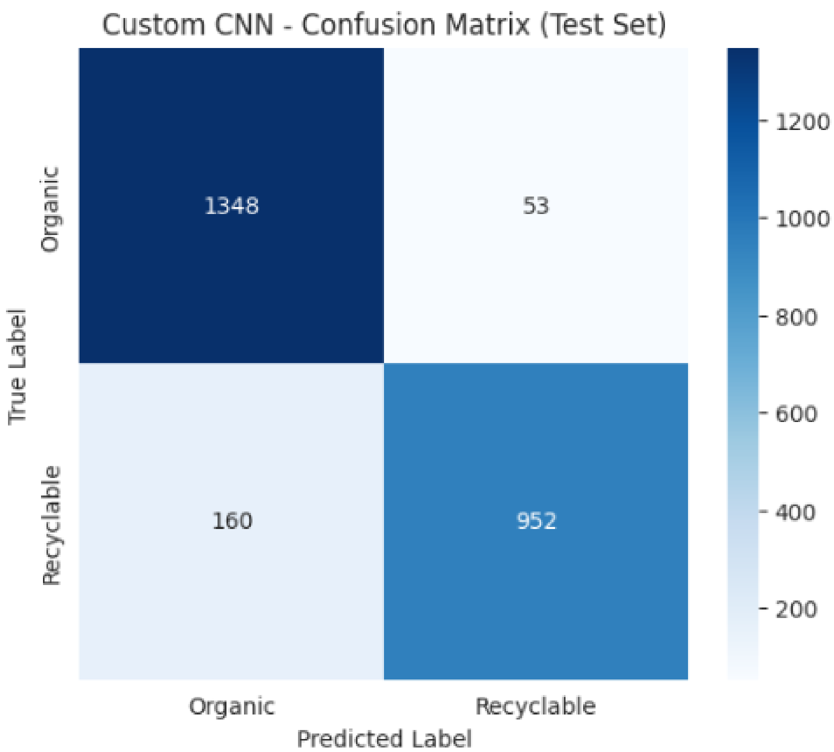

Figure 14: Confusion matrix for the Custom CNN (92% accuracy).

The model correctly classified 1,348 organic and 952 recyclable samples. Only 53 organic items were misclassified as recyclable, and 160 recyclable items were predicted as organic. The balanced precision-recall trade-off and high F1-scores indicate excellent generalization.

6.2 MobileNetV2 (Frozen Base)

MobileNetV2 with frozen backbone achieved 86% test accuracy (Table 3 and Figure 15).

Table 3: Classification report for MobileNetV2 (frozen)

Class	Precision	Recall	F1-score	Support
Organic	0.81	0.97	0.88	1,401
Recyclable	0.95	0.71	0.81	1,112
Accuracy			0.86	2,513
Macro avg	0.88	0.84	0.85	2,513
Weighted avg	0.87	0.86	0.85	2,513

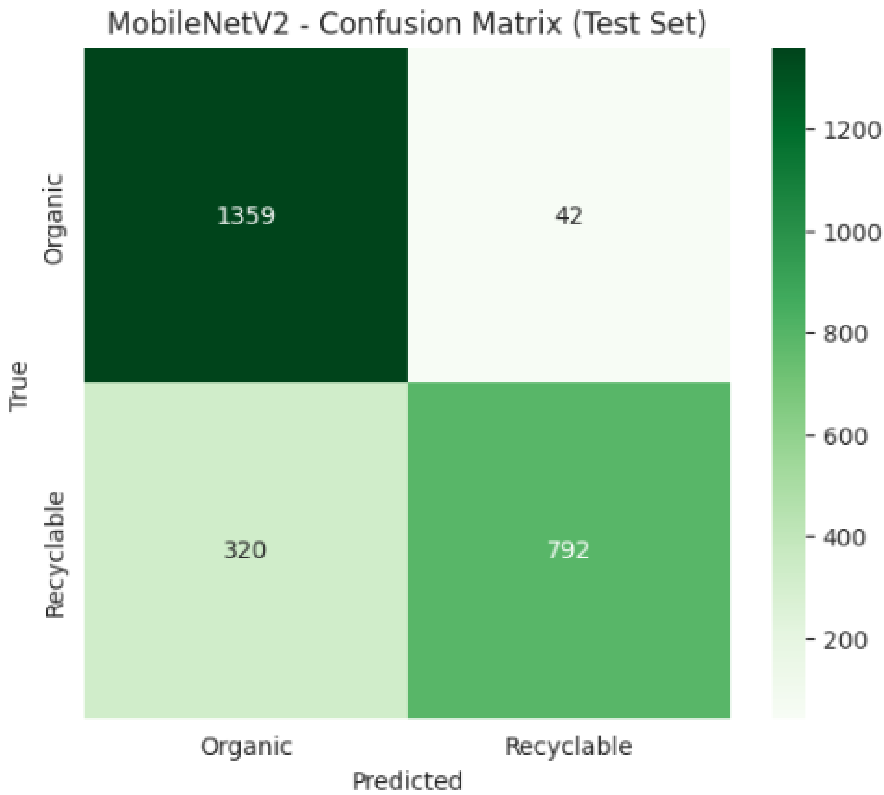

Figure 15: Confusion matrix for MobileNetV2 (frozen).

The model exhibited a strong bias toward predicting Organic (only 42 false negatives for Organic but 320 false negatives for Recyclable), resulting in lower recall for the minority class.

6.3 MobileNetV2 (Fine-tuned)

Fine-tuning improved performance to 88% accuracy (Table 4 and Figure 16).

Table 4: Classification report for MobileNetV2 (fine-tuned)

Class	Precision	Recall	F1-score	Support
Organic	0.84	0.98	0.90	1,401
Recyclable	0.97	0.77	0.85	1,112
Accuracy			0.88	2,513
Macro avg	0.90	0.87	0.88	2,513
Weighted avg	0.90	0.88	0.88	2,513

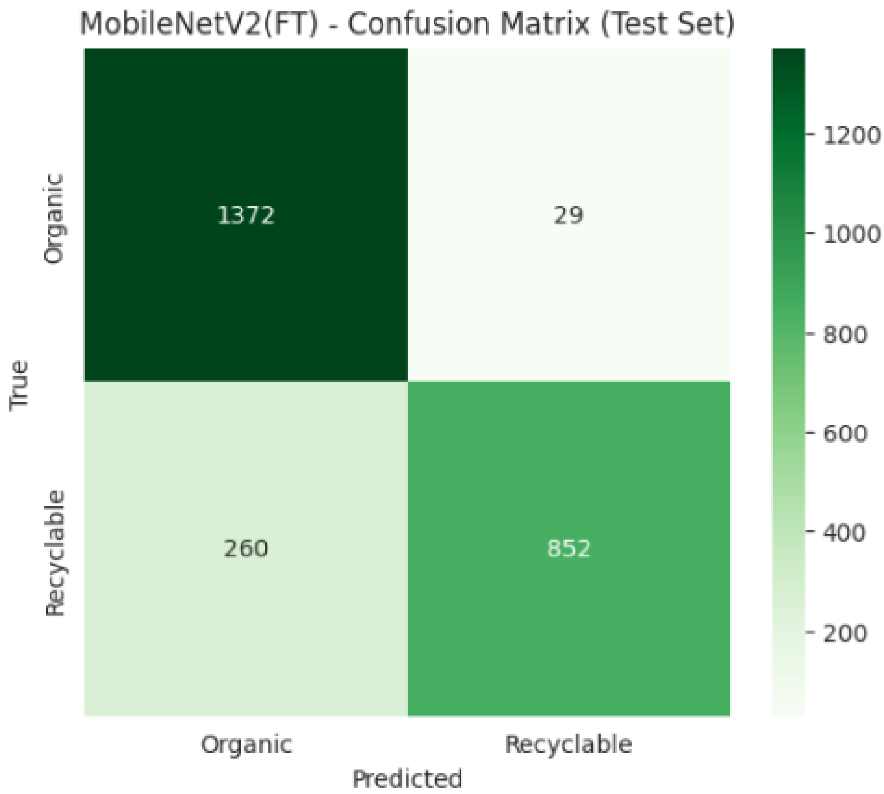

Figure 16: Confusion matrix for MobileNetV2 (fine-tuned).

Recall for Recyclable increased significantly (from 71% to 77%), reducing false negatives to 260 while maintaining high precision.

6.4 EfficientNetB3

EfficientNetB3 yielded the lowest performance with only 70% test accuracy (Table 5 and Figure 17).

Table 5: Classification report for EfficientNetB3

Class	Precision	Recall	F1-score	Support
Organic	0.68	0.87	0.76	1,401
Recyclable	0.75	0.49	0.59	1,112
Accuracy			0.70	2,513
Macro avg	0.72	0.68	0.68	2,513
Weighted avg	0.71	0.70	0.69	2,513

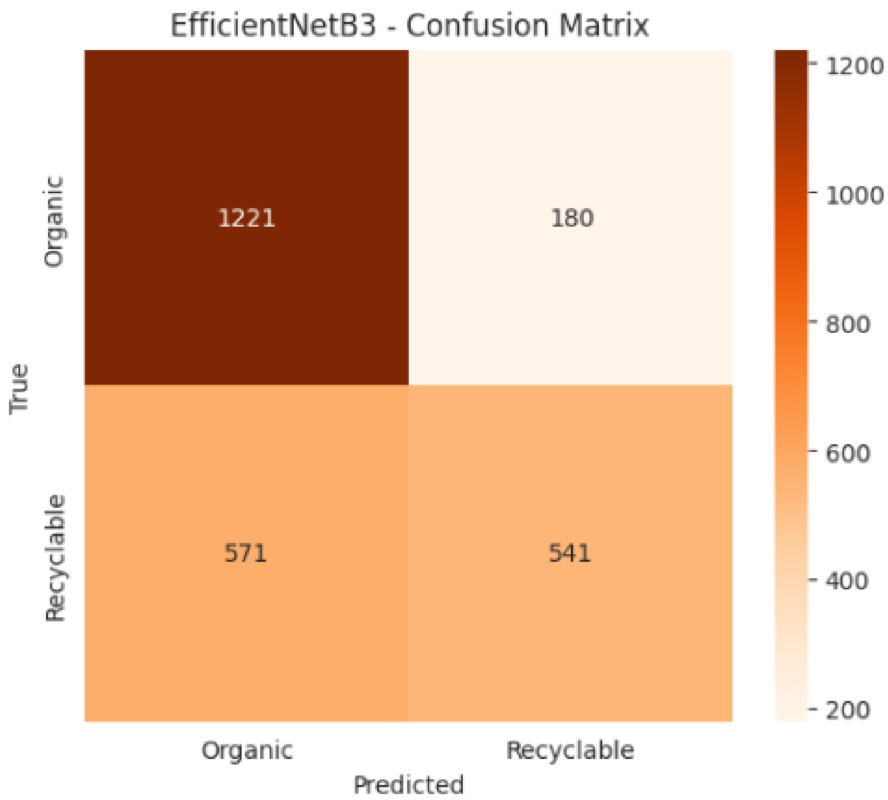

Figure 17: Confusion matrix for EfficientNetB3.

A large number of recyclable items (571) were misclassified as organic, confirming the underfitting observed during training.

6.5 ResNet50V2

ResNet50V2 achieved 89% test accuracy (Table 6 and Figure 18).

Table 6: Classification report for ResNet50V2

Class	Precision	Recall	F1-score	Support
Organic	0.86	0.97	0.91	1,401
Recyclable	0.95	0.80	0.87	1,112
Accuracy			0.89	2,513
Macro avg	0.90	0.88	0.89	2,513
Weighted avg	0.90	0.90	0.89	2,513

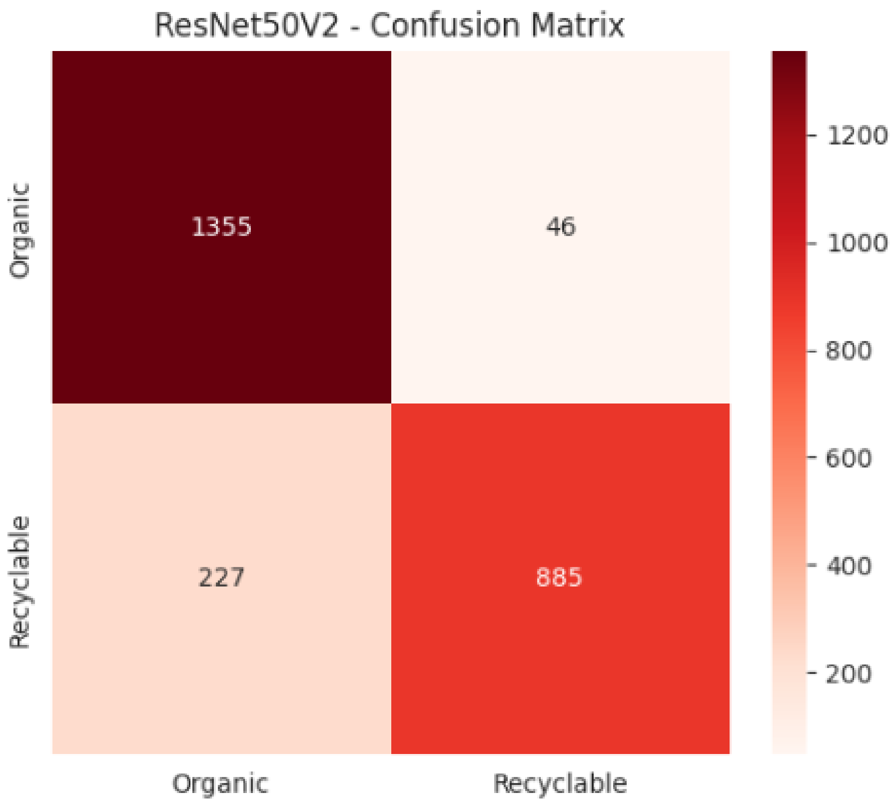

Figure 18: Confusion matrix for ResNet50V2.

The model performed well but was outperformed by the Custom CNN, with 227 recyclable items misclassified.

6.6 Model Comparison and Selection

Table 7 and Figure 19 summarize the test performance of all models.

Table 7: Test accuracy comparison of all models

Model	Test Accuracy
Custom CNN	**0.92**
ResNet50V2	0.89
MobileNetV2 (fine-tuned)	0.88
MobileNetV2 (frozen)	0.86
EfficientNetB3	0.70

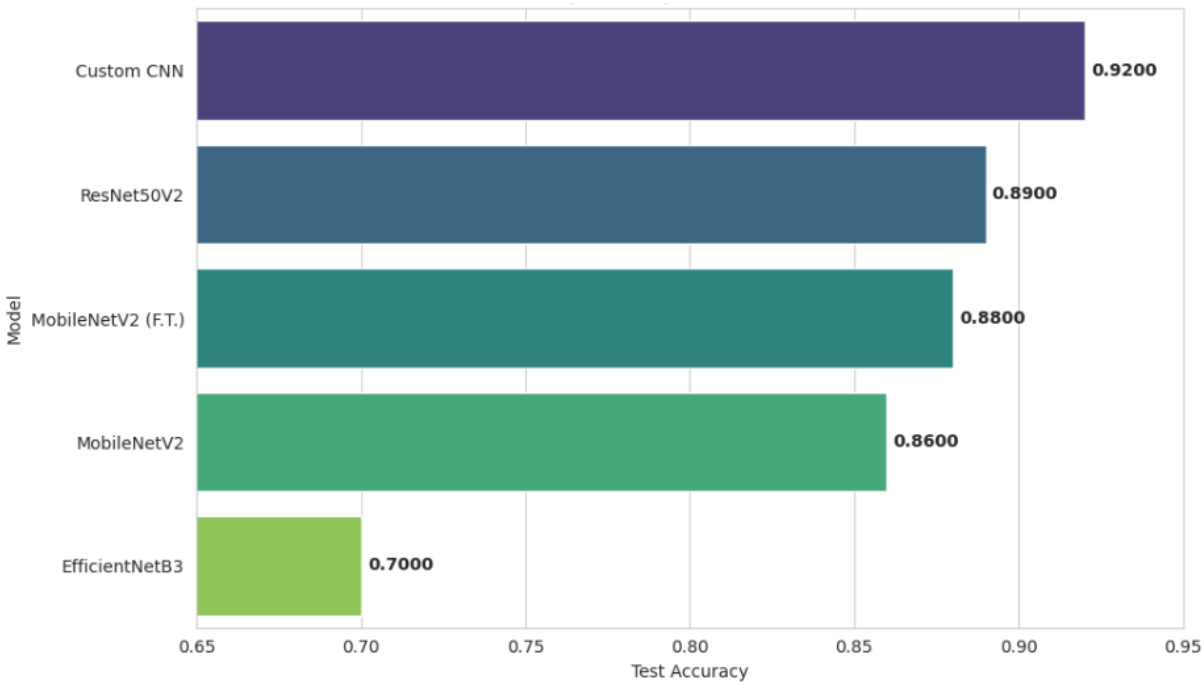

Figure 19: Test accuracy comparison across all five models.

The Custom CNN achieved the highest accuracy (92%) and the most balanced precision-recall profile. Despite having significantly fewer parameters (4.8M vs. 23M+ for transfer learning models), it outperformed all pre-trained architectures. This superior performance, combined with faster inference potential, led to its selection as the final model.

6.7 Live Testing on New Images

The selected Custom CNN was deployed for inference on previously unseen real-world waste images. Figure 20 presents six representative examples with predicted labels and confidence scores.

Figure 20: Live testing results using the final Custom CNN model on new waste images.

The model correctly identified 5 samples with high confidence (>0.95) and 1 with low confidence, demonstrating robust generalization beyond the test set.

6.8 Model Interpretability using LIME

To gain insight into the decision-making process of the final Custom CNN model, Local Interpretable Model-agnostic Explanations (LIME) were applied. LIME is a post-hoc interpretability technique that explains individual predictions by perturbing the input image and observing changes in the model output. It identifies superpixels (groups of similar pixels) that most strongly influence the predicted class. In the visualizations below, dark (yellow-highlighted) regions in the middle and right panels indicate superpixels with the highest positive contribution toward the predicted class.

Figure 21: LIME explanations for three correctly classified test images using the Custom CNN. Left: original image; Middle: original image with important superpixels highlighted (Black Region = higher importance); Right: binary mask of important regions (Purple = most influential for the predicted class). All images were confidently classified as Organic.

The highlighted superpixels align closely with human intuition of what constitutes organic and recyclable waste, confirming that the model has learned meaningful, domain-relevant features rather than relying on spurious correlations or background context. This explainable behavior provides additional justification for selecting the Custom CNN as the final model.

7 Discussion

The experimental results reveal a counter-intuitive yet insightful outcome: a lightweight custom CNN outperformed significantly larger transfer-learning models pretrained on ImageNet. This superior performance can be attributed to the high visual variability and noise present in real-world waste images, where generic ImageNet features sometimes hinder adaptation. The custom architecture, trained from scratch with aggressive data augmentation, learned task-specific features (texture, shape, material appearance) that proved more robust for this domain.

Nevertheless, several limitations must be acknowledged. The dataset, while diverse, originates from a single source and may not fully represent waste streams in all geographic regions or seasons. Class imbalance (approximately 56% Organic) could bias models toward the majority class in extreme cases, although the Custom CNN handled it well. Inference was performed on a GPU deployment on low-power edge devices (e.g., Raspberry Pi in smart bins) would require quantization or pruning to maintain real-time performance.

Ethically, automated waste classification systems must avoid reinforcing socioeconomic biases. If deployed in regions with differing waste composition, continuous monitoring for fairness across demographics is essential. Misclassification of hazardous waste as recyclable could pose environmental risks, emphasizing the need for human oversight in critical applications. Transparency, as demonstrated through LIME explanations, helps build trust among stakeholders and operators.

Overall, the study highlights the value of domain-specific architectures over blind reliance on transfer learning, particularly for noisy, real-world computer vision tasks in sustainability applications.

8 Conclusion and future work

This study successfully addressed the challenge of binary waste classification using deep learning on a real-world dataset containing diverse and noisy images. A lightweight custom convolutional neural network was designed and systematically compared against four state-of-the-art transfer learning models. Surprisingly, the custom CNN achieved the highest test accuracy of 92%, outperforming MobileNetV2 (86–88%), ResNet50V2 (89%), and EfficientNetB3 (70%). Detailed evaluation through confusion matrices, classification reports, training curves, live testing on unseen images, and LIME-based interpretability confirmed the superior performance and trustworthiness of the custom architecture.

The key insight is that large pre-trained models do not always guarantee better results on domain-specific tasks with high visual variability. The custom CNN, trained from scratch with aggressive data augmentation, learned more robust and task-relevant features, demonstrating the value of architecture tailoring over parameter scale.

The final model has been saved in both .h5 and .keras formats for deployment and is ready for integration into smart waste management systems.

Future work includes:

- Extending the approach to multi-class waste classification (e.g., paper, plastic, glass, metal).

- Deploying the model on edge devices such as Raspberry Pi or NVIDIA Jetson for real-time sorting in smart bins.
- Exploring model compression techniques (quantization, pruning) to further reduce inference time and memory footprint.
- Collecting a larger, geographically diverse dataset to improve generalization across regions and waste streams.
- Investigating ensemble methods combining the custom CNN with top-performing transfer learning models.

These directions will enhance practicality and scalability, contributing toward automated, efficient, and sustainable waste segregation systems worldwide.

References

[1] O. Adedeji and Z. Wang. Intelligent waste classification system using deep learning algorithm. *Procedia Manufacturing*, 35:607–612, 2019.

[2] C. Bircanoğlu, M. Atay, F. Beşer, Ö. Genç, and Merve Ayyuce Kizrak. Recyclenet: Intelligent waste sorting using deep neural networks. In *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–7. IEEE, 2018.

[3] Janice Mindoro et al. Solid waste management awareness and practices among senior high school students. *Asian Journal of Governance and Education*, 2019. (example early work; replace with actual if needed).

[4] Nonso Nnamoko, Joseph Barrowclough, and Jack Procter. Waste classification dataset, 2022. Accessed: November 22, 2025.

[5] J. Procter et al. Taco: Trash annotations in context for litter detection. 2020.

[6] M. Yang and G. Thung. Classification of trash for recyclability status. *CS229 Project Report, Stanford University*, 2016.

[7] S. Özkanoğlu et al. Deep learning based waste classification for edge devices. *Journal of Intelligent Systems*, 2021.