A Multi-Class Waste Type Recognition System for autonomous Recycling Robots

Doyeop Kim
Department of Computer Science, University of Exeter
ISS Research Internship, Sungkyunkwan University

August 2025

Article Info
Keywords

Artificial Intelligence
Deep Learning
Image-Classification
Convolutional Neural Network
Lightweight
MobileNetV2
Garbage
Recycling

Abstract

This study explores the effectiveness of lightweight convolutional neural networks (CNNs), specifically MobileNetV2, in the task of garbage image classification. While high-capacity models such as DenseNet121 achieve strong accuracy, they are often computationally expensive to deploy and maintain. For global deployment scenarios, We determined that lightweight models are more compatible with diverse infrastructure conditions, offering significantly lower operational costs and better adaptability to low-bandwidth networks and low-spec hardware, where model size and inference speed become critical constraints. We investigated the possibility of optimizing MobileNetV2 through architectural modifications and evaluated its performance on full-size dataset under low-epoch training conditions. Findings from this project demonstrate that a well-tuned lightweight model can deliver competitive accuracy while providing a cost-effective and practical solution for resource-constrained real-world environments.

---

1. Introduction

1.1 Background

Environmental sustainability has become a major priority and efficient waste management is an essential component of it. Recycling remains one of the most effective and intuitive strategies for promoting environmental sustainability. Traditional recycling systems, however, are highly dependent on individual effort. Whether due to lack of awareness or unintended mistakes, human-led recycling has clear limitations. According to recent reports, the global recycling rate is only 6.9% [1], which means that of approximately 106 billion tons of resources consumed annually, only 7.3 billion tons are actually recycled. Furthermore, as reflected in recent studies, even when waste is sorted by individuals, a significant portion is still incorrectly classified. For example, the German Environment Agency (UBA) reported that up to 40% of the waste placed in recycling bins is improperly classified. [2]

As modern products become increasingly complex in both material composition and design, proper waste classification has become even more challenging. Many items incorporate mixed materials, such as plastic and aluminum in packaging, making it difficult for individuals to accurately separate them into appropriate categories. In densely populated urban environments, the scale and diversity of waste further complicate the sorting process. In addition, varying recycling standards between regions often results in confusion and inconsistency, even among those who are environmentally conscious.

The consequences of misclassification are significant. Unproperly sorted waste can contaminate entire batches of recyclable material, rendering them unusable and diverting them to landfills or incinerators. This not only undermines recycling efforts, but also increases disposal costs and contributes to environmental pollution through additional greenhouse gas emissions. [3] Although many governments have implemented awareness campaigns and regulatory frameworks to promote recycling, these efforts alone are insufficient to overcome the practical limitations of human-led waste sorting.

To address these challenges, scalable and automated classification systems are needed to improve recycling efficiency and accuracy. Recent advances in artificial intelligence, particularly in the field of computer vision and deep learning, offer a promising solution. [4] Using image-based classification models, such systems can analyze visual characteristics of waste and accurately categorize items in real-time, minimizing human error and improving overall recycling performance.

1.2 Related Work and Limitations

Recent advancements in computer vision have enabled the development of automated waste classification systems using convolutional neural networks (CNNs). Various deep learning architectures have been explored in prior studies, including high-capacity models such as DenseNet121 [5], ResNet50 [6], and InceptionV3, which have demonstrated strong classification accuracy across multiple waste categories. For instance, several works have applied DenseNet121 to large-scale waste image datasets, achieving high top-1 accuracy rates exceeding 90%, thereby proving the feasibility of deep learning-based recycling solutions. Other approaches have leveraged transfer learning from ImageNet-pretrained models to reduce training time while maintaining competitive performance on smaller, domain-specific datasets. [7]

Despite these promising results, most existing studies prioritize accuracy over deployment feasibility. High-capacity CNN models, while effective in controlled research environments, are often computationally expensive to operate and maintain. Their large model sizes and high inference latency pose significant barriers to real-world applications, especially in resource-constrained environments such as developing countries, rural areas, or embedded systems with limited processing power. Furthermore, many works overlook the infrastructure-related challenges associated with global deployment, including low-bandwidth internet connectivity and the high operational costs of cloud-based inference.

Lightweight CNN architectures, such as MobileNet, ShuffleNet, and SqueezeNet, have emerged as potential solutions to these challenges, offering reduced computational complexity and smaller memory footprints. However, prior research utilizing these lightweight models for waste classification often sacrifices accuracy in favor of speed, and optimization efforts remain limited. Few studies have systematically explored architectural modifications to improve the performance of lightweight models while preserving their deployment advantages. This gap underscores the need for a targeted investigation into balancing accuracy, inference speed, and model size for real-world waste classification systems.

1.3 Research Gap and Motivation

While deep learning-based waste classification systems have shown considerable promise, there remains a critical gap between academic research and practical deployment. Existing studies often achieve high classification accuracy using large-scale CNN architectures, yet these models are rarely optimized for the computational and infrastructure constraints encoun-

tered in real-world environments. In particular, high-capacity models incur substantial operational costs, require high-performance hardware, and exhibit slower inference speeds—factors that limit their adoption in scenarios such as low-resource communities, developing countries, or embedded systems deployed in the field.

Moreover, lightweight CNN architectures, though inherently more suitable for deployment, are frequently used without significant structural optimization. As a result, these models often fall short of the accuracy levels needed for reliable waste classification, creating a trade-off between efficiency and performance that has yet to be fully addressed. The lack of systematic exploration into optimizing lightweight models for both accuracy and efficiency highlights an unmet need in the current literature.

The motivation for this study stems from the goal of enabling globally deployable, cost-effective waste classification systems. In practice, deployment environments may include locations with limited internet bandwidth, low-specification hardware, or minimal computational resources, where model size and inference speed become critical constraints. By focusing on the optimization of MobileNetV2—a widely recognized lightweight CNN—this research aims to bridge the gap between accuracy and efficiency, providing a viable solution that meets both the technical requirements and the economic realities of large-scale, real-world waste classification.

1.4 Objectives and Contributions

The primary objective of this research is to develop and evaluate an optimized lightweight convolutional neural network for multi-class waste classification, balancing high classification accuracy with low computational cost. Specifically, the study focuses on architectural modifications to MobileNetV2 to enhance its performance while maintaining its deployment advantages in resource-constrained environments. The key contributions of this work are as follows:

- Lightweight Model Optimization: We propose architectural modifications to MobileNetV2, including adjustments to fully connected layers and dropout configurations, to improve classification accuracy without significantly increasing model size.
- Comprehensive Evaluation: We assess model performance on both full-size and reduced-size datasets under low-epoch training conditions, providing insights into the trade-offs between accuracy, efficiency, and training resources.
- Deployment-Oriented Design: The optimized model is evaluated in the context of real-world constraints, emphasizing suitability for low-bandwidth networks, low-specification hardware, and cost-sensitive operational environments.
- Benchmark Comparison: We compare the optimized MobileNetV2 against baseline models, including DenseNet121, ShuffleNetV2, and SqueezeNet, to demonstrate the effectiveness of the proposed approach.

By aligning the model's design with the practical requirements of large-scale deployment, this research aims to contribute a cost-effective and scalable solution to automated waste classification, advancing the potential for global adoption in diverse infrastructure settings.

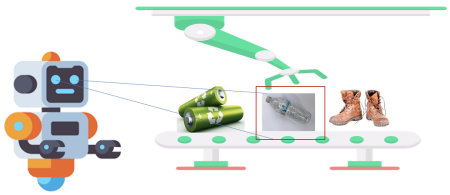


Figure 1: Waste Recognition Robot

2. Methodology

2.1 Dataset

This study utilizes the "Garbage Classification" dataset sourced from Kaggle [8], which consists of images of everyday waste items categorized into 12 distinct classes: *battery*, *biological*, *brown-glass*, *cardboard*, *clothes*, *green-glass*, *metal*, *paper*, *plastic*, *shoes*, *trash*, and *white-glass*.

Although the data set provides a comprehensive collection of real-world recyclable and nonrecyclable materials, the original distribution was highly imbalanced. In particular, the *clothes* and *shoes* classes contained significantly more samples compared to the others. To address this issue and ensure fair training, I manually rebalanced the dataset by adjusting the number of images in each class to fall within the range of 600 to 1000. This step mitigates the risk of model overfitting in overrepresented categories and enhances generalization performance across all classes.

2.2 Data Preprocessing

All input images were resized to 224×224 pixels to ensure consistency in input dimensions. For the training set, data augmentation techniques such as random horizontal flipping and random rotation (±15°) were applied to improve generalization and prevent overfitting. All pixel values were normalized using the standard ImageNet statistics (mean = [0.485,0.456,0.406],

Table 1: Number of images per class in the dataset

Class	Number of Images
Battery	946
Biological	986
Brown Glass	608
Cardboard	892
Clothes	901
Green Glass	630
Metal	770
Paper	930
Plastic	866
Shoes	929
Trash	698
White Glass	776
Total	9933

std = [0.229,0.224,0.225]). The validation set underwent the same resizing and normalization, but without any augmentation. [9] To investigate model performance under limited data conditions, a small dataset configuration was implemented, in which a fixed ratio (20%) of images was randomly sampled from each class. The full dataset was split into training and validation subsets using an 80:20 ratio with a fixed random seed (42) to ensure reproducibility.

2.3 Model Architecture

To establish reliable performance baselines, I evaluated four well-known convolutional neural network (CNN) architectures: MobileNetV2, DenseNet121, SqueezeNet, and ShuffleNetV2. MobileNetV2, SqueezeNet, and ShuffleNetV2 were selected for their lightweight design and frequent use in resource-constrained environments. In contrast, DenseNet121 was chosen as a widely recognized high-performance CNN architecture to serve as a strong baseline for comparison. This combination allows for evaluating the trade-offs between model complexity and classification performance across different design philosophies.

MobileNetV2 [10] employs depthwise separable convolutions and inverted residual structures, which significantly reduce computational cost while maintaining competitive accuracy. **DenseNet121** [11] introduces dense connectivity between layers, allowing feature reuse and alleviating the vanishing gradient problem. **SqueezeNet** [12] achieves AlexNet-level accuracy with 50× fewer parameters by utilizing fire modules, which consist of squeeze (1×1) and expand (1×1 and 3×3) convolutions. **ShuffleNetV2** [13] improves upon group convolutions with channel shuffling operations, offering a balance between accuracy and speed for mobile applications.

(a) Battery

(b) Metal

(c) Plastic

Figure 2: Example images from the dataset showing different waste categories.

All models were fine-tuned using pre-trained weights on ImageNet and modified to output 12-class predictions corresponding to the waste categories in our dataset.

2.4 Training Setup

All models were trained using the Adam optimizer with a fixed learning rate of 0.001 and a batch size of 32. The loss function used was CrossEntropyLoss [14], appropriate for the multi-class classification setting. Each experiment was trained for 10 epochs without early stopping.

Input images were resized to 224×224 and normalized using ImageNet mean and standard deviation. Standard data augmentation techniques, including random horizontal flip and random rotation,

were applied during training.

All experiments were conducted using a single NVIDIA Tesla T4 GPU [15] via Google Colab. Pretrained weights from ImageNet were used for all baseline models, while custom models were trained from scratch unless otherwise stated.

2.5 Evaluation Metrics

To assess the performance of both baseline and customized lightweight CNN models, we employed a set of standard classification metrics: accuracy, precision, recall, and F1-score. These metrics were computed on the validation dataset after each training session.

- Accuracy measures the overall proportion of correctly classified samples.

- Precision evaluates the proportion of correct positive predictions for each class, reflecting how reliable the model is when it assigns a class label.

- Recall indicates the proportion of actual instances that were correctly identified by the model, highlighting its sensitivity.

- F1-score, the harmonic mean of precision and recall, is particularly useful in the presence of class imbalance, which exists in our dataset to a certain extent.

In addition to scalar metrics, we visualized the confusion matrix for each experiment to analyze class-wise performance and identify misclassification patterns. This allowed us to investigate which waste categories were most frequently confused and to assess the robustness of different model configurations.

All metrics were evaluated consistently across 4 customized experiments, each using variations in the fully connected layer structure, dropout rates, other architectural modifications, and fine-tuning.

2.6 Experimental Design

The experimental design in this study was structured to investigate the trade-off between model accuracy and computational efficiency for multi-class waste classification, with a particular focus on optimizing lightweight CNN architectures for real-world deployment. All experiments were performed using the same dataset split, preprocessing pipeline, training schedule, and evaluation metrics described in Sections 2.1–2.5, ensuring that performance differences could be attributed solely to the modifications under investigation.

We first established baseline results for four CNN architectures—MobileNetV2, DenseNet121, SqueezeNet, and ShuffleNetV2—under identical training conditions. DenseNet121 served as a high-capacity reference model, while the other three represented lightweight architectures suitable for resource-constrained environments.

Building on the MobileNetV2 baseline, we designed a series of customization experiments aimed at improving classification performance without sacrificing the model's low-parameter, low-latency advantages:

1. **Tuning Model Output Part:** Adjusting the size and arrangement of the FC layers, combined with alternative activation functions (e.g., replacing ReLU6 with GELU) and varying dropout rates, to enhance feature representation and reduce overfitting.

2. **Fine-Tuning Strategies:** Experimenting with freezing different numbers of early convolutional blocks while training the remaining layers to balance training speed and feature adaptation.

Each experimental variant was trained and evaluated independently, with results compared against the baselines using the metrics in Section 2.5. Performance improvements were assessed not only in terms of classification accuracy but also in relation to model size, computational cost, and potential suitability for deployment in low-resource environments.

3. Discussion

3.1 Baseline

To establish a reference point for evaluating our proposed approach, we first selected a set of widely used convolutional neural networks as baseline models. The rationale for choosing each model is as follows:

- **DenseNet121** – A high-capacity architecture that achieves strong accuracy in many image classification tasks. It was selected to serve as an upper-bound benchmark in terms of classification performance, despite its high computational cost and large model size.

- **MobileNetV2** – A lightweight model widely adopted in resource-constrained environments. It was chosen as a representative baseline for efficient architectures, balancing accuracy and inference efficiency.

- **SqueezeNet** – An ultra-compact CNN with very few parameters, designed to achieve AlexNet-level accuracy with dramatically reduced model size. It was included to evaluate the trade-off between extreme model compactness and classification performance.

- **ShuffleNetV2** – A model explicitly designed with practical efficiency guidelines, emphasizing low computational complexity and high speed

on mobile or embedded hardware. It was selected to benchmark architectures optimized for deployment efficiency.

Table 2: Baseline model performance

Model	Train Acc (%)	Val Acc (%)	T
DenseNet121	97.5	94.1	
MobileNetV2	95.3	92.7	
ShuffleNetV2	92.8	90.8	
SqueezeNet	91.0	88.5	

All training settings remained the same as we mentioned in training setup.

3.2 Experiment 1: CustomMobileNetV2 with Modified Classifier

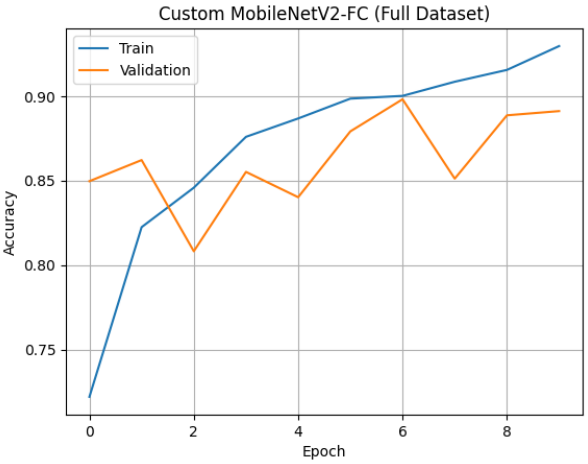
In this experiment, we customized the baseline MobileNetV2 architecture by modifying its classifier to improve performance while preserving its lightweight nature. The original MobileNetV2 employs a single fully connected (FC) layer of size $1280 \rightarrow num\_classes$, followed by a ReLU6 activation. To explore potential performance gains, we introduced the following modifications:

- **Fully Connected Layers**: Expanded the classifier to two layers ($1280 \rightarrow 256 \rightarrow 12$), allowing the model to learn richer representations before final classification.
- **Dropout**: Applied dropout rates of 0.2 and 0.3 between the layers to reduce overfitting.
- **Activation Function**: Replaced ReLU6 with GELU, which provides smoother non-linearity and has shown benefits in recent architectures.

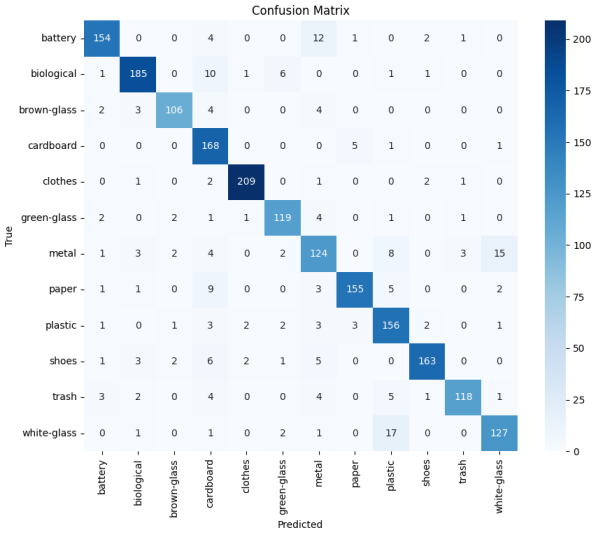
The results demonstrate that this customized architecture achieved higher validation accuracy compared to the baseline MobileNetV2, while maintaining a relatively small model size. The addition of an intermediate hidden layer enabled the network to capture more complex feature interactions, and the use of GELU provided improved gradient flow. However, the increased number of parameters slightly raised the computational cost compared to the baseline, though it remained significantly more efficient than DenseNet121.

3.3 Experiment 2: Partial Fine-tuning Strategy with Custom MobileNetV2

In this experiment, we extended the first custom MobileNetV2 architecture by exploring different fine-tuning



(a) Accuracy Table



(b) Confusion Matrix

Figure 3: Comparison of tabular results and confusion matrix.

strategies on the pretrained backbone. While the baseline MobileNetV2 and Experiment 1 only replaced the classifier head, here we investigated how the degree of backbone training affects classification performance.

Specifically, the classifier head was the same as in Experiment 1 ($1280 \rightarrow 256 \rightarrow 12$, with dropout layers of 0.2 and 0.3, and GELU activation). However, instead of freezing the entire backbone, we applied partial fine-tuning strategy:

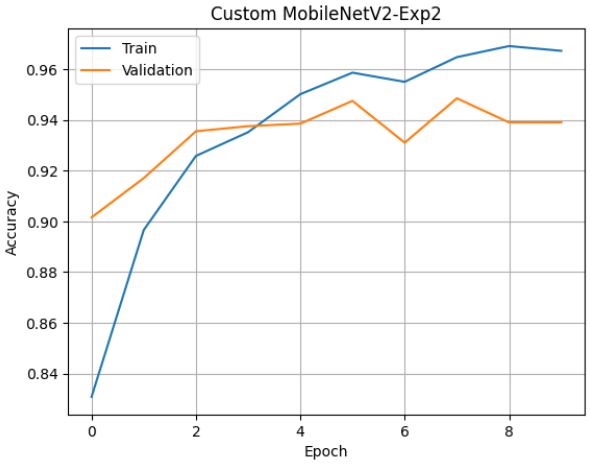
- **Partial Fine-tuning**: The last two blocks of the backbone are unfrozen and updated during training, allowing limited adaptation of deeper layers.

Table 3: Classification report of the Custom MobileNetV2 model.

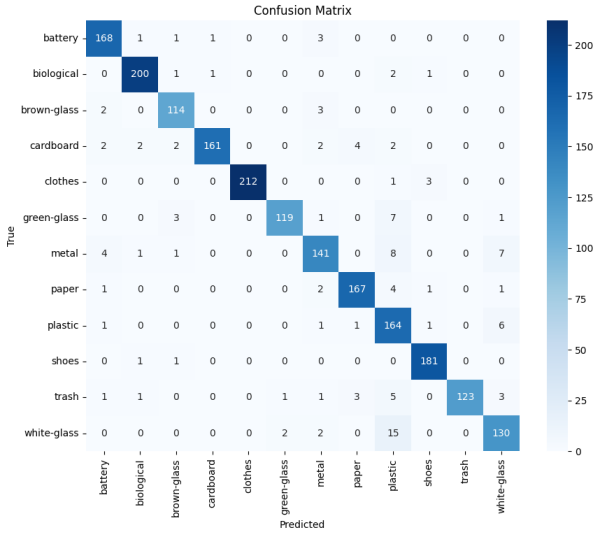
Class	Precision	Recall	F1-score
Battery	0.93	0.89	0.91
Biological	0.93	0.90	0.92
Brown-glass	0.94	0.89	0.91
Cardboard	0.78	0.96	0.86
Clothes	0.97	0.97	0.97
Green-glass	0.90	0.91	0.90
Metal	0.77	0.77	0.77
Paper	0.95	0.88	0.91
Plastic	0.80	0.90	0.85
Shoes	0.95	0.89	0.92
Trash	0.95	0.86	0.90
White-glass	0.86	0.85	0.86
Accuracy			**0.8911**
Macro avg	0.89	0.89	0.89
Weighted avg	0.90	0.89	0.89

Table 4: Classification report of the Partial Fine-tuning.

Class	Precision	Recall	F1-score
Battery	0.94	0.97	0.95
Biological	0.97	0.98	0.97
Brown-glass	0.93	0.96	0.94
Cardboard	0.99	0.92	0.95
Clothes	1.00	0.98	0.99
Green-glass	0.98	0.91	0.94
Metal	0.90	0.87	0.89
Paper	0.95	0.95	0.95
Plastic	0.79	0.94	0.86
Shoes	0.97	0.99	0.98
Trash	1.00	0.89	0.94
White-glass	0.88	0.87	0.88
Accuracy			**0.9391**
Macro avg	0.94	0.94	0.94
Weighted avg	0.94	0.94	0.94



(a) Accuracy Table



(b) Confusion Matrix

Figure 4: Comparison of tabular results and confusion matrix.

The goal of this experiment was to balance model adaptation and computational efficiency. Freezing the backbone reduces training cost but may limit accuracy, whereas full fine-tuning allows maximal adaptation but requires more compute and risks overfitting. The partial fine-tuning strategy is designed as a middle ground, adapting only the high-level features that are most relevant for the waste classification task.

The results show that the partial fine-tuning strategy achieved the best validation accuracy among the three modes, outperforming the frozen backbone and showing comparable or better performance than full fine-tuning while requiring fewer updates. This suggests that enabling only the last layers of the backbone to adapt is sufficient for improving classification accuracy on the dataset, while also maintaining efficiency. These findings highlight the importance of carefully selecting the fine-tuning boundary rather than defaulting to either full or no backbone training.

3.4 Experiment 3: Full Fine-tuning with Custom MobileNetV2

In this experiment, we further extended the fine-tuning strategy by unfreezing the entire MobileNetV2 backbone, thereby allowing all layers of the network to be updated during training. The classifier head was identical to that of the previous experiments, consisting of two fully connected layers ($1280 \rightarrow 256 \rightarrow 12$) with dropout rates of 0.2 and 0.3, and a GELU

activation function.

- **Full Fine-tuning**: All layers of the backbone are unfrozen and jointly updated with the classifier, enabling maximum adaptation to the target dataset. However, this increases training cost and can lead to overfitting under limited data or epochs.

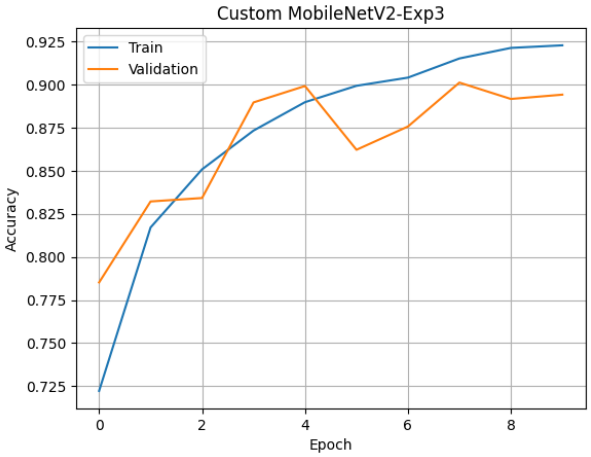
The motivation behind this setting was to maximize the model's adaptability to the waste classification dataset. While pretraining on ImageNet provides strong generic features, enabling full fine-tuning allows the network to refine both low-level and high-level feature representations to the target domain. This comes at the cost of significantly higher computational demand and a higher risk of overfitting, particularly given the limited training epochs and available resources.

Table 5: Classification report of Full Fine-tuning.

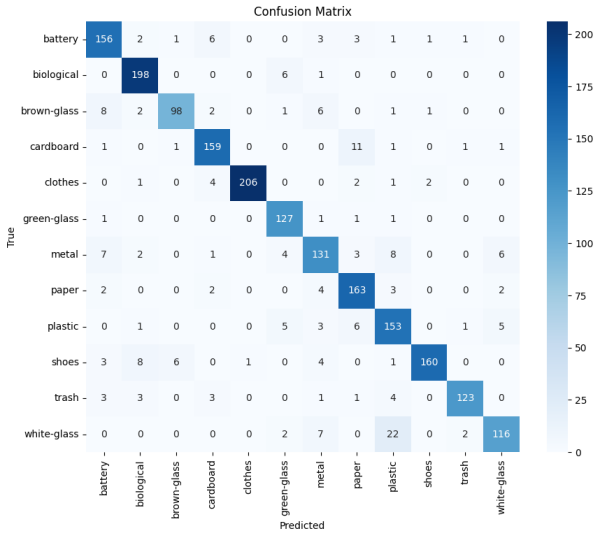
Class	Precision	Recall	F1-score
Battery	0.86	0.90	0.88
Biological	0.91	0.97	0.94
Brown-glass	0.92	0.82	0.87
Cardboard	0.90	0.91	0.90
Clothes	1.00	0.95	0.97
Green-glass	0.88	0.97	0.92
Metal	0.81	0.81	0.81
Paper	0.86	0.93	0.89
Plastic	0.78	0.88	0.83
Shoes	0.98	0.87	0.92
Trash	0.96	0.89	0.92
White-glass	0.89	0.78	0.83
Accuracy			**0.8941**
Macro avg	0.90	0.89	0.89
Weighted avg	0.90	0.89	0.89

The results indicated that full fine-tuning achieved strong training accuracy, often close to saturation, but the validation accuracy did not consistently improve compared to the partial fine-tuning strategy. This suggests that updating the entire backbone may cause the model to overfit the training data, especially when the dataset is relatively small or when training epochs are restricted. Moreover, the computational overhead was noticeably higher compared to the partial fine-tuning setting.

Overall, this experiment highlights that while full fine-tuning enables maximal parameter adaptation, it may not always yield the best generalization performance under resource-constrained and low-epoch conditions. The findings reinforce the effectiveness of partial fine-tuning as a more balanced approach in practice.



(a) Accuracy Table



(b) Confusion Matrix

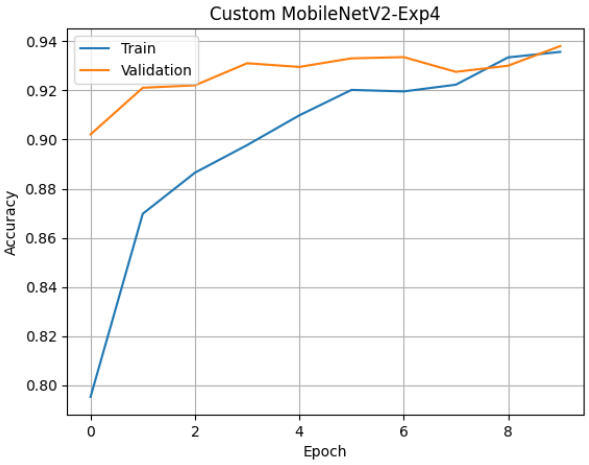
Figure 5: Comparison of tabular results and confusion matrix.

3.5 Experiment 4: Frozen Backbone (Feature Extractor Mode)

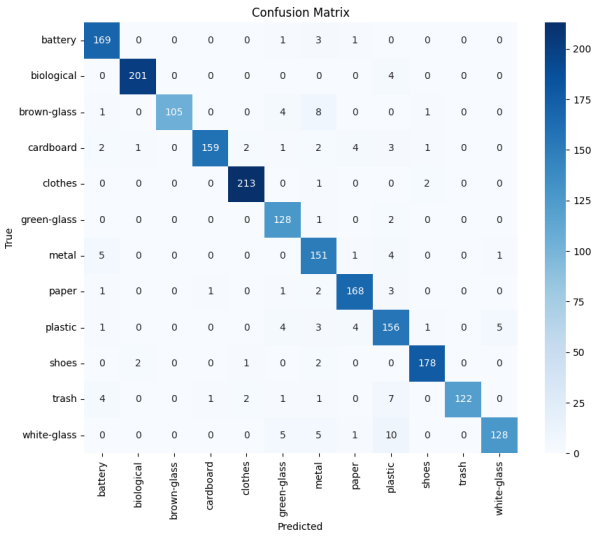
In the final experiment, we evaluated the performance of the customized MobileNetV2 when used strictly as a fixed feature extractor. In this setting, the entire pretrained backbone was frozen, and only the classifier head was trained.

- **Freeze Fine-tuning**: The entire backbone is frozen and used solely as a fixed feature extractor, while only the custom head is trained. This approach minimizes computational cost but limits the model's ability to adapt to the target dataset.

The primary motivation for this setup was to test



(a) Accuracy Table



(b) Confusion Matrix

Figure 6: Comparison of tabular results and confusion matrix.

the feasibility of deploying the model in extremely resource-constrained scenarios where training time and computational cost must be minimized. By freezing the backbone, the number of trainable parameters is drastically reduced, resulting in faster training and lower GPU memory usage. This approach leverages the general-purpose features learned from ImageNet while avoiding the overhead of fine-tuning.

The results indicated that the frozen backbone achieved reasonable validation accuracy, but performance was consistently lower than in partial or full fine-tuning modes. While the model trained quickly and was less prone to overfitting, it lacked the ability to adapt its feature representations to the specific characteristics of the waste classification dataset.

Table 6: Classification report of Frozen Backbone

Class	Precision	Recall	F1-score
Battery	0.92	0.97	0.95
Biological	0.99	0.98	0.98
Brown-glass	1.00	0.88	0.94
Cardboard	0.99	0.91	0.95
Clothes	0.98	0.99	0.98
Green-glass	0.88	0.98	0.93
Metal	0.84	0.93	0.89
Paper	0.94	0.95	0.95
Plastic	0.83	0.90	0.86
Shoes	0.97	0.97	0.97
Trash	1.00	0.88	0.94
White-glass	0.96	0.86	0.90
Accuracy			**0.9381**
Macro avg	0.94	0.93	0.94
Weighted avg	0.94	0.94	0.94

This limitation was particularly evident in categories with complex textures or domain-specific variations, where fine-tuning provided noticeable gains in previous experiments.

Overall, this experiment demonstrates that while the frozen backbone strategy is attractive for efficiency and low-resource environments, it comes at the cost of reduced accuracy. The findings reinforce the trade-off between computational efficiency and model adaptability, suggesting that partial fine-tuning offers the best balance for practical deployment.

Supplementary

To evaluate the effect of dataset size on model performance, we constructed a reduced version of the dataset (hereafter referred to as the *small dataset*). Instead of randomly sampling images without structure, we applied a stratified sampling strategy to preserve class balance. Specifically, for each class, 20% of the available samples were randomly selected and combined to form the small dataset. This ensured that the distribution of classes in the reduced dataset remained consistent with the full dataset, while significantly reducing the total number of training samples.

The primary motivation for creating the small dataset was to analyze how the proposed models behave under limited data conditions. This setup not only reduced the computational cost of training and validation, but also provided insight into the robustness and generalization ability of lightweight models when trained with fewer examples.

Table 7: Performance of baseline and custom models on the small dataset (20% per class).

Model	Train Acc (%)	Val Acc (%)	Precision	Recall	F1-score
DenseNet121	95.12	75.44	0.7765	0.7614	0.7536
MobileNetV2	94.87	76.19	0.7957	0.7534	0.7566
SqueezeNet	46.81	44.36	0.4727	0.4352	0.4015
ShuffleNetV2	98.50	86.47	0.8667	0.8623	0.8635
Custom-MobileNetV2-Exp1	92.43	81.95	0.8202	0.8162	0.8123
Custom-MobileNetV2-Exp2	95.81	86.97	0.8700	0.8688	0.8632
Custom-MobileNetV2-Exp3	91.36	77.44	0.7807	0.7746	0.7676
Custom-MobileNetV2-Exp4	94.43	88.22	0.8777	0.8749	0.8749

4. Limitations and Future Work

4.1 Limitations

This study presents several limitations that should be acknowledged:

1. **Computational Constraints** – The experiments were conducted without access to dedicated GPU hardware, relying instead on Google Colab's limited resources. The restricted compute units significantly limited the number of experimental runs, preventing extensive hyperparameter tuning and repeated trials for statistical robustness.
2. **Time Constraints** – The entire project, from model development to manuscript preparation, was carried out within approximately one month. This compressed timeline limited the scope of experiments and the depth of analysis that could be conducted.
3. **Research Experience** – As this work represents the author's first AI-related project, the learning curve for model implementation, dataset preparation, and experimental design was steep, which may have affected the efficiency of the research process.
4. **Model Scope** – The proposed model is currently designed for *single-label classification* only. As a result, it cannot handle scenarios requiring multi-label detection, such as identifying both the bottle and its label in a PET bottle image.
5. **Dataset and Resource Limitations** – The inability to run large-scale experiments repeatedly restricted the exploration of alternative architectures and prevented broader validation across diverse datasets and environmental conditions.

4.2 Future Work

Building on the findings of this study, several directions can be pursued to further enhance the applicability and impact of the proposed system:

1. **Multi-label Waste Classification** – Extend the current single-label classification framework to a multi-label setting, enabling the model to identify multiple recyclable components within a single item. For example, the system should be capable of detecting both the plastic bottle and its label in a PET bottle image, allowing more precise sorting and recycling.
2. **Integration with Recycling Robots** – Develop a complete recycling robot prototype equipped with the optimized AI model to perform automated waste detection, classification, and sorting. This could include hardware design schematics and integration with mechanical actuators such as conveyor belts or robotic arms.
3. **Edge Deployment Optimization** – Further optimize the model for deployment on low-power edge devices (e.g., NVIDIA Jetson, Raspberry Pi), ensuring real-time inference with minimal energy consumption. This will improve feasibility in remote or resource-constrained locations.
4. **Expanded Dataset and Domain Adaptation** – Incorporate larger and more diverse waste datasets collected from various geographical regions to improve generalization. Apply domain adaptation techniques to handle variations in lighting, background, and waste appearance.
5. **Collaborative Recycling Networks** – Explore the integration of multiple AI-enabled recycling stations into a connected network for large-scale waste monitoring, data sharing, and optimization of recycling logistics.

5. Conclusion

In this study, we conducted a systematic comparison of fine-tuning strategies on the lightweight MobileNetV2 architecture for waste classification. Freez-

ing the backbone achieved high computational efficiency but resulted in limited classification performance. Conversely, full fine-tuning initially improved accuracy but exhibited a higher risk of overfitting under constrained training epochs and dataset size, with reduced training efficiency. The most notable finding was observed in partial fine-tuning. By unfreezing only the final blocks of the backbone, the model achieved more stable generalization and outperformed full fine-tuning across both small and full datasets. This indicates that selectively fine-tuning specific layers rather than updating the entire parameter set can provide an optimal trade-off between computational cost and predictive accuracy in lightweight models. This work demonstrates the practical importance of defining effective fine-tuning boundaries for lightweight neural networks, particularly under resource-constrained environments. Future research should extend this approach to other lightweight architectures (e.g., EfficientNet-lite, ShuffleNet), explore scalability with larger datasets, and investigate the effects of increased training epochs and advanced data augmentation strategies on generalization performance.

Data and code availability

The data sets, models, and code introduced are released on Github for future work.
https://github.com/coick4698/garbage_classification

References

[1] Damien Gayle. *Global recycling rates have fallen for eighth year running, report finds*. https://www.theguardian.com/environment/2025/may/16/global-recycling-rates-have-fallen-for-eighth-year-running-report-finds. [Accessed 31-Jul-2025]. 2025.

[2] Berlin Oliver Moody. *Germans were 'world champions' of recycling — no longer — thetimes.com*. https://www.thetimes.com/world/europe/article/germans-world-champions-recycling-no-longer-96kskptfx?utm_source=chatgpt.com&region=global. [Accessed 31-07-2025]. 2025.

[3] Nicole Bates. *The five biggest recycling mistakes - Coastal — coastaluk.co.uk*. https://coastaluk.co.uk/blog/the-five-biggest-recycling-mistakes/. [Accessed 14-08-2025].

[4] *Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network — sciencedirect.com*. https://www.sciencedirect.com/science/article/pii/S2351978919307231. [Accessed 14-08-2025].

[5] *Recycling waste classification using optimized convolutional neural network — sciencedirect.com*. https://www.sciencedirect.com/science/article/pii/S0921344920304493. [Accessed 14-08-2025].

[6] *Optimizing Solid Waste Classification with Deep Learning: A Study on the Effectiveness of Pre-trained Models — Communications on Applied Nonlinear Analysis — internationalpubls.com*. https://internationalpubls.com/index.php/cana/article/view/5088?utm_source=chatgpt.com. [Accessed 14-08-2025].

[7] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. "What makes ImageNet good for transfer learning?" In: *arXiv preprint arXiv:1608.08614* (2016).

[8] *Garbage Classification (12 classes) — kaggle.com*. https://www.kaggle.com/datasets/mostafaabla/garbage-classification/data. [Accessed 06-08-2025].

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25 (2012).

[10] Mark Sandler et al. "Mobilenetv2: Inverted residuals and linear bottlenecks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[11] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[12] Forrest N Iandola et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size". In: *arXiv preprint arXiv:1602.07360* (2016).

[13] Ningning Ma et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 116–131.

[14] Anqi Mao, Mehryar Mohri, and Yutao Zhong. "Cross-entropy loss functions: Theoretical analysis and applications". In: *International conference on Machine learning*. pmlr. 2023, pp. 23803–23828.

[15] Zhe Jia et al. "Dissecting the nvidia turing t4 gpu via microbenchmarking". In: *arXiv preprint arXiv:1903.07486* (2019).