

Benchmarking Modern CNN Architectures for Transfer Learning-Based Recyclable Waste Classification

Novi Febrianti^{1*}, Kamarul Hawari Ghazali¹, Jusmardi², Melrideswina², Rahmadani Ira², Taufiqul Israt³

¹Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia.

² Faculty of Engineering, Universitas Negeri Padang, Indonesia.

³Informatics Study Program, Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang, Indonesia.

Article Info

Article history:

Received June 15, 2026

Revised September 18, 2026

Accepted September 25, 2026

Keywords:

computer vision
convolutional neural networks
deep learning
model benchmarking
recyclable waste classification
transfer learning

ABSTRACT

The growing need for automatic sorting of recyclable materials has led researchers to adopt Convolutional Neural Network (CNN)-based image classification as a practical solution. A persistent challenge in this line of research, however, is that prior studies rarely use matching datasets, preprocessing pipelines, or evaluation schemes, making direct comparisons across studies unreliable. To address this gap, six widely used CNN architectures, namely ResNet50, MobileNetV2, MobileNetV3-L, EfficientNetB0, ShuffleNetV2, and InceptionV3, were evaluated in this study under a single, fixed experimental setup instead of relying on results drawn from separate studies with differing conditions. This controlled, side-by-side design, applying identical data and an identical protocol to six architectures with fundamentally different design philosophies, constitutes the main contribution of this work. Testing was carried out on 4,000 images spanning four waste categories (glass, cans, paper, and plastic), split 80/10/10 for training, validation, and testing, with all models relying on transfer learning through frozen, pretrained ImageNet feature extractors. Among the six architectures, ResNet50 yielded the best performance, reaching 86.00% accuracy, 86.10% precision, 86.00% recall, and an F1-score of 85.64%, while MobileNetV2 and MobileNetV3-L both achieved 83.50% accuracy with considerably lower computational cost. Taken together, these findings offer practical guidance for selecting a suitable CNN architecture in automated waste-sorting systems, depending on whether classification accuracy or model efficiency is the higher priority for a given deployment.

Corresponding Author:

Novi Febrianti

Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia.

Email: pfz24016@adab.umpsa.edu.my

Kamarul Hawari Ghazali

Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang Al-Sultan Abdullah (UMPSA), Malaysia.

Email: kamarul@umpsa.edu.my

1. INTRODUCTION

Efficient waste management increasingly relies on automated sorting mechanisms capable of identifying recyclable materials without manual intervention [1], [2]. Among the methods proposed for this purpose, computer vision techniques based on Convolutional Neural Networks (CNNs) have proven particularly effective at recognizing waste categories from visual features such as shape, texture, and color [3], [4], [5]. A

key factor behind this effectiveness is transfer learning, which allows a model to reuse feature representations already learned from large-scale datasets and adapt them to a smaller, task-specific dataset rather than training a network from scratch [6], [7], [8].

Despite this progress, comparing results across existing studies remains difficult because CNN performance depends heavily on both the chosen architecture and the surrounding experimental setup. For instance, MobileNetV2 and EfficientNetB0 have been shown to perform differently when evaluated under the same task [6], and a similar architecture-dependent gap has been reported between MobileNetV2 and VGG16, both in terms of accuracy and computational cost [7]. Beyond architecture alone, the choice of preprocessing method can noticeably shift classification outcomes [9], and factors such as dataset composition or real-world imaging conditions further affect how well a model generalizes [10]. A closer look at these studies, however, reveals a more fundamental limitation: each was designed to answer a narrow, self-contained question, typically comparing only two architectures at a time, under a preprocessing pipeline and dataset configuration tailored specifically to that study. As a result, none of them offers a basis for ranking architectures against one another beyond the pair they originally investigated, and any apparent advantage reported for one architecture cannot be reliably attributed to the architecture itself rather than to the particular experimental conditions under which it was tested. This is precisely the gap this study seeks to close: the literature currently lacks a benchmark that evaluates a broader set of architecturally diverse CNNs side by side, under one shared protocol, so that performance differences can be attributed to architecture rather than to inconsistent experimental conditions [2], [6], [7], [8], [9], [10].

Motivated by this need, six CNN architectures were selected for evaluation in this study: EfficientNetB0, ResNet50, InceptionV3, MobileNetV2, MobileNetV3-L, and ShuffleNetV2. Rather than being chosen arbitrarily, these six represent distinct design philosophies within the CNN family: EfficientNetB0 relies on compound scaling of depth, width, and resolution; ResNet50 introduces residual connections to ease the training of deep networks; InceptionV3 extracts features at multiple spatial scales through parallel convolutional branches; and MobileNetV2, MobileNetV3-L, and ShuffleNetV2 prioritize computational efficiency through lightweight design choices. These architectures were preferred over more recent alternatives, such as transformer-based vision models, for two practical reasons. First, all six have mature, well-documented pretrained weights and established deployment support, which is a prerequisite for the resource-constrained embedded systems typically used in automated waste-sorting hardware. Second, their diversity in parameter counts and computational cost, ranging from lightweight mobile-oriented designs to deeper, more computationally demanding networks, allows the trade-off between accuracy and efficiency to be examined directly, which is of greater practical relevance to sorting-system deployment than testing architectures that are largely similar in design. This diversity allows the study to examine how differences in representational capacity and computational complexity translate into classification performance once all other experimental variables are held constant.

The novelty of this study lies in closing the gap identified above: rather than comparing two architectures at a time under study-specific conditions, as in prior work, six architecturally diverse CNNs are evaluated together under a single, fixed experimental protocol, covering identical data splits, preprocessing, augmentation, training configuration, and evaluation metrics. It should be noted that this study does not introduce a new CNN architecture; its contribution lies instead in the controlled, multi-architecture comparison itself, together with an analysis of the specific misclassification patterns that occur across the four waste categories: glass, cans, paper, and plastic. In practical terms, the resulting benchmark is intended to help practitioners select an architecture appropriate to a given deployment context, whether the priority is maximizing classification accuracy or minimizing computational demand for resource-constrained sorting hardware.

Beyond the limited number of architectures compared in prior studies, a closer examination reveals that their limitations also extend to the dimensions along which performance was evaluated. Most previous studies assessed the success of an architecture almost exclusively through final accuracy values, without incorporating an analysis of class-level misclassification patterns or considerations of model complexity as part of their conclusions [6], [7], [9]. Yet in practical deployment contexts, two architectures with comparable accuracy can exhibit markedly different failure characteristics, for instance, a systematic tendency to misclassify a particular category, or can require substantially different computational resources, rendering one considerably more suitable than the other for implementation on resource-constrained hardware. The absence of such analyses in prior work makes it difficult to translate their conclusions into actionable recommendations for the design of automated waste-sorting systems, which in practice must account not only for the architecture with the highest accuracy but also for the architecture best suited to hardware constraints and tolerable error patterns. This study therefore extends prior work not only by broadening the comparison from two to six architectures, but also by broadening the evaluation itself, incorporating class-level misclassification analysis and parameter-complexity comparison as integral components of its conclusions.

The practical contribution of the resulting benchmark is intended to be directly applicable to those designing or developing computer vision-based automated waste-sorting systems, particularly at the initial stage of architecture selection. Rather than independently testing architectures from scratch, or relying on findings from prior studies whose experimental conditions vary and are therefore not directly comparable, the results of this study provide a single reference point that can be used to identify both the architecture that excels in classification accuracy and the architecture that excels in computational efficiency, with both evaluated under fully equivalent experimental conditions. The remainder of this article is organized accordingly: Section 2 details the dataset, preprocessing procedures, and training configuration applied identically to all six architectures; Section 3 presents the performance comparison results, class-level misclassification analysis, and model complexity comparison; and Section 4 summarizes the main findings and their implications for architecture selection in real-world recyclable waste-sorting applications.

2. METHOD

This study employed a controlled comparative experimental approach to evaluate the performance of six Convolutional Neural Network (CNN) architectures, namely EfficientNetB0, ResNet50, InceptionV3, MobileNetV2, MobileNetV3-L, and ShuffleNetV2, for recyclable waste classification. All models were trained and evaluated using the same dataset, data split, preprocessing and augmentation procedures, transfer learning strategy, training configuration, and evaluation metrics. This approach was adopted to ensure an objective comparison and to identify the architecture with the best classification performance across four recyclable waste categories: glass, cans, paper, and plastic.

2.1 Dataset Collection and Preparation

This study relies on a dataset organized specifically to enable a fair, controlled comparison across the six CNN architectures under investigation. Images were grouped into four recyclable waste categories, namely glass, cans, paper, and plastic, and the exact same set of images was used for every architecture tested, so that no model was given an advantage through a different or more favorable data sample.

From this shared pool, the data were partitioned into three subsets serving distinct purposes. The training subset was used exclusively to update the parameters of the classification head, while the validation subset served to track performance throughout training and to determine which set of weights performed best. Final performance, by contrast, was assessed only once, on the held-out test subset. Because this same partitioning scheme was applied identically across all six architectures, any performance differences observed later can be attributed to the architectures themselves rather than to inconsistencies in the samples each model happened to see.

Prior to training, all images passed through a preprocessing and augmentation pipeline tailored to the requirements of each architecture, and every model was initialized with ImageNet-pretrained weights under a transfer learning scheme. Specifically, the convolutional backbone of each network was kept frozen throughout training, meaning that only the classification head, appended on top of each backbone, was allowed to learn from the waste dataset. Beyond this shared strategy, several training settings, namely the number of epochs, batch size, optimizer, learning rate, loss function, and evaluation metrics, were held constant across all six models to further isolate architecture as the sole variable under study. The only exceptions permitted were architecture-specific requirements that could not reasonably be standardized, such as the input resolution and the auxiliary classifier used internally by InceptionV3.

Taken together, the full research workflow proceeded through seven stages: preparing the dataset, splitting it into subsets, preprocessing and augmenting the images, implementing each of the six CNN architectures, training and validating the models, testing them on unseen data, and finally comparing their performance. This workflow is summarized in Fig. 1.

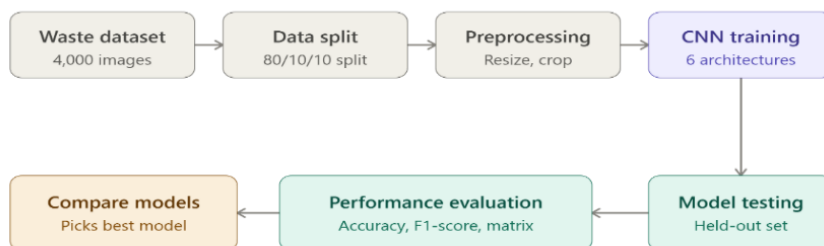


Fig. 1. Overall research workflow

Fig. 1 traces this workflow step by step. It begins with the waste dataset itself, which is then split into training, validation, and test sets. Every image in the resulting subsets is preprocessed through resizing and cropping, but only the training subset is further augmented, a step introduced specifically to expose the model to greater visual variability during learning. Each of the six CNN architectures is subsequently trained and validated on its own, though always under the identical protocol established earlier. For every architecture, the particular set of weights that yields the highest validation accuracy is the one carried forward for testing on the held-out set. Once tested, each architecture's results are analyzed using accuracy, precision, recall, F1-score, and the confusion matrix, and these metrics ultimately form the basis on which the six models are compared against one another and the best-performing architecture is identified.

2.2 Dataset and Data Split

The dataset consisted of 4,000 recyclable waste images that were evenly distributed across four classes: glass, cans, paper, and plastic. Each class contained 1,000 images. This balanced distribution was maintained to reduce the potential influence of class imbalance on model performance evaluation. Representative samples from each class are presented in Fig. 2 to illustrate the visual characteristics of the dataset used in the experiments.



Fig. 2. Sample images from the recyclable waste dataset

The dataset was divided using an 80:10:10 ratio, with 80% of the images allocated for training, 10% for validation, and 10% for testing. Accordingly, the training set contained 3,200 images, while the validation and test sets each contained 400 images. The same data split was applied to all architectures, ensuring that every model was trained, validated, and tested using identical samples.

Table 1. Distribution of the Recyclable Waste Dataset

Dataset Split	Glass	Cans	Paper	Plastic	Total
Training	800	800	800	800	3,200
Validation	100	100	100	100	400
Testing	100	100	100	100	400
Total	1,000	1,000	1,000	1,000	4,000

As shown in Table 1, each class contained the same number of samples across all subsets, with 800 images used for training, 100 images for validation, and 100 images for testing. This balanced distribution was maintained to reduce the influence of class imbalance on the evaluation results. Applying an identical data split across all architectures also ensured that each model was trained and tested under the same data conditions, allowing the observed performance differences to more accurately reflect the characteristics of each architecture.

2.3 Image Preprocessing and Augmentation

Before training, all images were resized according to the input requirements of each architecture. EfficientNetB0, ResNet50, MobileNetV2, MobileNetV3-L, and ShuffleNetV2 used an input size of 224×224 pixels, whereas InceptionV3 used an input size of 299×299 pixels.

Data augmentation was applied only to the training set to increase visual diversity and improve model generalization. The augmentation procedures included random rotations of up to $\pm 20^\circ$, horizontal and vertical translations of up to 10%, random resized cropping with a scale range of 0.8-1.0, and random horizontal flipping. The images were then converted into tensors and normalized using mean values of [0.485, 0.456, 0.406] and standard deviation values of [0.229, 0.224, 0.225].

No random augmentation was applied to the validation or test sets. Images in these subsets were only resized, converted into tensors, and normalized using the same parameters. This procedure was intended to ensure that the validation and test results reflected the models' ability to classify images without random transformations.

The preprocessing configuration was consistent with the implementation used in the experiments. The training set underwent resizing, rotation, translation, random resized cropping, horizontal flipping, and normalization, whereas the validation and test sets were subjected only to resizing and normalization.

2.4 CNN Architectures and Transfer Learning Strategy

This study evaluated six Convolutional Neural Network (CNN) architectures, namely EfficientNetB0, ResNet50, InceptionV3, MobileNetV2, MobileNetV3-L, and ShuffleNetV2. These architectures were selected because they represent different network design approaches. EfficientNetB0 employs compound scaling to balance network depth, width, and resolution. ResNet50 applies residual learning through skip connections to facilitate training in deeper networks, while InceptionV3 uses a multi-branch convolutional structure to extract features at multiple scales. MobileNetV2 employs inverted residuals and linear bottlenecks, MobileNetV3-L was designed with an emphasis on computational efficiency, and ShuffleNetV2 utilizes a channel shuffle mechanism to improve information exchange across channels with low computational cost. These architectural differences enable several CNN design principles to be evaluated under the same experimental conditions.

Table 2. CNN Architectures and Transfer Learning Strategy

No	Architecture	Input Size	Pretrained Weights	Backbone	Training Strategy
1	ResNet50	224×224	ImageNet-1K V2	Frozen	Feature Extraction
2	MobileNetV2	224×224	ImageNet-1K V2	Frozen	Feature Extraction
3	MobileNetV3-L	224×224	ImageNet-1K V2	Frozen	Feature Extraction
4	EfficientNetB0	224×224	ImageNet-1K V1	Frozen	Feature Extraction
5	ShuffleNetV2	224×224	ImageNet-1K V1	Frozen	Feature Extraction
6	InceptionV3	299×299	ImageNet-1K V1	Frozen	Feature Extraction

All models used pretrained ImageNet weights and employed a transfer learning strategy based on feature extraction. The parameters of the convolutional backbone were frozen during training, while the classification head remained trainable. This approach allowed previously learned visual features to be utilized for extracting relevant characteristics from waste images, while the training process focused on adapting the classifier to the four target classes: glass, cans, paper, and plastic. For EfficientNetB0, ResNet50,

MobileNetV2, MobileNetV3-L, and ShuffleNetV2, the classification head consisted of a fully connected layer with 128 neurons, a ReLU activation function, a dropout rate of 0.3, and a final fully connected layer with four output units. InceptionV3 used the same classifier structure for the main output while retaining its auxiliary classifier during training.

As shown in Table 2, all architectures used pretrained ImageNet weights with frozen backbones and employed a feature extraction strategy. Most models used an input size of 224×224 pixels, whereas InceptionV3 used an input size of 299×299 pixels in accordance with its architectural configuration. Differences in input size and pretrained weight versions reflect the specific characteristics of each architecture, while the training strategy was kept consistent to ensure that model performance could be compared under comparable experimental conditions.

2.5 CNN Benchmarking Experimental Framework

The experimental framework in this study was designed to evaluate the performance of six Convolutional Neural Network (CNN) architectures for recyclable waste classification under controlled and consistent experimental conditions. Each input image underwent preprocessing before being processed by one of the CNN backbones to extract feature representations. The extracted features were then passed to the classification head to predict one of four recyclable waste categories: glass, cans, paper, and plastic.

The six architectures were evaluated independently and were not combined using an ensemble strategy. To ensure a fair comparison, all models used the same dataset, data split, preprocessing and augmentation procedures, transfer learning strategy, training configuration, and evaluation protocol. This controlled setup was intended to reduce the influence of experimental variations so that performance differences could be interpreted more directly in relation to the characteristics of each architecture.

The contribution of this study lies in establishing a controlled benchmarking framework rather than proposing a new CNN architecture. Model performance was evaluated using accuracy, precision, recall, and F1-score, while confusion matrices were used to analyze class-level prediction errors. Validation performance was monitored during training, and the weights corresponding to the highest validation accuracy were selected for final testing.

The benchmarking results were then used to assess the relative suitability of the evaluated architectures for recyclable waste classification. In addition to predictive performance, model complexity was considered through the analysis of total and trainable parameters. This framework therefore provides a systematic basis for comparing classification performance, error patterns, and model efficiency across different CNN architectures.

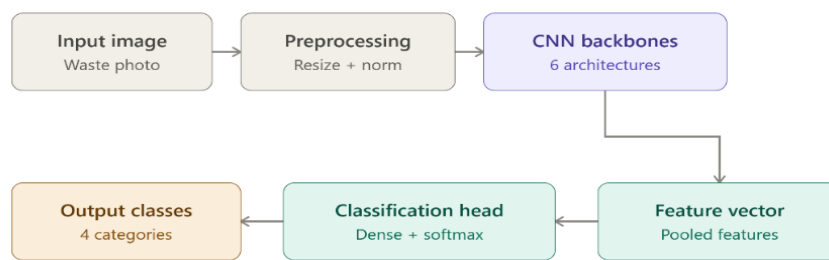


Fig. 3. CNN benchmarking experimental framework

After preprocessing, each input image was processed by one of the six CNN backbones to generate a feature vector or aggregated feature representation. The resulting representation was then passed to the classification head, which consisted of classification layers that produced the final output corresponding to one of four recyclable waste classes. As illustrated in Fig. 3, all architectures followed the same processing pipeline, ensuring that the evaluation was conducted within a consistent and controlled experimental framework.

2.6 Model Training Configuration

All models were implemented using the PyTorch and Torchvision frameworks. The training data were loaded using a DataLoader with shuffling enabled at each epoch, while the validation and test data were loaded without shuffling. CUDA was utilized when a GPU was available to accelerate the training process.

Each architecture was trained for 20 epochs with a batch size of 32. The Adam optimizer was used with a learning rate of 0.0001, and Cross-Entropy Loss was employed as the loss function. Because the parameters of the convolutional backbone were frozen, the optimization process updated only the parameters of the

classification head. These main training settings were applied consistently across all architectures to ensure that performance comparisons were conducted under comparable training conditions.

At each epoch, model performance was evaluated using the validation set. The model weights were saved whenever the validation accuracy exceeded the best value obtained in the previous epochs. After training was completed, the weights corresponding to the highest validation accuracy were reloaded and used for the final evaluation on the test set. This strategy ensured that the final evaluation was based on the best-performing model obtained during training rather than simply using the weights from the final epoch.

For InceptionV3, the auxiliary classifier was retained during training. The loss function consisted of the main classification loss and the auxiliary loss, with the auxiliary loss weighted by 0.4. The auxiliary classifier was used only during training, whereas validation and test evaluations relied solely on the output of the main classifier.

Table 3. Model Training Configuration

Parameter	Configuration
Number of classes	4
Epochs	20
Batch size	32
Optimizer	Adam
Learning rate	0.0001
Loss function	Cross-Entropy Loss
Hidden units	128
Activation	ReLU
Dropout	0.3
Dataset split	80:10:10
Evaluation metrics	Accuracy, Precision, Recall, F1-score
Metric averaging	Macro
Model selection	Best validation accuracy

As summarized in Table 3, all architectures used the same main training configuration, including the number of epochs, batch size, optimizer, learning rate, loss function, basic classification head structure, and model selection criterion. Maintaining these settings consistently across all models was intended to reduce the influence of differences in training procedures on the experimental results, allowing the performance comparison to focus more directly on the characteristics of each CNN architecture.

2.7 Model Performance Evaluation

The final evaluation was conducted using the test set, which consisted of 400 images, with 100 images assigned to each class: glass, cans, paper, and plastic. All models were evaluated using the same test set, ensuring that performance was assessed under identical evaluation conditions.

The performance of each architecture was measured using four metrics: accuracy, precision, recall, and F1-score. Accuracy was used as the primary metric for ranking the models because the objective of the study was to identify the architecture with the highest proportion of correct predictions. Precision, recall, and F1-score were used as complementary metrics to provide a more comprehensive assessment of classification performance. Accuracy is calculated as:

$$Accuracy = \frac{N_{correct}}{N_{total}} \quad (1)$$

Precision for class k is calculated as:

$$Precision_k = \frac{TP_k}{TP_k + FP_k} \quad (2)$$

Recall is calculated as:

$$Recall_k = \frac{TP_k}{TP_k + FN_k} \quad (3)$$

whereas the F1-score is calculated as:

$$F1_k = 2 \times \frac{Precision_k \times Recall_k}{Precision_k + Recall_k} \quad (4)$$

Precision, recall, and F1-score are reported using macro averaging, ensuring that each class contributes equally to the final metric values:

$$Metric_{macro} = \frac{1}{K} \sum_{k=1}^K Metric_k \quad (5)$$

where $K = 4$ represents the number of classes. In addition to the numerical metrics, the confusion matrix and normalized confusion matrix were used to analyze the distribution of correct predictions and classification error patterns for each class. Training accuracy, validation accuracy, training loss, and validation loss curves were also used to evaluate the convergence behavior and generalization capability of each model. The evaluation procedure computed accuracy as well as macro-averaged precision, recall, and F1-score. Confusion matrices, normalized confusion matrices, and accuracy and loss curves were also generated as part of the evaluation process. After all models had been evaluated, their accuracy, precision, recall, and F1-score values were compared. The architecture with the highest test accuracy was identified as the best-performing model in terms of classification performance, while precision, recall, F1-score, and the confusion matrix were used to support the interpretation of the classification behavior of each architecture.

3. RESULTS AND DISCUSSION

This section presents the evaluation results and analysis of the six CNN architectures tested under a consistent experimental protocol. The discussion includes performance comparisons based on accuracy, precision, recall, and F1-score, as well as analyses of the training process, generalization capability, and classification error patterns to identify the characteristics and performance of each architecture.

3.1 Performance Comparison of CNN Architectures

It should be emphasized that the results reported below are based on a single training run per architecture, consistent with the training configuration described in Section 2.6. As such, the reported values reflect a single point estimate rather than an average with associated variability, and the observed performance differences, while still informative, should be interpreted with this limitation in mind. The six CNN architectures were evaluated using four performance metrics: accuracy, precision, recall, and F1-score. All models were tested using the same dataset and evaluation procedure, allowing for a more objective comparison of the classification performance of each architecture. The evaluation results are presented in Table 4.

Table 4. Performance Comparison of Six CNN Architectures

Rank	CNN Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
1	ResNet50	86.00	86.10	86.00	85.64
2	MobileNetV2	83.50	83.99	83.50	83.08
3	MobileNetV3-L	83.50	84.23	83.50	82.84
4	EfficientNetB0	81.00	81.85	81.00	80.30
5	ShuffleNetV2	71.75	74.59	71.75	72.06
6	InceptionV3	71.75	72.58	71.75	71.02

As shown in Table 4, ResNet50 achieved the best overall performance, with an accuracy of 86.00%, precision of 86.10%, recall of 86.00%, and F1-score of 85.64%. MobileNetV2 and MobileNetV3-L achieved the same accuracy of 83.50%. However, MobileNetV2 obtained a slightly higher F1-score (83.08%) than MobileNetV3-L (82.84%), placing MobileNetV2 in second position. EfficientNetB0 achieved an accuracy of 81.00%, while ShuffleNetV2 and InceptionV3 each achieved an accuracy of 71.75%. Although both models obtained the same accuracy, ShuffleNetV2 produced higher precision and F1-score values than InceptionV3. Overall, these results indicate that the six architectures exhibited distinct performance characteristics despite being evaluated using the same dataset and experimental protocol. Under the experimental configuration used in this study, ResNet50 achieved the strongest overall classification performance among the six architectures evaluated across all four measured metrics.

3.2 Training Behavior, Generalization, and Classification Error Analysis

To provide a more comprehensive interpretation of the quantitative results presented in Table 4, additional analysis was conducted on the training behavior, generalization capability, and class-level prediction patterns of the six CNN architectures. Although the evaluation metrics summarize the overall classification performance of each model, they do not fully describe how the models learned during training or how prediction errors were distributed across the four waste classes.

Therefore, training and validation accuracy and loss curves were examined together with the confusion matrix and normalized confusion matrix. The learning curves were used to assess convergence behavior and the relationship between training and validation performance, while the confusion matrices were used to identify correct predictions and recurring misclassification patterns for each class. This combined analysis provides a more detailed understanding of the strengths and limitations of each architecture.

The analysis of the learning curves is particularly useful for examining whether improvements on the training set were accompanied by comparable performance on the validation set. Differences between training and validation behavior can provide an indication of how consistently each model generalizes beyond the data used for optimization. In this study, these curves were therefore used as supporting evidence when interpreting the final test performance of the six architectures, rather than relying only on the numerical metrics reported in Table 4.

In addition, class-level evaluation was considered important because similar overall accuracy values may still be associated with different prediction patterns across individual classes. The confusion matrix and normalized confusion matrix were therefore used to examine how each architecture distinguished among glass, cans, paper, and plastic, and to identify the classes that were more frequently confused. The corresponding training and classification results for all six architectures are presented in Fig. 4.

As shown in Fig. 4, all models exhibited increasing training accuracy and decreasing training loss throughout the training process. However, the validation performance patterns indicated differences in generalization capability across the evaluated architectures. ResNet50 showed relatively more stable and consistent validation performance, which was consistent with the test results in which it achieved the highest accuracy of 86.00%. In several of the other architectures, a larger gap between training and validation performance suggested a lower tendency to generalize to data that were not used during model optimization.

The confusion matrix analysis showed that the can class was generally recognized more consistently by most models, whereas classification errors occurred more frequently in the glass and plastic classes. In several architectures, plastic images tended to be misclassified as glass or cans, while glass images were also occasionally assigned to other classes. These patterns suggest that distinguishing the visual characteristics of glass and plastic was more challenging than distinguishing the other categories. The differences in error patterns across architectures further indicate that the feature representations produced by the respective backbones influenced how effectively each model distinguished among the four waste classes.

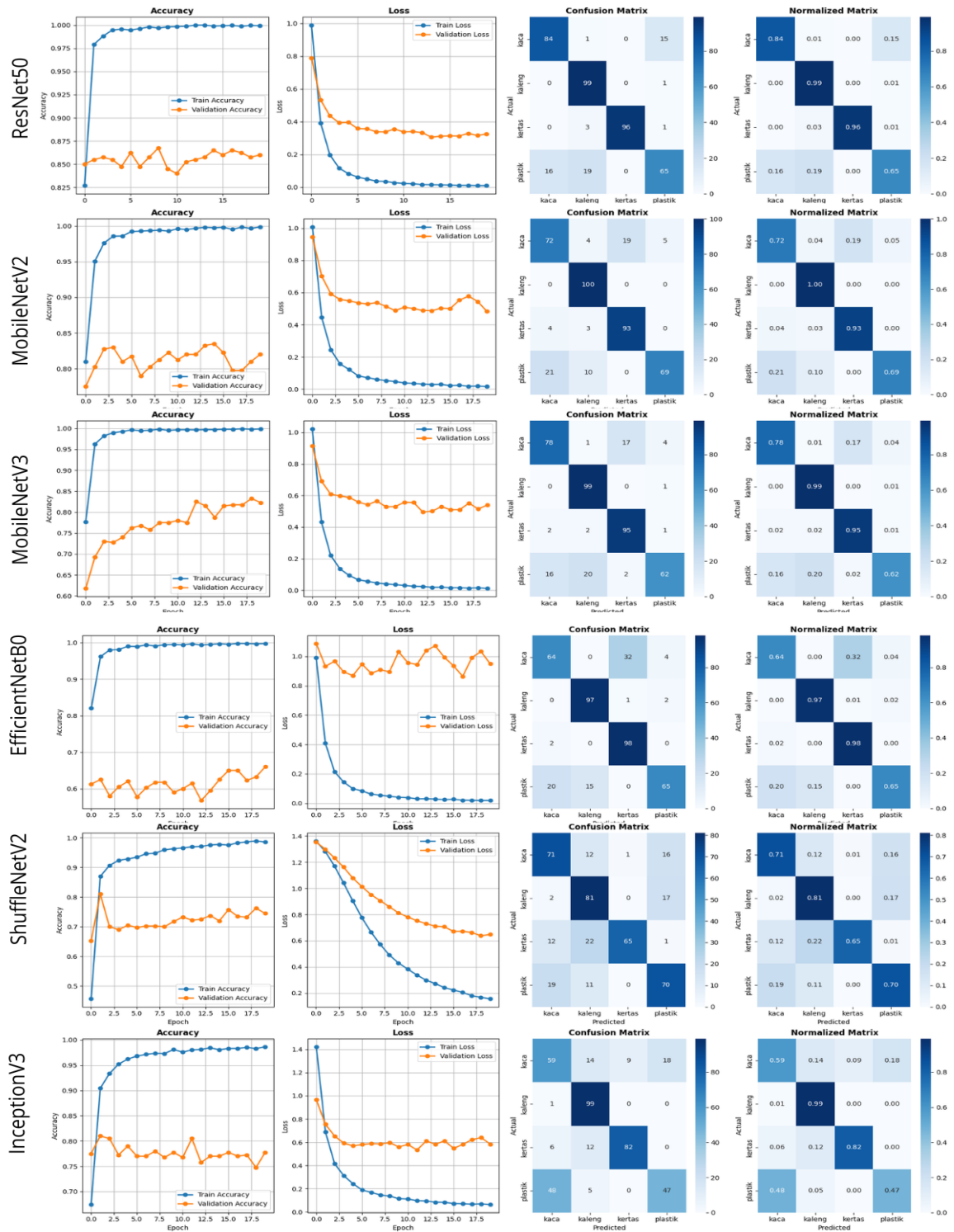


Fig. 4. Training and Evaluation Results of the Six CNN Architectures

3.3 Model Complexity and Practical Implications

In addition to classification performance, model complexity was also analyzed to assess the efficiency of each architecture. Model complexity was represented by the total number of parameters and the number of trainable parameters. In this study, most parameters in the backbone were frozen, while the optimization process focused on the classification head. As a result, the number of trainable parameters was substantially lower than the total number of parameters in each model. A comparison of model complexity and test accuracy is presented in Table 5.

Table 5. Model Complexity and Classification Performance

CNN Model	Total Parameters	Trainable Parameters	Accuracy (%)
ResNet50	23,770,820	262,788	86.00
MobileNetV2	2,388,356	164,484	83.50
MobileNetV3-L	3,095,476	123,524	83.50
EfficientNetB0	4,172,032	164,484	81.00
ShuffleNetV2	1,385,320	131,716	71.75
InceptionV3	24,705,000	361,736	71.75

As shown in Table 5, ResNet50 achieved the highest classification accuracy of 86.00%, with approximately 23.77 million total parameters. In comparison, MobileNetV2 and MobileNetV3-L each achieved an accuracy of 83.50% while requiring substantially fewer parameters, approximately 2.39 million and 3.10 million, respectively. These results indicate that a larger parameter count does not necessarily correspond to higher classification accuracy.

This observation is further supported by the performance of InceptionV3. Despite having the largest number of parameters among the evaluated architectures, approximately 24.71 million, InceptionV3 achieved an accuracy of 71.75%. In contrast, MobileNetV2 and MobileNetV3-L achieved considerably higher accuracy with substantially lower model complexity. This result highlights the trade-off between model capacity and classification performance and suggests that architecture selection should consider computational complexity in addition to predictive accuracy.

Within the experimental setting of this study, ResNet50 provided the strongest overall classification performance, whereas MobileNetV2 and MobileNetV3-L demonstrated greater parameter efficiency with a relatively limited reduction in accuracy. In particular, the lightweight architectures achieved competitive classification performance while using only a fraction of the parameters required by ResNet50. This characteristic may be advantageous for applications in which computational resources, memory usage, or deployment efficiency are important considerations.

Overall, the comparison demonstrates that model complexity alone is not a sufficient indicator of classification capability. Different architectural designs provide different balances between representational capacity, parameter efficiency, and predictive performance. Therefore, the selection of a CNN architecture for recyclable waste classification should consider not only the highest achievable accuracy but also the computational characteristics and practical requirements of the intended implementation.

3.4 Comparison with Previous Studies

Previous studies have applied Convolutional Neural Network architectures to waste classification using different research focuses and experimental configurations. A comparison of MobileNetV2 and EfficientNetB0 using transfer learning was conducted in [6], while MobileNetV2 and VGG16 were evaluated in [7] to examine differences in classification performance and computational efficiency. The study in [9] focused on the influence of preprocessing methods using MobileNetV1 and MobileNetV2, whereas [10] emphasized waste classification using a real-world dataset to investigate model generalization. Differences in research focus, datasets, number of classes, preprocessing procedures, training strategies, and evaluation protocols make direct performance comparisons across these studies inappropriate. Therefore, the comparison in this study focuses primarily on differences in experimental approaches and evaluation scope rather than on direct comparisons of numerical performance values.

As shown in Table 6, previous studies have adopted different research focuses and experimental scopes in CNN-based waste classification. The study in [6] compared MobileNetV2 and EfficientNetB0, while [7] evaluated MobileNetV2 and VGG16. Both studies focused on comparisons between a limited number of architectures. In contrast, [9] primarily investigated the influence of preprocessing methods on the performance of MobileNetV1 and MobileNetV2, whereas [10] addressed waste classification using a real-world dataset, with particular attention to dataset characteristics and model generalization. These differences in research focus indicate that previous studies evaluated architectures, preprocessing strategies, and data characteristics under experimental configurations that were not necessarily uniform, making direct performance comparisons across studies inappropriate.

Table 6. Comparison of Previous Studies with the Present Study

Reference	Model / Focus	Experimental Focus	Difference from the Present Study
[6]	MobileNetV2 and EfficientNetB0	Comparison of two CNN architectures using transfer learning	The evaluation was limited to two architectures, whereas the present study compares six architectures under a consistent experimental protocol
[7]	MobileNetV2 and VGG16	Comparison of the performance and efficiency of two architectures using transfer learning	The study compared two architectures, whereas the present study includes six architectures with different design characteristics and computational complexities
[9]	MobileNetV1 and MobileNetV2	Evaluation of the effect of preprocessing methods on waste classification	The study focused on preprocessing, whereas the present study applies consistent preprocessing and augmentation procedures to emphasize architectural differences
[10]	RealWaste dataset and waste classification under real-world conditions	Evaluation of classification performance and model generalization on real-world data	The study focused on real-world dataset characteristics, whereas the present study applies controlled benchmarking to six architectures using the same dataset and evaluation protocol
Present Study	EfficientNetB0, ResNet50, InceptionV3, MobileNetV2, MobileNetV3-L, and ShuffleNetV2	Controlled benchmarking of six CNN architectures using transfer learning	All models are compared using the same dataset, data split, preprocessing, augmentation, transfer learning strategy, training configuration, and evaluation procedure, together with performance, error-pattern, and model-complexity analyses

In contrast to these previous studies, the present study conducted controlled benchmarking of six CNN architectures: EfficientNetB0, ResNet50, InceptionV3, MobileNetV2, MobileNetV3-L, and ShuffleNetV2. These architectures were selected to represent different design characteristics, including compound scaling, residual learning, multi-scale feature extraction, inverted residuals, lightweight network design, and channel shuffling. All models were evaluated using the same dataset, data split, preprocessing and augmentation procedures, transfer learning strategy, training configuration, and evaluation protocol. This controlled experimental design was intended to reduce the influence of procedural variations and enable performance differences to be interpreted more consistently in relation to the characteristics of each architecture.

In addition to expanding the comparison to six architectures, the present study evaluated the models from multiple perspectives. Classification performance was assessed using accuracy, precision, recall, and F1-score, while confusion matrices were used to identify class-level error patterns. Model complexity was also examined in terms of total and trainable parameters. Therefore, the comparison was not limited to identifying the model with the highest classification performance but also provided information on misclassification patterns and the relationship between predictive performance and model complexity.

Under the same experimental conditions, ResNet50 achieved the highest accuracy of 86.00%, followed by MobileNetV2 and MobileNetV3-L, both of which achieved 83.50%. EfficientNetB0 achieved an accuracy of 81.00%, while ShuffleNetV2 and InceptionV3 each achieved 71.75%. These values were not used for direct numerical comparison with previous studies because of differences in datasets and experimental configurations. Instead, the results illustrate the relative performance characteristics of the six architectures when evaluated under the same experimental protocol.

4. CONCLUSION

This study evaluated six CNN architectures, namely ResNet50, MobileNetV2, MobileNetV3-L, EfficientNetB0, ShuffleNetV2, and InceptionV3, for the classification of four recyclable waste categories using a transfer learning strategy based on feature extraction. All models were evaluated using the same dataset, data split, preprocessing procedures, augmentation strategy, training configuration, and evaluation protocol, thereby enabling performance comparisons under consistent experimental conditions.

The experimental results showed that ResNet50 achieved the best overall performance, with an accuracy of 86.00%, precision of 86.10%, recall of 86.00%, and F1-score of 85.64%. MobileNetV2 and MobileNetV3-L each achieved an accuracy of 83.50% while using substantially fewer parameters than ResNet50. The confusion matrix analysis indicated that misclassifications occurred more frequently between the glass and plastic classes, whereas cans and paper were generally recognized more accurately by most models.

The model complexity analysis further demonstrated that a larger number of parameters does not necessarily result in better classification performance. ResNet50 achieved the highest accuracy, whereas MobileNetV2 and MobileNetV3-L provided a more favorable balance between classification performance and model complexity. These findings indicate that the selection of a CNN architecture should consider not only predictive accuracy but also model efficiency and the computational requirements of the intended application.

Future work may explore fine-tuning strategies, more diverse datasets, and real-time deployment of the classification models to further evaluate their generalization capability and computational efficiency under practical operating conditions.

REFERENCES

- [1] S. M. Cheema, A. Hannan, and I. M. Pires, "Smart Waste Management and Classification Systems Using Cutting Edge Approach," *Sustainability*, vol. 14, no. 16, p. 10226, 2022, doi: 10.3390/su141610226.
- [2] F. Fotovvatikhah, I. Ahmedy, R. M. Noor, and M. U. Munir, "A Systematic Review of AI-Based Techniques for Automated Waste Classification," *Sensors*, vol. 25, no. 10, p. 3181, 2025, doi: 10.3390/s25103181.
- [3] U. N. Oktaviana and Y. Azhar, "Garbage Classification Using Ensemble DenseNet169," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 6, pp. 1206–1215, 2021, doi: 10.29207/resti.v5i6.3673.
- [4] E. Oktayaessofa, C. A. Sari, E. H. Rachmawanto, and N. M. Yaacob, "Classification of Organic and Non-Organic Waste with CNN-MobileNet-V2," *Jurnal Teknik Informatika (JUTIF)*, vol. 5, no. 4, pp. 1173–1180, 2024, doi: 10.52436/1.jutif.2024.5.4.2165.
- [5] V. Kaya, "Classification of Waste Materials with a Smart Garbage System for Sustainable Development: A Novel Model," *Front. Environ. Sci.*, vol. 11, p. 1228732, 2023, doi: 10.3389/fenvs.2023.1228732.
- [6] S. E. Thio and J. Susilo, "Identifikasi Pemilahan Sampah Berbasis Algoritma Transfer Learning CNN Menggunakan MobileNetV2 dan EfficientNetB0," *bit-Tech*, vol. 8, no. 1, pp. 25–32, 2025, doi: 10.32877/bt.v8i1.1900.
- [7] R. A. Kumala, C. A. Sari, and E. H. Rachmawanto, "A Comparison of MobileNetV2 and VGG16 Architectures with Transfer Learning for Multi-Class Image-Based Waste Classification," *Journal of Applied Informatics and Computing*, vol. 9, no. 4, pp. 1610–1624, 2025, doi: 10.30871/jaic.v9i4.9958.
- [8] A. S. Ouedraogo, A. Kumar, and N. Wang, "Landfill Waste Segregation Using Transfer and Ensemble Machine Learning: A Convolutional Neural Network Approach," *Energies (Basel)*, vol. 16, no. 16, p. 5980, 2023, doi: 10.3390/en16165980.
- [9] A. Afifah, E. R. Arumi, M. Maimunah, and S. Nugroho, "Comparative Evaluation of Preprocessing Methods for MobileNetV1 and V2 in Waste Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 3, pp. 444–452, 2025, doi: 10.29207/resti.v9i3.6211.
- [10] S. Single, S. Iranmanesh, and R. Raad, "RealWaste: A Novel Real-Life Data Set for Landfill Waste Classification Using Deep Learning," *Information*, vol. 14, no. 12, p. 633, 2023, doi: 10.3390/info14120633.