Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3411~3423
e-ISSN: 2548-6861

Waste Image Classification Using EfficientNet B4 with MD5 and pHash Data Duplication Analysis on Two Datasets

Abdul Qohhar [1], Christy Atika Sari [2]*, Hidayah Rahmalan [3]
[1,2]Department of Informatics Engineering, Universitas Dian Nuswantoro, Semarang, Indonesia
[3]Department of Software Engineering, Universiti Teknikal Malaysia Melaka, Malacca, Malaysia
111202315111@mhs.dinus.ac.id [1],christy.atika.sari@dsn.dinus.ac.id [2], hidayah@utem.edu.my [3]

Article Info

Article history:

Received 2026-07-01
Revised 2026-07-24
Accepted 2026-08-08

Keyword:

Waste,
Image Classification,
EfficientNet-B4,
MD5,
pHash.

ABSTRACT

Waste sorting automation through deep learning is an important approach to support sustainable waste management systems. However, many studies still overlook dataset quality issues, especially exact and near-duplicate images that can cause information leakage and overly optimistic evaluation metrics. This study proposes a waste image classification pipeline that integrates an explicit data quality analysis stage using MD5 hashing for exact duplicate detection and perceptual hashing (pHash) for near-duplicate detection, followed by fine-tuned EfficientNet-B4 as the classification backbone. Experiments are conducted on two public datasets with distinct characteristics: Garbage Classification V2 (6 classes, 9,421 images) and RealWaste (9 classes, 4,749 images). With a Hamming distance threshold $\tau \leq 2$, the pHash cleansing identifies zero duplicates in Dataset 1 and only three near-duplicates (0.06%) out of 1,404,077 compared pairs in Dataset 2, confirming no evidence of image-duplication-based information leakage in either dataset. EfficientNet-B4 achieves 97.77% test accuracy with a macro F1-Score of 0.9768 on Dataset 1 and 93.26% accuracy with a macro F1-Score of 0.9379 on Dataset 2, demonstrating consistent performance across these two datasets with different numbers of classes, data volumes, and visual heterogeneity. These findings should be interpreted as evidence of robustness within the scope of the two evaluated datasets, rather than as a claim of generalization to unseen, external, or cross-domain waste image datasets.

I. INTRODUCTION

The increasing volume of household and municipal waste remains a major challenge for environmental management in many countries, including Indonesia, because it directly affects public health, economic activity, and overall environmental outcomes [1]. Various studies have shown that limited infrastructure capacity, low source-level waste separation, and community behavior that does not fully support sustainable waste management exacerbate waste handling problems in urban areas. These conditions increase the risk of water, soil, and air pollution and raise the financial burden on local governments, thereby creating a strong demand for more efficient, technology-driven approaches to improve the performance of waste management systems [1], [2]. At the operational level, waste sorting is still largely performed manually by workers at collection points and processing facilities, making the process highly dependent on individual accuracy and experience. Manual sorting is not only inefficient but also poses occupational safety risks and results in inconsistent classification due to worker fatigue and subjectivity. Consequently, automating the sorting process using information technology and artificial intelligence has become an important direction for achieving more reliable, consistent, and sustainable waste management systems. Consequently, automating the sorting process through information technology and artificial intelligence has become an important development direction to achieve more reliable, consistent, and sustainable waste management systems [2], [3].

Advances in computer vision and deep learning methods, particularly convolutional neural networks (CNNs), have

driven numerous studies that focus on waste image classification as a core component of automatic sorting systems [4], [5], [6]. Through a critical review of computer vision approaches for waste sorting, Lu and Chen reported that CNN-based models are among the most widely adopted techniques because they can automatically extract visual features and handle variations in object shape and texture [3]. In line with this, Islam et al. proposed an enhanced deep CNN-based framework for waste classification and reported high accuracy on several public datasets and real-world scenarios, reinforcing the potential of CNNs as the core solution for intelligent waste classification systems [2]. More recent reviews also emphasize the central role of deep learning models in intelligent waste sorting systems for supporting sustainable environments [7]. Beyond the waste sorting domain, higher-capacity variants such as EfficientNet-B4 have also demonstrated strong performance on other complex image classification tasks, for example, nine-class paddy leaf disease classification with test accuracy close to 97% and F1-Score around 0.94 [8].

A growing body of empirical studies indicates that the choice of CNN architecture and transfer learning strategy plays a crucial role in determining waste image classification performance. Angdresey et al. employed MobileNetV2 for multi-class waste classification and achieved 95.28% training accuracy and 89.48% validation accuracy on diverse waste images [9]. Other works have applied architectures such as ResNet and EfficientNet to household waste classification tasks and reported competitive performance across datasets collected in both controlled environments and real-world settings [5]-[7]. These studies typically rely on fine-tuning pretrained models, reusing rich feature representations while adapting them to the waste image domain [2], [3], [12]. In addition, several works focus on classifying organic versus inorganic waste or material-based categories, for example using CNNs and ResNet-50 to distinguish organic and inorganic waste and adopting EfficientNet for multi-class household waste classification [7]-[9]. Nevertheless, the high performance reported in many of these studies is closely tied to the characteristics of the datasets used [3], [13]. Differences in the number of classes, per-class sample distribution, background complexity, lighting conditions, and visual noise can significantly affect task difficulty and model generalization [3], [13]. Some studies rely on waste image datasets captured in real-world environments with complex backgrounds, overlapping objects, and large variations in viewpoint, while others use curated datasets collected in more controlled settings or from public repositories with relatively clean backgrounds [3], [11], [12], [15]. As a result, evaluation outcomes are not always directly comparable, and important questions remain regarding how well models trained on one dataset generalize to another with different characteristics [3], [13].

Beyond visual content variability, dataset quality is also affected by the presence of exact duplicate and near-duplicate images, which can bias training and evaluation processes [2], [16]. Undetected duplication can cause information leakage between training and test splits, leading to overly optimistic accuracy and other metrics that do not reflect true generalization capability [16]. The image hashing literature has introduced various perceptual hashing methods designed to produce compact image representations that are sensitive to visual content yet relatively robust to certain geometric and photometric transformations [16]. These methods are widely used for image retrieval, duplicate detection, and near-duplicate identification in large-scale repositories [16], [17]. Some studies exploit hashing techniques explicitly as a data quality improvement step by detecting and removing duplicate and near-duplicate images so that training datasets become more compact and evaluation metrics more trustworthy [18]. In line with this, recent work has also shown that uncontrolled image duplication in training data can reduce training efficiency and, under certain conditions, even degrade the generalization of deep neural network models [19]. S. Prathima et al. demonstrated that combining hashing and feature extraction can effectively capture and eliminate duplicate images in large-scale datasets, while Jakhar and Borah explored integrating perceptual hashing with deep learning to improve near-duplicate detection [17], [20]. However, to the best of the authors' knowledge, most existing waste image classification studies still do not explicitly incorporate dataset duplication analysis and cleansing into their data processing pipelines [2], [3], [12], [13].

Motivated by these gaps, this study addresses the problem of how to design a data quality analysis and cleansing process for waste image datasets using a combination of MD5 for exact duplicate detection and perceptual hash (pHash) for near-duplicate identification, and how this process affects the performance of an EfficientNet-B4-based waste classification model on two datasets with different characteristics [2], [4], [7], [11]-[13]. Specifically, the objectives of this study are: (1) to implement an automated duplication analysis and data cleansing pipeline on two waste image datasets, (2) to train and evaluate an EfficientNet-B4 model using transfer learning on each dataset before and after duplication cleansing, and (3) to perform a comparative performance evaluation using accuracy, precision, recall, and per-class F1-Score. While MD5 hashing, perceptual hashing, and EfficientNet-B4 are each individually well-established techniques, their explicit integration into a unified verification-and-classification pipeline for waste imagery has not been reported in prior studies. The novelty of this work is therefore not the individual components themselves, but the systematic combination of exact- and near-duplicate detection as a mandatory data quality gate prior to model training, coupled with a cross-dataset comparative evaluation that isolates the effect of dataset characteristics from model architecture choice.

II. METHOD

A. Research Workflow

This study was conducted through a series of systematic, chronologically ordered stages to ensure a clear and reproducible research process. In general, the workflow is divided into seven main stages. Figure 1 illustrates the overall workflow of the proposed batik motif classification framework. The process begins with dataset acquisition, consisting of two datasets with 6 and 9 classes, followed by data preprocessing and cleaning to eliminate exact duplicate images using MD5 hashing and detect near-duplicate images using pHash with a threshold of $\tau = 2$. The cleaned dataset is then divided into 70% training, 20% validation, and 10% testing using a stratified split to preserve class distribution. To improve model generalization, data augmentation techniques, including flip, rotation, and color jitter, are applied exclusively to the training set. The classification model is developed using EfficientNetB4 with ImageNet-based transfer learning and trained using the Adam optimizer, a learning rate of 0.0001, 100 training epochs, and Automatic Mixed Precision (AMP) for computational efficiency. Finally, the model performance is comprehensively evaluated using accuracy, precision, recall, F1-score, and the confusion matrix to assess its classification capability.

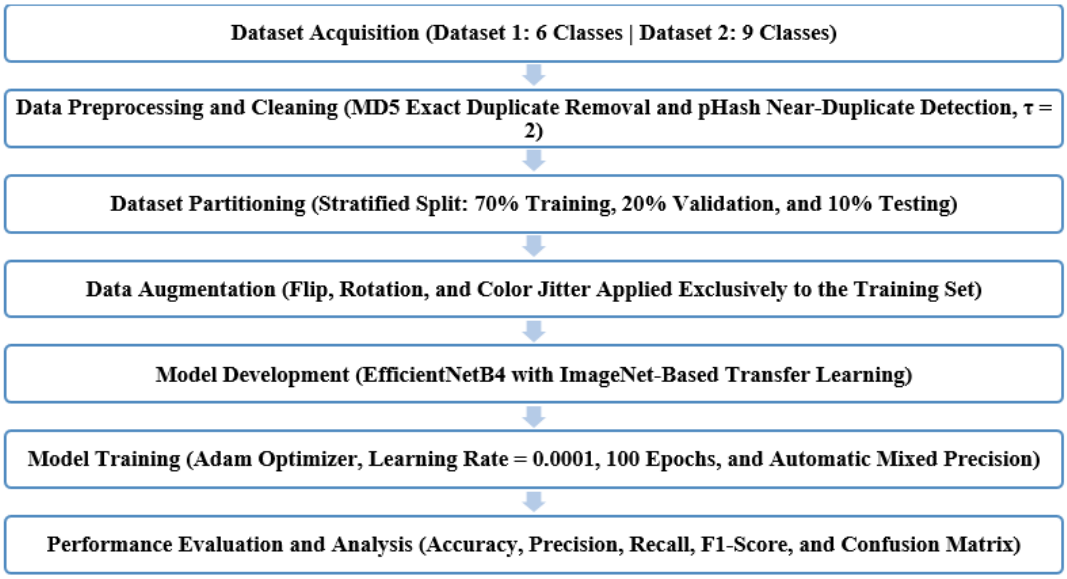

Figure 1. Proposed method

B. Experimental Environment

All experiments were carried out in a controlled computing environment with the specifications summarized in Table I. To ensure reproducibility, all sources of randomness (random seeds) were fixed to a constant value of 42, covering the Python random library, NumPy, PyTorch CPU, and PyTorch CUDA. In addition, deterministic mode was enabled for CUDA by disabling cudnn.benchmark so that identical runs produce identical results

TABLE I
COMPUTING ENVIRONMENT

Component	Specification
GPU	NVIDIA GeForce RTX 5050 Laptop GPU
Deep learning framework	PyTorch
Programming language	Python 3.10
Numerical precision	Automatic Mixed Precision
Random Seed	42 (for reproducibility)

C. Datasets

This study uses two different waste image for comparative experiments, as illustrated in Figure 2. The first dataset is the public Garbage Classification V2 dataset obtained from the Kaggle platform (https://www.kaggle.com/datasets/sumn2u/garbage-classification-v2/data). The original dataset contains 10 waste classes, but only 6 classes are used in this study, namely cardboard, clothes, glass, paper, plastic, and shoes, because each of these classes has more than 1,000 images.

This selection is intended to obtain a more adequate data distribution for model training and evaluation. Before data cleansing, the total number of images used from this dataset is 9,421, and the per-class distributions before and after cleansing are presented in Table II. The second dataset is the public RealWaste dataset obtained from the UCI Machine Learning Repository (https://archive.ics.uci.edu/dataset/908/realwaste). This dataset consists of waste images categorized into nine classes such as Cardboard, Food Organics, Glass, Metal,

Miscellaneous Trash, Paper, Plastic, Textile Trash, and Vegetation. The initial total number of images in this dataset is 4,752, and the per-class distributions before and after cleansing are shown in Table III.

Figure 2. Example images from the two waste datasets

TABLE II
CLASS DISTRIBUTION OF DATASET 1

Class	cardboard	clothes	glass	paper	plastic	shoes
Initial images	1.411	1.892	1.736	1.336	1.597	1.449
Total images	9.421					

TABLE III
CLASS DISTRIBUTION OF DATASET 2

Class	Cardboard	Food Organics	Glass	Metal	Miscellaneous Trash	Paper	Plastic	Textile Trash	Vegetation
Initial images	461	411	420	790	495	500	921	318	436
Total images	4.752								

D. Pre-processing and Data Cleansing

This stage aims to improve dataset quality by removing redundant images that could bias model training and evaluation [16], [17]. The cleansing process is carried out in three sequential steps: exact duplicate detection using MD5, near-duplicate detection using perceptual hashing (pHash), and summarization of the cleansing results. The first step uses the MD5 (Message Digest 5) algorithm to detect images that are pixel-wise identical to one another [16], [20].

TABLE IV
DATA CLEANSING RESULTS

Description	Dataset 1	Dataset 2
Initial images	9,421	4,752
MD5 drops (exact duplicates)	–	0
pHash drops (near-duplicates)	–	3
Clean images	9,421	4,749
Compared pairs (pHash)	~7,500,000+	1,404,077
Average Hamming distance	–	30.07

Each image is converted into a byte sequence and processed by the MD5 hash function to produce a 128-bit digest as shown in Equation (1). Two images are considered exact duplicates and one of them is removed when their MD5 hash values are identical within the same class, as formalized in Equation (2). The second step uses perceptual hashing (pHash) to detect images that are visually very similar even if they are not pixel-wise identical, for example due to differences in resolution, compression, or minor color changes [16], [17]. The pHash procedure converts each image to grayscale, resizes it to a small fixed resolution, and then applies the Discrete Cosine Transform (DCT), as expressed in Equation (3) [16]. The pHash value is obtained by comparing each low-frequency DCT coefficient with the mean value of these coefficients, as shown in Equation (4). The similarity between two images is then measured using the Hamming distance between their hash codes, as given in Equation (5).

Two images are considered near-duplicates and one of them is removed if they satisfy the condition in Equation (6) with a predefined Hamming distance threshold τ [16], [17], [20]. The threshold $\tau \leq 2$ was deliberately chosen to be conservative, prioritizing precision over recall in near-duplicate detection. Perceptual hashing is commonly used for duplicate or near-duplicate identification by comparing compact image signatures through Hamming distance, but threshold selection remains application-dependent and should be chosen carefully to avoid over-removal of visually distinct images. In this study, a strict threshold was preferred so that only images that were almost identical in global visual structure would be removed. This choice was also supported empirically by the observed Hamming-distance distributions: the minimum distance found among all compared pairs was 4 in Dataset 1 and 2 in Dataset 2, whereas the mean distances were 30.93 and 30.07, respectively, indicating that most image pairs were highly dissimilar. The third step summarizes the cleansing results for both datasets, as presented in Table IV. For Dataset 2, out of a total of 1,404,077 image pairs compared in the pHash stage, only three images are removed using the chosen threshold τ, indicating that the dataset exhibits a very high level of visual uniqueness.

$$H_{\mathrm{MD5}}(I) = \mathrm{MD5}(\mathrm{bytes}(I)) \quad (1)$$

$$H_{\mathrm{MD5}}(I_a) = H_{\mathrm{MD5}}(I_b), I_a \neq I_b \quad (2)$$

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) cos\left(\frac{(2x+1)u\pi}{2N}\right) cos \quad (3)$$

$$p_i = \begin{cases} 1 & if F_i \geq \bar{F} \\ 0 & if F_i < \bar{F} \end{cases} \quad (4)$$

$$d_H(P_a, P_b) = \sum_{i=1}^{n} \left(p_a^{(i)} \oplus p_b^{(i)}\right) \quad (5)$$

$$d_H(P_a, P_b) \leq \tau, \tau = 2 \quad (6)$$

E. Dataset Splitting

After the cleansing process, each dataset is divided into three subsets using a stratified splitting strategy to preserve the class proportion in every subset [21], [22]. Stratified splitting ensures that each class is represented proportionally in the training, validation, and test subsets, thereby reducing the risk of class imbalance affecting the evaluation results. The splitting ratios applied are formalized in Equation (7) and the resulting distributions for each dataset are summarized in Table V. Since the MD5- and pHash-based cleansing stage is completed before the dataset partitioning process, all detected exact duplicates and near-duplicate images are removed prior to subset assignment. Consequently, visually redundant images identified by the proposed MD5- and pHash-based screening pipeline cannot be assigned to different subsets, thereby minimizing the risk of image-duplication-based information leakage during model evaluation. Nevertheless, it should be noted that hash-based duplicate detection only addresses visually redundant samples. Other potential sources of information leakage, such as multiple images of the same physical object captured under substantially different viewpoints, illumination conditions, or acquisition settings that are not identified as near-duplicates, cannot be completely excluded and are therefore acknowledged as a limitation of the proposed approach.

$$D = D_{train} \cup D_{val} \cup D_{test}, |D_{train}| : |D_{val}| : |D_{test}| \quad (7)$$

TABLE V
DATASET SPLIT DISTRIBUTION

Subset	Ratio	Dataset 1	Dataset 2
Train	70%	6.595	3.324
Validation	20%	1.883	950
Test	10%	943	475
Total	100%	9.421	4.749

F. Data Augmentation

Data augmentation is applied exclusively to the training subset in order to artificially increase data diversity and reduce the risk of overfitting [23], [24]. Applying augmentation only to the training subset ensures that validation and test evaluations remain deterministic and unbiased.

TABLE VI
DATA AUGMENTATION CONFIGURATION

Augmentation technique	Parameter
Random horizontal flip	Probability 0.5
Random rotation	Range ±15°
Color jitter — brightness	Factor ±0.2
Color jitter — contrast	Factor ±0.2
Color jitter — saturation	Factor ±0.2
Color jitter — hue	Factor ±0.1
Resize	380 × 380 pixels

The augmentation techniques used are listed in Table VI. For the validation and test subsets, only resizing to 380×380 pixels is performed without any augmentation so that the evaluation results are fully reproducible. All images are subsequently normalized using ImageNet statistics as shown in Equation (8), where μ and σ denote the mean and standard deviation of the RGB channels computed on the ImageNet dataset.

$$I_{norm} = \frac{I - \mu_{ImageNet}}{\sigma_{ImageNet}} \quad (8)$$

G. EfficientNet-B4

The model used in this study is EfficientNet-B4, a CNN architecture from the EfficientNet family that has been widely adopted for various image classification tasks, including waste image classification and medical image classification [12], [25], [26], [27]. EfficientNet applies a compound scaling strategy to simultaneously balance the depth, width, and resolution of the network, as expressed in Equations (9) and (10), where d, w, and r represent the scaling factors for

depth, width, and resolution respectively; ϕ is the compound scaling coefficient; and α, β, γ are constants determined through grid search [12], [25], [27]. For EfficientNet-B4, $\phi = 4$ is used, yielding a model with a 380×380 pixel input resolution and approximately 19 million parameters. The EfficientNet-B4 model is initialized with pre-trained weights from the ImageNet dataset. This approach, known as transfer learning, enables the model to reuse low- and mid-level feature representations already learned from millions of ImageNet images, thereby reducing the amount of labeled training data required and accelerating convergence [12], [21], [25]. The final classifier layer is modified to match the number of target classes for each dataset (6 classes for Dataset 1 and 9 classes for Dataset 2), as expressed in Equation (11), where W_c is the weight matrix of the new classifier layer, C is the number of target classes, d_f is the feature dimension output by global average pooling, and b is the bias vector.

$$d = \alpha^{\phi}, w = \beta^{\phi}, r = \gamma^{\phi} \quad (9)$$

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2, \alpha \geq 1,\ \beta \geq 1,\ \gamma \geq 1 \quad (10)$$

$$\hat{y} = W_c \cdot \boldsymbol{h}_{pool} + b_c \quad (11)$$

H. Training Procedure

The model training process follows the hyperparameter configuration summarized in Table VII. The loss function used is Cross-Entropy Loss, which is the standard choice for multi-class classification tasks, as defined in Equation (12), where N is the number of samples in a batch, C is the number of classes, $y_{i,c}$ is the one-hot encoded label, and $p_{i,c}$ is the predicted probability for class c of sample i [2], [21]. The predicted probability $p_{i,c}$ is obtained by applying the Softmax activation function to the output layer logits, as shown in Equation (13), where $z_{i,c}$ denotes the raw (unnormalized) logit for class c of sample i. Model parameters are updated using the Adam (Adaptive Moment Estimation) optimizer, as described in Equations (14)–(17), where g_t is the gradient at iteration t, β_1 and β_2 are the moment decay coefficients, η is the learning rate, and ϵ is a small constant for numerical stability [28]. To improve computational efficiency, Automatic Mixed Precision (AMP) is used to combine single-precision (FP32) and half-precision (FP16) arithmetic (Islam et al., 2025). A gradient scaling mechanism is applied to prevent numerical underflow, as expressed in Equation (18), where s is a scaling factor that is dynamically updated by the GradScaler. The best model is selected based on the highest validation accuracy achieved after epoch 10, as defined in Equation (19).

TABLE VII
TRAINING HYPERPARAMETERS

Hyperparameter	**Value**
Epochs	100
Batch size	16
Learning rate	0.0001
Optimizer	Adam
Loss function	Cross-Entropy Loss
Minimum epoch for model selection	10

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\, log\ \square(\hat{p}_{ik}) \quad (12)$$

$$\hat{p}_{ik} = \frac{e^{z_{ik}}}{\sum_{j=1}^{K} e^{z_{ij}}} \quad (13)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (14)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (15)$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (16)$$

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad (17)$$

$$\tilde{g}_t = s \cdot g_t, \theta_t \leftarrow \theta_{t-1} - \eta \cdot \frac{\tilde{g}_t / s}{\sqrt{\hat{v}_t} + \epsilon}\hat{m}_t \quad (18)$$

$$\text{best\_model} = \arg\max_{\theta_e} \text{Acc}_{\text{val}}(e), e > 10 \quad (19)$$

I. Evaluation Metrics

The model performance is evaluated using several standard classification metrics. All metrics are computed from the values of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) obtained from the confusion matrix [21], [22]. Accuracy measures the proportion of correct predictions over all samples, as defined in Equation (20). Precision measures the proportion of correctly predicted positive samples among all predicted positives, as given in Equation (21). Recall measures the proportion of actual positive samples correctly identified by the model, as defined in Equation (22). The F1-Score is the harmonic mean of Precision and Recall, as shown in Equation (23). Macro averaging computes the unweighted mean of the metric values across all classes, giving equal importance to each class regardless of its frequency, as described in Equations (24)–(26) [2], [12]. In contrast, weighted averaging computes a frequency-weighted mean based on the number of samples (support) in each class, as formulated in Equation (27), where n_k denotes the number of samples in class k. The confusion matrix $C \in \mathbb{R}^{K \times K}$ defines element C_{ij} as the number of samples from true class i that are predicted as class j, as shown in Equation (28), with the diagonal elements C_{ii} representing correct predictions for class i. After training is completed, the final evaluation is performed on the test subset, which contains samples that are never seen by the model during either training or validation. The model used for this evaluation is the checkpoint that achieves the highest validation accuracy and is stored during training according to the selection criterion in Equation (19). The evaluation is carried out separately for each dataset: Dataset 1 uses 943 test samples, while Dataset 2 uses 475 test samples. The comparative analysis of evaluation results across the two datasets is presented and discussed in detail in Section III.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

$$Precision = \frac{TP}{TP + FP} \quad (21)$$

$$Recall = \frac{TP}{TP + FN} \quad (22)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (23)$$

$$Macro\ Precision = \frac{1}{K}\sum_{k=1}^{K} Precision_k \quad (24)$$

$$Macro\ Recall = \frac{1}{K}\sum_{k=1}^{K} Recall_k \quad (25)$$

$$Macro\ F_1 = \frac{1}{K}\sum_{k=1}^{K} F_{1,k} \quad (26)$$

$$\text{Weighted } F_1 = \frac{\sum_{k=1}^{K} n_k \cdot F_{1,k}}{\sum_{k=1}^{K} n_k} \quad (27)$$

C_{ij} = the number of samples of class " i" predicted as class " j" (28)

III. Results and Discussion

A. Training Results

The training process was carried out for 100 epochs on both datasets independently using an identical configuration: Adam optimizer with learning rate $lr = 0.0001$, Cross-Entropy Loss, batch size of 16, input resolution of 380×380 pixels, and mixed-precision training (AMP). Reproducibility was ensured by fixing the random seed to 42 across all libraries, including Python random, NumPy, PyTorch CPU, and PyTorch CUDA. For Dataset 1 (Garbage Classification V2, 6 classes, 9,421 clean images), the best model checkpoint was obtained at epoch 52, with a validation accuracy of 98.04% and a training accuracy of 99.88%. For Dataset 2 (RealWaste, 9 classes), two scenarios were considered: in the cleaning scenario (4,749 images after eliminating 3 pHash-based near-duplicates), the best model was obtained at epoch 74 with a validation accuracy of 95.89%; in the non-cleaning scenario (4,752 raw images), the best model was obtained at epoch 90 with a validation accuracy of 96.11%. Table VIII summarizes the best training results for both datasets. In the cleaning scenario of Dataset 2, the model required more epochs to converge compared to Dataset 1 (74 vs. 52 epochs), which can be attributed to the larger decision space (9 vs. 6 classes) and the smaller number of training samples (3,324 vs. 6,595 samples). The non-cleaning scenario required even more epochs (epoch 90), which is likely due to the absence of an explicit data quality selection stage, forcing the model to undergo more iterations before reaching a stable feature representation.

TABLE VIII
SUMMARY OF BEST TRAINING RESULTS

Metric	D1 (clean)	D2 (clean)	D2 (raw)
Best epoch	52	74	90
Train loss	0.0036	0.0061	0.0030
Train accuracy	99.88%	99.79%	99.91%
Val. accuracy	98.04%	95.89%	96.11%

Figure 3 shows the training and validation loss curves for EfficientNet-B4 over 100 epochs in three scenarios: Dataset 1 (original), Dataset 2 without cleansing (raw), and Dataset 2 after MD5- and pHash-based cleansing (clean). These trajectories illustrate the convergence behavior of the model and provide insight into potential overfitting. For Dataset 1, both training and validation loss decrease smoothly and stabilize shortly after epoch 20, reaching a final validation loss of 0.1302 with a narrow train-validation gap of 0.1286, indicating strong convergence and good generalization. In contrast, Dataset 2 exhibits a substantially larger train-validation gap in both scenarios, reflecting greater classification difficulty. In the raw Dataset 2 scenario, the training loss approaches near-zero (0.0030), whereas the validation loss remains considerably higher (0.2737), resulting in a gap of approximately 0.2707 and indicating stronger overfitting tendencies than in Dataset 1. After MD5- and pHash-based cleansing, the validation loss decreases slightly to 0.2605 and the train-validation gap narrows to 0.2546. Because only three near-duplicate images were detected in Dataset 2, this improvement should be interpreted cautiously and not attributed solely to duplicate removal. Overall, the observed convergence behavior appears to be influenced primarily by intrinsic dataset characteristics, including the larger number of classes, smaller training set size, and greater visual heterogeneity, rather than by duplicate removal alone.

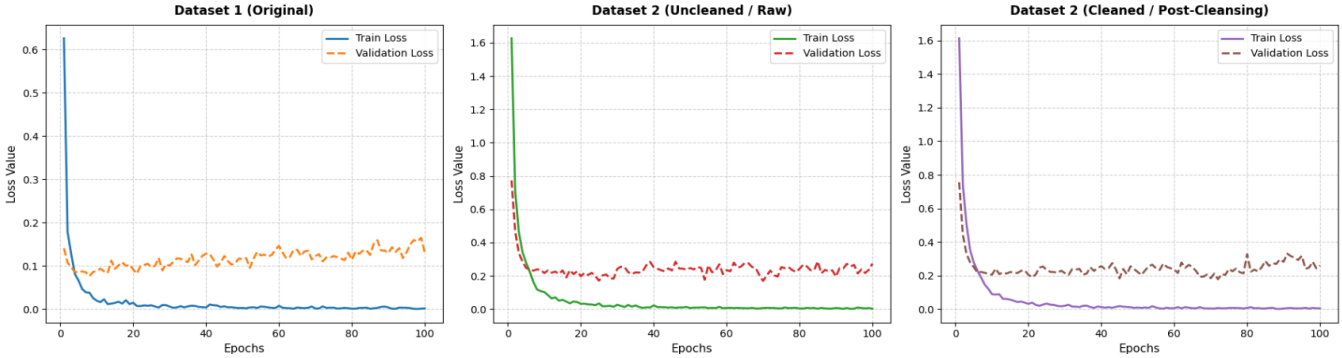

Figure 3. Comparison of Training and Validation Loss Curves

B. *Evaluation on Dataset 1 — Garbage Classification V2*

The best model checkpoint for Dataset 1 (epoch 52) is evaluated on 943 test images that were never used during training or model selection. The complete classification report is presented in Table IX. The model achieves a test accuracy of 97.77%, with a macro F1-Score of 0.9768 and a weighted F1-Score of 0.9777. The very small difference between macro and weighted F1 (less than 0.001) indicates that performance is well balanced across classes, with no substantial bias toward any particular class. The Clothes class attains the highest F1-Score (0.9973) with perfect precision (1.0000), followed by the Shoes class (F1 = 0.9966) with perfect recall (1.0000). Both classes exhibit textures and shapes that are clearly distinct from inorganic materials such as glass and plastic, which makes them easier for the model to distinguish. The Paper class records the lowest F1-Score (0.9591), mainly due to its visual similarity to Cardboard (white or brownish color and fibrous surface), which leads to occasional misclassification between these two classes. The confusion matrix for Dataset 1 in the cleaning scenario is shown in Figure 4.

TABLE IX
CLASSIFICATION REPORT ON DATASET 1 (N = 943)

Class	Precision	Recall	F1-Score	Support
Cardboard	0.9783	0.9574	0.9677	141
Clothes	1.0000	0.9947	0.9973	189
Glass	0.9770	0.9770	0.9770	174
Paper	0.9556	0.9627	0.9591	134
Plastic	0.9568	0.9688	0.9627	160
Shoes	0.9932	1.0000	0.9966	145
Accuracy	—	—	0.9777	943
Macro avg	0.9768	0.9768	0.9768	943
Weighted avg	0.9778	0.9777	0.9777	943

The confusion matrix shown in Figure 4 further confirms this observation. Most misclassifications are concentrated between the Paper and Cardboard classes, whereas the remaining classes exhibit very few classification errors. This pattern suggests that the residual errors are mainly caused by inter-class visual similarity rather than insufficient feature representation learned by the EfficientNet-B4 model. Overall, the results demonstrate that the proposed model provides robust and well-balanced classification performance on Dataset 1 while remaining challenged by categories with highly similar visual characteristics.

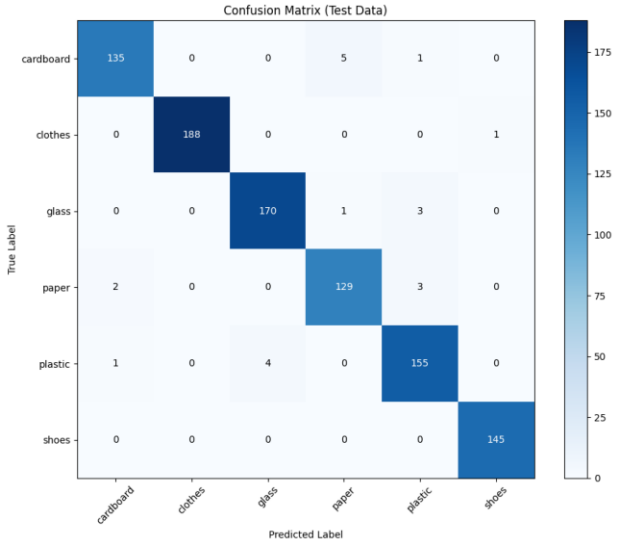

Figure 4. Confusion matrix of model predictions on the Dataset 1 test set

C. *Evaluation on Dataset 2 — RealWaste (with Data Cleansing)*

For Dataset 2, the best model in the cleaning scenario (epoch 74) is evaluated on 475 test images. The full classification report is provided in Table X. The model achieves a test accuracy of 93.26% with a macro F1-Score of 0.9379 on the 475 test images. Compared to Dataset 1, this represents a decrease of 4.51 percentage points in accuracy, which can be attributed to three structural factors: a larger number of classes (9 vs. 6), fewer training samples (3,324 vs. 6,595), and the presence of the Miscellaneous Trash category, which is visually heterogeneous by definition. The Vegetation class attains perfect precision (1.0000) because green organic colors and leaf textures are highly distinctive. The Food Organics class achieves perfect recall (1.0000) but lower precision (0.9318), as some brownish Cardboard/Paper images are misclassified into this class. The Miscellaneous Trash class yields the lowest F1-Score (0.8776) because it lacks consistent visual patterns, while the Plastic class (F1 = 0.9011) is also relatively challenging due to its appearance sometimes resembling Metal. The confusion matrix for Dataset 2 in the cleaning scenario is shown in Figure 5.

The confusion matrix indicates that the most challenging classes in Dataset 2 are Miscellaneous Trash, Plastic, Metal, and Food Organics, all of which exhibit overlapping visual characteristics in real-world scenes. Among these, Miscellaneous Trash is particularly difficult to classify because it encompasses a heterogeneous collection of objects rather than a single material category, resulting in highly variable visual features and the lowest F1-score. In addition, similarities in color, texture, and reflective surfaces contribute to confusion between Plastic and Metal, while the visual appearance of Food Organics occasionally overlaps with brown-colored Cardboard and Paper samples. Furthermore, variations in background complexity, illumination, object deformation, and partial occlusion make the decision

boundaries between these classes less distinct than those observed in Dataset 1. These findings suggest that the remaining classification errors are primarily driven by intrinsic dataset complexity and inter-class visual similarity rather than limitations of the proposed EfficientNet-B4 model.

\

TABLE X
CLASSIFICATION REPORT ON DATASET 2 (N = 475)

Class	Precision	Recall	F1-Score	Support
Cardboard	0.9565	0.9565	0.9565	46
Food Organics	0.9318	1.0000	0.9647	41
Glass	0.9524	0.9524	0.9524	42
Metal	0.9036	0.9494	0.9259	79
Miscellaneous Trash	0.8958	0.8600	0.8776	50
Paper	0.9592	0.9400	0.9495	50
Plastic	0.9111	0.8913	0.9011	92
Textile Trash	0.9375	0.9375	0.9375	32
Vegetation	1.0000	0.9535	0.9762	43
Accuracy	—	—	0.9326	475
Macro avg	0.9387	0.9378	0.9379	475
Weighted avg	0.9330	0.9326	0.9325	475

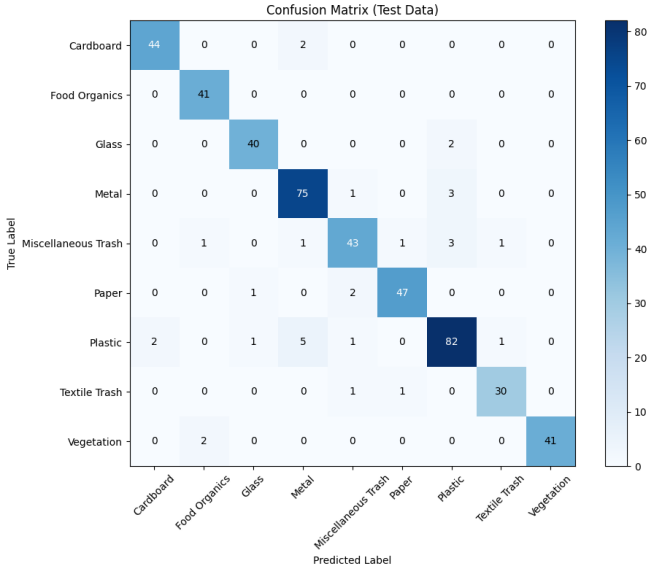

Figure 5. Confusion Matrix of Dataset 2

D. Evaluation on Dataset 2 — RealWaste (without Data Cleansing)

In the non-cleaning scenario, all 4,752 RealWaste images are used without MD5 or pHash-based elimination. The best model is obtained at epoch 90 with a validation accuracy of 96.11%. Final evaluation is performed on 476 test images (10% stratified split, seed = 42), and the classification report is summarized in Table XI. In this scenario, the model reaches a test accuracy of 94.96% with a macro F1-Score of 0.9561 on 476 test images. The Paper and Cardboard classes achieve perfect precision (1.0000), and the Paper and Textile Trash classes achieve perfect recall (1.0000). The Miscellaneous Trash class remains the lowest performing class (F1 = 0.9000), consistent with the cleaning scenario, which confirms that its lower performance is inherent to its heterogeneous visual characteristics rather than an artifact of the cleansing process.The confusion matrix for the non-cleaning scenario is shown in Figure 6.

The confusion matrix for the non-cleansing scenario confirms a similar error structure, where Miscellaneous Trash remains the most unstable category and confusion persists among material classes with similar appearance. This consistency suggests that the main classification difficulty in Dataset 2 arises from intrinsic dataset complexity rather than from the extremely low number of detected near-duplicate images (three images) alone.

TABLE XI
CLASSIFICATION REPORT ON DATASET 2 WITHOUT CLEANING (N = 476)

Class	Precision	Recall	F1-Score	Support
Cardboard	1.0000	0.9565	0.9778	46
Food Organics	0.9524	0.9756	0.9639	41
Glass	0.9318	0.9762	0.9535	42
Metal	0.9250	0.9367	0.9308	79
Miscellaneous Trash	0.9000	0.9000	0.9000	50
Paper	1.0000	1.0000	1.0000	50
Plastic	0.9231	0.9130	0.9180	92

Textile Trash	0.9697	1.0000	0.9846	32
Vegetation	1.0000	0.9545	0.9767	44
Accuracy	—	—	0.9496	476
Macro avg	0.9558	0.9570	0.9561	476
Weighted avg	0.9500	0.9496	0.9496	476

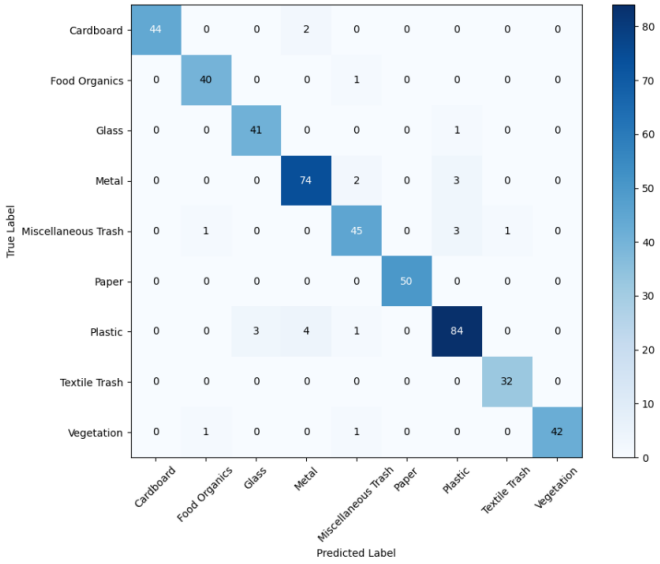

Figure 6. Confusion matrix of model predictions on the Dataset 2 test set (without cleaning)

E. Ablation Study

To address the research question regarding the impact of MD5 and pHash-based data cleansing on model performance, a direct comparison is conducted between the two Dataset 2 scenarios under identical training configurations. The test-set results are summarized in Table XII. The non-cleaning scenario yields a higher test accuracy (+1.70 percentage points) than the cleaning scenario. However, this difference must be interpreted cautiously due to several confounding factors: (1) the test subsets differ slightly in size and composition (476 vs. 475 samples) as a result of the three removed images, making the comparison not strictly "apples-to-apples"; (2) the best models are selected at different epochs (90 vs. 74), implying longer training for the non-cleaned case; and (3) experiments are conducted with a single random seed, so variance estimates are not available. From the perspective of the cleansing objective, the key finding is not the small difference in accuracy but the confirmation that only three near-duplicate images (0.06%) are detected among 1,404,077 compared pairs in Dataset 2. This extremely low duplication rate indicates that Dataset 2 has a high degree of visual uniqueness and does not suffer from information leakage between subsets that could artificially inflate evaluation metrics. This observation is consistent with prior work reporting that pHash-detectable duplication typically arises in web-crawled datasets with resized or recompressed images, rather than in independently collected real-world datasets such as RealWaste [16]. This near-zero duplication rate should not be interpreted as evidence that the cleansing stage is unnecessary. Rather, it demonstrates the value of the pipeline as a verification mechanism: without explicitly running MD5 and pHash checks, this level of dataset integrity would remain an unverified assumption rather than a quantified fact. In datasets curated from web-crawled or aggregated sources, where duplication rates are typically much higher, the same pipeline would be expected to yield a more substantial impact on reported metrics.

As shown in Table XIII, the computational efficiency of the proposed cleansing pipeline was evaluated separately on both datasets. Dataset 1 required 279.13 s to process 9,421 images and 7,504,883 pHash comparison pairs, whereas Dataset 2 required 110.34 s to process 4,752 images and 1,404,077 comparison pairs. In both cases, the pure MD5 computation was substantially faster than the pure pHash computation, confirming the effectiveness of using MD5 as a lightweight exact-duplicate filtering stage before the more computationally intensive perceptual hashing process. Although the computational cost of the pipeline increases with the number of image pairs, the results demonstrate that MD5 effectively reduces unnecessary perceptual comparisons while maintaining the ability to identify exact duplicates efficiently. These findings are consistent with the image-hashing literature, which positions cryptographic hashing for exact identity verification and perceptual hashing for similarity-aware duplicate detection.

TABLE XII
ABLATION RESULTS: WITH VS. WITHOUT DATA CLEANSING (DATASET 2)

Metric	With cleansing (4,749)	Without cleansing (4,752)	Δ
Total images	4,749	4,752	3
Test images (N)	475	476	1
Best epoch	74	90	16
Best val. accuracy	95.89%	96.11%	+0.22pp
Test accuracy	93.26%	94.96%	+1.70pp
Macro avg F1	0.9379	0.9561	+0.0182
Weighted avg F1	0.9325	0.9496	+0.0171

TABLE XIII
COMPUTATIONAL EFFICIENCY OF THE MD5+PHASH CLEANSING PIPELINE

Dataset	Total images	pHash pairs compared	MD5 pure time (s)	pHash pure time (s)	Hamming comparison (s)	Total wall-clock (s)
Dataset 1	9,421	7,504,883	2.68	41.87	129.27	279.13
Dataset 2	4,752	1,404,077	3.57	19.17	23.93	110.34

F. Comparison with Previous Studies

A comparison is made with previous works that employ CNN architectures with transfer learning for waste image classification. The results are summarized in Table XIV. The result on Dataset 1 (97.77%) surpasses all comparison methods. Relative to Kurniawan et al., who used EfficientNet-B0 (88.50%), there is a 9.27-point improvement that is directly related to the higher representational capacity of EfficientNet-B4 (~19 million parameters, 380×380 input resolution) compared with B0 (~5.3 million parameters, 224×224 input), under the same compound scaling framework [12], [27]. Angdresey et al., published in the same journal, reported a validation accuracy of 89.48% for MobileNetV2, whereas in this work EfficientNet-B4 reaches 98.04% validation accuracy on Dataset [9]. On Dataset 2, the achieved accuracies (93.26% and 94.96%) are lower in absolute terms than some comparison results, but the comparison is not fully equivalent because RealWaste is collected in real-world environments with high variation in background and lighting, making it inherently more challenging than datasets acquired in controlled settings.

TABLE XIV
COMPARISON WITH PREVIOUS CNN-BASED WASTE CLASSIFICATION STUDIES

Study	Architecture	Dataset	Accuracy
Kurniawan et al.[12]	EfficientNet-B0	TrashNet	88.50%
Angdresey et al. [9]	MobileNetV2	Multi-kelas	89.48%
Mohammed et al. [13]	ANN+Feature Fusion	Custom Waste	90.00%
Islam et al. [2]	DenseNet201+SE	Multi-dataset	94.67%
Ahmad et al. [11]	EfficientNetV2 (mod)	Urban Waste	95.40%
This work — D1	EfficientNet-B4	GC-V2 (6 kls)	97.77%
This work — D2 (clean)	EfficientNet-B4	RealWaste (9 kls)	93.26%
This work — D2 (raw)	EfficientNet-B4	RealWaste (9 kls)	94.96%

G. Strengths and Limitations

Three main factors contribute to the strong performance of the proposed model. First, EfficientNet-B4 with 380 × 380 input resolution allows the extraction of richer material texture features than smaller variants such as B0 or lightweight architectures like MobileNetV2. Second, initialization with ImageNet pre-trained weights provides strong hierarchical feature representations, enabling fine-tuning to focus on adapting to the waste image domain with relatively limited data. Third, stratified splitting with a deterministic seed ensures representative and reproducible class distributions in each subset.

The main limitations of this study are as follows. (1) The quantitative impact of data cleansing on Dataset 2 cannot be measured definitively because the detected duplication rate is extremely low (0.06%); consequently, the observed performance differences between the cleaning and non-cleaning scenarios are also influenced by changes in test-set composition and differences in the selected training epochs. (2) All experiments were conducted using a single random seed; therefore, variance estimates, confidence intervals, and statistical significance tests of the reported performance metrics were not performed. (3) The relatively small test set of Dataset 2, with several classes containing fewer than 50 samples, increases the uncertainty of per-class F1-score estimates. (4) The pHash method employed in this study primarily captures global structural similarity through low-frequency DCT coefficients and is therefore less robust to substantial viewpoint changes, perspective distortion, object rotation, or significant illumination variations. Consequently, images of the same physical object captured under markedly different acquisition conditions may not be identified as near-duplicates by the adopted threshold-based approach.

Future work should investigate datasets with substantially higher duplication rates to better quantify the practical impact of data cleansing through additional ablation studies. Furthermore, repeated experiments using multiple random seeds should be performed to evaluate the statistical reliability of the reported performance and enable formal statistical significance analysis. The computational timing reported in Table XIII should also be interpreted as end-to-end pipeline execution time because it includes image I/O, hash computation, and pairwise comparison overhead in addition to the pure hashing operations.

From a practical deployment perspective, several factors relevant to real-world automated sorting systems remain to be addressed. Waste sorting facilities typically involve variable

lighting conditions, overlapping or partially occluded objects on conveyor belts, and camera angles that differ substantially from the relatively clean, single-object images used in this study's datasets. Additionally, deployment on edge devices with limited computational resources may require model compression or the use of lighter EfficientNet variants (e.g., B0–B2) to meet real-time inference constraints. These considerations are left for future work involving field-collected data and hardware-in-the-loop testing.

IV. CONCLUSION

This study proposes a garbage image classification pipeline that integrates data quality analysis and duplication cleaning based on MD5 and perceptual hash (pHash) before training EfficientNet-B4 on two public datasets. The analysis shows that Garbage Classification V2 is practically duplication-free, while RealWaste contains only three near-duplicates (0.06% of 1,404,077 compared pairs), so both datasets show no evidence of significant image-duplication-based information leakage between subsets. With this pipeline, EfficientNet-B4 achieves a test accuracy of 97.77% with a macro F1-score of 0.9768 on Dataset 1 and 93.26% with a macro F1-score of 0.9379 on Dataset 2 after cleaning, indicating consistent and reliable performance across two datasets with different numbers of classes, training data sizes, and levels of visual heterogeneity. This consistency should be interpreted as evidence of robustness within the scope of the two datasets evaluated, rather than as a claim of generalization to unseen, external, or cross-domain waste image datasets.

The no-cleansing scenario on Dataset 2 yielded an accuracy of 94.96% and a macro F1-score of 0.9561; however, this difference was influenced by test set composition, best epoch, and the single-seed setting, so on datasets with very low duplication, cleansing served primarily as a data quality verification step rather than a primary driver of metric improvement. Compared with several other CNN approaches, the results on Dataset 1 were competitive, while the results on Dataset 2 remained strong for a more challenging real-world dataset. Overall, the findings demonstrate that integrating systematic dataset quality verification with EfficientNet-B4 provides a reliable and reproducible pipeline for waste image classification under the evaluated experimental settings.

The main contributions of this study are: (1) the explicit integration of MD5- and pHash-based data quality analysis and duplication cleaning into the garbage classification pipeline; (2) demonstrating the effectiveness of EfficientNet-B4 for multi-class garbage image classification on two datasets with different characteristics; and (3) a comparative analysis highlighting the influence of the number of classes, training data size, and visual heterogeneity on model performance and generalization. For further research, it is recommended to evaluate the impact of cleansing on datasets with higher levels of duplication, replicate the experiments with multiple seeds, and test the model on more diverse real-world scenarios and field-collected data.

REFERENCES

[1] Z. Zhang *et al.*, "Municipal solid waste management challenges in developing regions: A comprehensive review and future perspectives for Asia and Africa," *Science of The Total Environment*, vol. 930, p. 172794, Jun. 2024, doi: 10.1016/j.scitotenv.2024.172794.

[2] Md. M. Islam, S. M. M. Hasan, Md. R. Hossain, Md. P. Uddin, and Md. Al Mamun, "Towards sustainable solutions: Effective waste classification framework via enhanced deep convolutional neural networks," *PLoS One*, vol. 20, no. 6, p. e0324294, Jun. 2025, doi: 10.1371/journal.pone.0324294.

[3] W. Lu and J. Chen, "Computer vision for solid waste sorting: A critical review of academic research," *Waste Management*, vol. 142, pp. 29–43, Apr. 2022, doi: 10.1016/j.wasman.2022.02.009.

[4] R. A. Kumala, C. A. Sari, and 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 (JAIC)*, vol. 9, no. 4, pp. 1610–1624, 2025, doi: https://doi.org/10.30871/jaic.v9i4.9958.

[5] 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, Jul. 2024, doi: 10.52436/1.jutif.2024.5.4.2165.

[6] M. Zulhusni, C. A. Sari, and E. H. Rachmawanto, "Implementation of DenseNet121 Architecture for Waste Type Classification," *Advance Sustainable Science, Engineering and Technology*, vol. 6, no. 3, May 2024, doi: 10.26877/asset.v6i3.673.

[7] U. Kumar Lilhore, S. Simaiya, S. Dalal, M. Radulescu, and D. Balsalobre-Lorente, "Intelligent waste sorting for sustainable environment: A hybrid deep learning and transfer learning model," *Gondwana Research*, vol. 146, pp. 252–266, Oct. 2025, doi: 10.1016/j.gr.2024.07.014.

[8] J. Padhi, L. Korada, A. Dash, P. K. Sethy, S. K. Behera, and A. Nanthaamornphong, "Paddy Leaf Disease Classification Using EfficientNet B4 With Compound Scaling and Swish Activation: A Deep Learning Approach," *IEEE Access*, vol. 12, pp. 126426–126437, 2024, doi: 10.1109/ACCESS.2024.3451557.

[9] A. Angdresey, I. Y. Kairupan, and A. G. Mongkareng, "Leveraging Convolutional Neural Networks for Multiclass Waste Classification," *Journal of Applied Informatics and Computing*, vol. 9, no. 4, pp. 1137–1145, Aug. 2025, doi: 10.30871/jaic.v9i4.9373.

[10] J. J. C. Simbolon, Robet, and Hendri, "Household Waste Image Classification Using Deep Learning Model," *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, vol. 5, no. 1, pp. 1681–1690, Oct. 2025, doi: 10.59934/jaiea.v5i1.1690.

[11] G. Ahmad *et al.*, "Intelligent waste sorting for urban sustainability using deep learning," *Sci. Rep.*, vol. 15, no. 1, p. 27078, Jul. 2025, doi: 10.1038/s41598-025-08461-w.

[12] T. Kurniawan, K. Khadijah, and R. Kusumaningrum, "An Efficient Model for Waste Image Classification Using EfficientNet-B0," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 3, pp. 1147–1158, Jun. 2025, doi: 10.52436/1.jutif.2025.6.3.4417.

[13] M. A. Mohammed *et al.*, "Automated waste-sorting and recycling classification using artificial neural network and features fusion: a digital-enabled circular economy vision for smart cities," *Multimed. Tools Appl.*, vol. 82, no. 25, pp. 39617–39632, Oct. 2023, doi: 10.1007/s11042-021-11537-0.

[14] I. K. M. C. Qinantha, I. G. A. Indrawan, I. P. S. U. Putra, I. G. A. A. M. Aristamy, and A. G. Willdahlia, "Classification Of Organic And Inorganic Waste Using Resnet50," *Indonesian Journal of Data and Science*, vol. 6, no. 2, pp. 232–240, Jul. 2025, doi: 10.56705/ijodas.v6i2.267.

[15] M. Haqqi, L. Rochmah, A. D. Safitri, R. A. Pratama, and Tarwoto, "Implementation Of Machine Learning To Identify Types Of Waste Using CNN Algorithm," *JURNAL FASILKOM*, vol. 14, no. 3, pp. 761–765, Dec. 2024, doi: 10.37859/jf.v14i3.8116.

[16] L. Du, A. T. S. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Process. Image Commun.*, vol. 81, p. 115713, Feb. 2020, doi: 10.1016/j.image.2019.115713.

[17] Y. Jakhar and M. D. Borah, "Effective near-duplicate image detection using perceptual hashing and deep learning," *Inf. Process. Manag.*, vol. 62, no. 4, p. 104086, Jul. 2025, doi: 10.1016/j.ipm.2025.104086.

[18] A. Joshi, A. V Shet, A. S. Thambi, and S. R, "Quality Improvement of Image Datasets using Hashing Techniques," in *2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*, IEEE, Jan. 2023, pp. 18–23. doi: 10.1109/IITCEE57236.2023.10091044.

[19] A. Aghabagherloo, A. Abadi, S. Sarkar, V. A. Dasu, and B. Preneel, "Impact of Data Duplication on Deep Neural Network-Based Image Classifiers: Robust vs. Standard Models," in *2025 IEEE Security and Privacy Workshops (SPW)*, IEEE, May 2025, pp. 177–183. doi: 10.1109/SPW67851.2025.00023.

[20] P. Ch, R. Swathi, K. Suneetha, I. Suneetha, B. V. Suresh Reddy, and S. K. Depuru, "Image Capturing and Deleting Duplicate Images through Feature Extraction using Hashing Techniques," *International Journal of Engineering Trends and Technology*, vol. 72, no. 1, pp. 64–70, Jan. 2024, doi: 10.14445/22315381/IJETT-V72I1P107.

[21] Q. Zhang, Q. Yang, X. Zhang, Q. Bao, J. Su, and X. Liu, "Waste image classification based on transfer learning and convolutional neural network," *Waste Management*, vol. 135, pp. 150–157, Nov. 2021, doi: 10.1016/j.wasman.2021.08.038.

[22] M. Malik *et al.*, "Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models," *Sustainability*, vol. 14, no. 12, p. 7222, Jun. 2022, doi: 10.3390/su14127222.

[23] L. Nanni, M. Paci, S. Brahnam, and A. Lumini, "Comparison of Different Image Data Augmentation Approaches," *J. Imaging*, vol. 7, no. 12, p. 254, Nov. 2021, doi: 10.3390/jimaging7120254.

[24] W. Zeng, "Image data augmentation techniques based on deep learning: A survey," *Mathematical Biosciences and Engineering*, vol. 21, no. 6, pp. 6190–6224, 2024, doi: 10.3934/mbe.2024272.

[25] S. Agustiani, A. Junaidi, R. Aryanti, A. Abdul, and B. Kamil, "Waste Classification using EfficientNetB3-Based Deep Learning for Supporting Sustainable Waste Management," *Bulletin of Informatics and Data Science*, vol. 4, no. 1, pp. 62–70, 2025, doi: 10.61944/bids.v4i1.108.

[26] H. Singh and D. Rai, "24. Improved Efficientnet Transfer Learning Framework with Data Augmentation For Multiclass Skin Cancer Classification," *Int. J. Environ. Sci.*, vol. 11, no. 21, 2025, doi: 10.64252/bjajys38.

[27] Y. Puspitasari, J. S. Jong, and A. G. Prabawati, "23. Comparison Of Convolutional Neural Network Architectures Efficientnet-B4 And Mobilenetv2 In Cataract Disease Detection," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 11, no. 4, pp. 1020–1027, May 2026, doi: 10.33480/jitk.v11i4.7502.

[28] Z. Yao, A. Gholami, S. Shen, M. Mustafa, K. Keutzer, and M. Mahoney, "ADAHESSIAN: An Adaptive Second Order Optimizer for Machine Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10665–10673, May 2021, doi: 10.1609/aaai.v35i12.17275.