

Performance analysis of MobileNetV2 based automatic waste classification using transfer learning

Ricy Firnando¹, Muhammad Ali Buchari², Anna Dwi Marjusalinah³, Willy⁴, Abdurahman⁵,
Rahmat Fadli Isnanto⁶

^{1,2,5,6} Computer Engineering, Universitas Sriwijaya, Indralaya, Indonesia

^{3,4} Informatics Management, Universitas Sriwijaya, Indralaya, Indonesia

ARTICLE INFO

Article history:

Received Jul 06, 2025

Revised Jul 27, 2025

Accepted Jul 31, 2025

Keywords:

CPU-based AI;
Deep Learning;
MobileNetV2;
Transfer Learning;
Waste Classification.

ABSTRACT

The significant increase in global waste requires innovative and accessible solutions, which aligns with Sustainable Development Goal (SDG) 12, which focuses on reducing the environmental impact of human activities. Automatic waste sorting using Computer Vision and Deep Learning offers a promising alternative to labor-intensive and risky manual methods. This study presents the design, implementation, and comprehensive performance analysis of an automated waste classification system, with a specific focus on evaluating its feasibility on hardware without specialized GPU accelerators. By leveraging transfer learning on a lightweight Convolutional Neural Network (CNN) architecture, MobileNetV2, a model was trained to classify six common waste categories: cardboard, glass, metal, paper, plastic, and other waste. The public "Garbage Classification" dataset from Kaggle, consisting of 2,527 images, was used as the basis for training and validation. The experiment was conducted using the tensorflow-cpu library, which does not require a dedicated GPU accelerator. After 10 training epochs, the model achieved a significant validation accuracy of 86.73%. Computational performance analysis showed an efficient average training time of 31.17 seconds per epoch and a fast average inference time of 14.47 milliseconds per image (~69 FPS) on the validation dataset. These findings demonstrate the feasibility of developing an effective AI-based waste classification system on hardware without a GPU accelerator, providing a realistic performance benchmark for the development of low-cost smart bins with embedded waste sorting solutions in the future, thereby contributing to sustainable waste management practices.

This is an open access article under the [CC BY-NC](#) license.



Corresponding Author:

Anna Dwi Marjusalinah,
Faculty of Computer Science,
Universitas Sriwijaya,
Palembang-Prabumulih Road, Indralaya, 30662, Indonesia.
Email: annadwimarjusalinah@unsri.ac.id

1. INTRODUCTION

The increasing quantity of municipal solid waste (MSW) has become a significant global environmental issue, placing substantial strain on ecosystems and human health (Dehghani et al., 2021; Kaza et al., 2018; Vinti et al., 2021). Improper waste disposal, particularly in illegal dumpsites or unmonitored landfills, leads to detrimental effects such as groundwater contamination, land degradation, and increased incidence of diseases like cancer and congenital disabilities (Kaza et al., 2018). The Ministry of Environment and Forestry (KLHK) in Indonesia reports that national waste generation totals tens of millions of tons each year, predominantly consisting of organic and plastic waste (Kementerian Lingkungan Hidup dan Kehutanan (Kementerian Lingkungan Hidup dan Kehutanan (KLHK), n.d.). A considerable amount of this garbage is deposited in landfills without

sufficient sorting and processing, resulting in detrimental effects including soil, water, and air pollution, as well as the forfeiture of potential economic value from recyclable items (George Tchobanoglous & Vigil, 1993). The United Nations' Sustainable Development Goal (SDG) 12, "Responsible Consumption and Production," aims for a significant decrease in waste generation by 2030 via prevention, reduction, recycling, and reuse (3R) initiatives (UN, 2015). Efficient waste segregation is a crucial foundation for attaining elevated recycling rates and promoting a circular economy (Fang et al., 2023; Fotovvatikhah et al., 2025; Lu & Chen, 2022).

Conventional waste sorting techniques predominantly rely on human labor. While human sorters can attain a degree of precision, this method is intrinsically sluggish, erratic, labor-intensive, and subjects workers to toxic substances and unsanitary circumstances (Lu & Chen, 2022). These constraints impede the scalability and efficacy necessary for effective national recycling initiatives. Consequently, there is an immediate want for automated, intelligent, and economical solutions to enhance or substitute manual sorting procedures.

Artificial Intelligence (AI), especially in the domains of Computer Vision and Deep Learning, has exhibited exceptional proficiency in addressing intricate visual perception challenges hitherto deemed exclusive to human capability (Fang et al., 2023; Jin et al., 2023; Mao et al., 2023). CNNs, a leading category of deep neural networks, have established themselves as the standard for image classification, object recognition, and semantic segmentation, attaining superior performance across multiple domains (LeCun et al., 2015; Mao et al., 2023). The utilization of CNNs for automated garbage categorization entails training models to recognize and categorize various waste products based on digital images obtained from cameras (LeCun et al., 2015; Mao et al., 2021). These systems have the capacity to improve the speed, precision, and safety of the sorting process, while concurrently diminishing dependence on manual labor (Jin et al., 2023; Mao et al., 2021; Nowakowski & Pamuła, 2020; Zhang et al., 2021). AI in waste management can achieve waste identification and sorting accuracy ranging from 72.8% to 99.95% , and can also optimize waste logistics, reducing transportation distance by up to 36.8%, cost by 13.35%, and time by 28.22% (Fang et al., 2023).

Transfer Learning (TL) has received a lot of interest in recent years as a machine learning technique that overcomes the problem of insufficient training data by using knowledge from similar tasks or domains. Traditional machine learning techniques frequently fail when applied to tiny data distributions because they presume that training and testing samples are drawn from the same limited data distribution (Hosna et al., 2022). TL tries to bridge this gap by transmitting important information from a source domain to a target domain, hence improving task performance (Hosna et al., 2022; Tian & Feng, 2023). This technique improves learning by linking previous tasks to the current target task, resulting in faster and more efficient answers (Hosna et al., 2022). Transfer learning has been widely used in a variety of domains, including customer review classification, medical diagnosis e.g., skin cancer classification using DCNN models (Ali et al., 2021), and even smart building energy management and occupancy detection (Pinto et al., 2022; Tian & Feng, 2023)

Multiple studies have investigated the application of CNNs for automated trash classification (Gyawali et al., 2020; Mao et al., 2021; Zhang et al., 2021). Architectures such as VGG16, ResNet (He et al., 2016), Inception (Szegedy et al., 2015), and DenseNet (Mao et al., 2021; Zhang et al., 2021) have been effectively utilized, frequently demonstrating elevated classification accuracy on benchmark datasets such as TrashNet (Mao et al., 2021; Yang & Thung, 2016). Particularly, the MobileNetV2 architecture has gained significant attention due to its efficiency and effectiveness across various image classification tasks. It has been successfully applied in diverse domains, including fruit classification for industrial applications (Shahi et al., 2022), lung tumor classification from CT images (Gao et al., 2025), face mask detection in real-time (Almghraby & Elnady*, 2021; Hussain et al., 2022), and even identification of maize seed varieties (Ma et al., 2023). More directly relevant to this study, MobileNetV2 has also been successfully applied for waste classification, demonstrating its suitability for this specific problem (Yong et al., 2023). For instance, an optimized DenseNet121 model achieved 99.6% accuracy on TrashNet (Mao et al., 2021), and transfer learning with DenseNet169 has been proposed to improve efficiency and accuracy in waste classification (Zhang et al., 2021). While these accuracy outcomes are quite encouraging, the main emphasis frequently lies in attaining the utmost accuracy (Dong et al., 2025). This is typically achieved through the utilization of big and intricate models that necessitate training and inference on high-performance computing equipment, such as GPUs. Furthermore, existing public waste datasets often suffer from

uneven data distribution, single backgrounds, obvious features, and small sample sizes, which do not fully reflect real-world needs (Zhang et al., 2021).

Nonetheless, in practical implementations, particularly in developing nations or for small-scale initiatives such as smart bins at the household or municipal level, cost-effectiveness and computing efficiency are essential considerations. Deploying substantial models on constrained hardware (edge devices or conventional laptops) is frequently unfeasible because of financial limitations, energy consumption, and latency issues (Sheng et al., 2020; Yong et al., 2023). (Fan et al., 2023). Fine-tuning the entire parameter set of large pre-trained models can also become impractical as model sizes grow rapidly, leading to significant memory and storage costs (Sung, 2022). More in-field studies and collaboration between academia, industry, and policymakers are needed to validate TL performance in real-world applications and make a true impact (Pinto et al., 2022).

While previous studies have demonstrated impressive accuracy using deep learning models like VGG16 and DenseNet (Mao et al., 2021; Zhang et al., 2021), they often achieve these results using powerful and expensive GPU hardware. This reliance on high-performance computing creates a practical barrier for widespread, affordable implementation. A significant research gap, therefore, exists in evaluating the performance of efficient models on standard, CPU-only hardware. This gap is critical because real-world solutions, especially for household or small-city scale applications, must be cost-effective.

This study directly addresses this gap. We specifically choose the MobileNetV2 architecture because it is designed for computational efficiency on resource-constrained devices. By analyzing its performance on a standard CPU, we aim to provide a realistic benchmark for developing low-cost edge computing systems, such as smart bins. This approach shifts the focus from achieving the highest possible accuracy to finding a practical balance between good performance and real-world feasibility, a justification that is central to our research.

This study seeks to address that deficiency by developing, executing, and rigorously evaluating the efficacy of a waste classification system utilizing the MobileNetV2 architecture. This study's originality is in its concentrated examination of the trade-off among accuracy, training efficiency, and the feasibility of real-time implementation on conventional CPUs. This emphasis establishes a vital basis for the future advancement of affordable smart gadgets. This study specifically intends to:

1. Create an automated garbage classification model utilizing the MobileNetV2 CNN architecture via transfer learning methodologies.
2. Train and validate the model utilizing a public dataset, meticulously assessing its categorization efficacy across six prevalent trash categories.
3. Perform a comprehensive evaluation of the model's computational efficiency, including total training duration, mean time per epoch, and average inference time per image, when executed alone on a conventional laptop CPU.
4. Supply empirical benchmarks for the practical execution of AI-driven waste sorting systems on consumer hardware, thus aiding the advancement of more accessible and scalable intelligent waste management solutions, in accordance with the principles of SDG 12.

The organization of this study is as follows: Section 2 delineates the research technique, encompassing dataset preparation, model architecture, and experimental configuration. Section 3 delineates and analyzes the experimental findings, emphasizing classification accuracy and computing efficiency. Ultimately, Section 4 ends the work and presents recommendations for future research.

2. RESEARCH METHOD

This section delineates the methodical methodology utilized in this research, including data collection and preparation, deep learning model development, the training procedure, and the assessment metrics applied. All phases were engineered to be reproducible with conventional computational resources and publicly accessible software.

Data Collection and Preprocessing

The principal dataset included in this study is Garbage Classification, a publicly accessible dataset sourced from the Kaggle platform (Cchang, 2018). This dataset was selected for its accessibility and its thorough representation of frequently encountered trash kinds. The dataset

consists of 2527 RGB photos, rigorously classified into six unique categories: 'cardboard', 'glass', 'metal', 'paper', 'plastic', and 'trash' (other waste). Table 1 presents the comprehensive distribution of photos for each category.

Table 1. Distribution of Images by Class in the "Garbage Classification" Dataset

No	Waste Classification	Quantity of Images	Proportion of Total
1	cardboard	403	15.95%
2	glass	501	19.83%
3	metal	410	16.22%
4	paper	594	23.51%
5	plastic	482	19.07%
6	trash	137	5.42%
Total		2527	100.00%

The dataset was partitioned into training and validation sets in an 80:20 ratio, resulting in 2022 images for training and 505 images for validation. The Keras `image_dataset_from_directory` function was employed for this partitioning, with the `option seed = 123` to ensure reproducibility and prevent data leakage between the datasets.

While a standard 80:20 split with a fixed seed ensures reproducibility for this specific study, we acknowledge that the relatively small size of the dataset (2,527 images) presents a potential limitation. A single data partition might not fully represent the model's generalization capability on entirely new data. For future, more robust evaluations, employing a k-fold cross-validation strategy (e.g., 5-fold or 10-fold) would be highly recommended. This approach would provide a more reliable estimate of the model's performance by training and validating it on different subsets of the data, thereby reducing the risk of evaluation bias.

A series of preprocessing processes were implemented on the photos prior to their input into the neural network:

1. **Image Resizing:** All images, irrespective of their original proportions, were consistently downsized to 224x224 pixels. This dimension is the standard input size required by the MobileNetV2 architecture, ensuring uniformity across all input samples.
2. **Data Augmentation:** To reduce the likelihood of overfitting, especially with a moderately big dataset, and to improve the model's capacity to generalize to unseen data, data augmentation techniques were implemented in real-time on the training data throughout the training process. The employed augmentations comprised *RandomFlip('horizontal')*, which horizontally flips photos with a random chance, and *RandomRotation(0.2)*, which randomly rotates images by up to 20% of 360 degrees. The operations were executed as *tf.keras.Sequential* layers incorporated into the model pipeline.
3. **Model Input Normalization:** The MobileNetV2 architecture, pre-trained on ImageNet, requires input pixel values to be within the range of $[-1, 1]$. The normalization was accomplished by incorporating a *tf.keras.layers.Rescaling(1./127.5, offset = -1)* layer into the model. This layer normalizes pixel values (initially within the range $[0, 255]$) by multiplying by $1/127.5$ and subsequently subtracting 1, so converting them to the intended range. The incorporation of a Rescaling layer guarantees that this normalization procedure is integrated into the exported model, eliminating the necessity for manual execution during inference.
4. **Dataset Performance Optimization:** The *cache()* and *prefetch(buffer_size = tf.data.AUTOTUNE)* methods from the *tf.data.Dataset* API were employed to enhance the data loading pipeline by caching data in memory and pre-fetching batches to reduce I/O latency during training.

Model Architecture and Transfer Learning

This categorization technique is fundamentally based on a CNN model utilizing Transfer Learning principles. This method allows the model to utilize information gained from extensive, general datasets (ImageNet) and subsequently tailor it to a more specialized job (waste classification). In parameter-based transfer learning, the weights of a pre-trained model are used for initialization and can be adjusted through a fine-tuning process (Pinto et al., 2022).

1. The fundamental architecture chosen is MobileNetV2 (Sandler et al., 2018). MobileNetV2 is distinguished for its computational efficiency, rendering it appropriate for mobile devices and

embedded systems. This design utilizes depthwise separable convolutions, inverted residual blocks, and linear bottlenecks to attain an optimal equilibrium between accuracy and model size. Pre-trained weights from the ImageNet (Deng et al., 2009) dataset were employed, with the top classification layer (the fully connected layer for 1000 ImageNet classes) omitted by configuring the parameter *include_top = False*. This approach aligns with common practices in transfer learning for image classification tasks, where the pre-trained convolutional base acts as a robust feature extractor (Iman et al., 2023; Kim et al., 2022).

2. Feature Extraction (Freezing the Base Model): To maintain the extensive, generalized features acquired by MobileNetV2 from ImageNet (including edge detection, textures, and shapes), all layers of the pre-trained base model (*base_model*) were rendered non-trainable by setting *base_model.trainable = False*. This guarantees that their weights remain unchanged during the initial training period, enabling the model to operate as a static feature extractor. This strategy is particularly beneficial when the target dataset is small and similar to the source dataset, as it prevents overfitting and reduces training time (Iman et al., 2023).
3. A Custom Head Classifier was incorporated atop the output from the frozen MobileNetV2 basic model, specifically engineered for the classification of six trash kinds. This head classifier consists of the subsequent layers:
 - a. *tf.keras.layers.GlobalAveragePooling2D()*: This layer calculates the average of each feature map produced by the last convolutional block of MobileNetV2, hence condensing spatial dimensions into a singular feature vector for each image.
 - b. A dropout layer with a rate of 0.2 is applied to the pooled features using *tf.keras.layers.Dropout(0.2)*. Dropout is a regularization method that randomly deactivates a proportion of input units after each training update, hence mitigating overfitting.
 - c. A dense output layer with *NUM_CLASSES* (i.e., 6) units, utilizing the ‘softmax’ activation function, corresponds to the six waste categories. The softmax activation function is utilized to generate a probability distribution over these classes, ensuring that the total probability for each input image equals 1.

The entire model architecture is encapsulated in Table 2. The model comprises a total of 2,265,670 parameters. Nevertheless, as a result of the MobileNetV2 base being frozen, merely 7,686 parameters (about 0.34% of the total) associated with the custom head classifier are subject to training. This significantly diminishes computational expenses and data prerequisites for training. Such parameter-efficient transfer learning algorithms are critical given the continuously rising complexity of models, making fine-tuning the full parameter set impracticable due to memory and storage costs (Sung, 2022).

Table 2. Overview of Waste Classification Model Architecture

Type of Layer	Output Configuration	Quantity of Parameters	Adjustable Parameters
Input Layer	(None, 224, 224, 3)	0	0
Sequential (Data Augmentation)	(None, 224, 224, 3)	0	0
Rescaling (Normalization)	(None, 224, 224, 3)	0	0
MobileNetV2 (Static Based Model)	(None, 7, 7, 1280)	2.257.984	0
Global Average Pooling 2D	(None, 1280)	0	0
Dropout (rate=0.2)	(None, 1280)	0	0
Dense (Output Layer, 6 Classes)	(None, 6)	7.686	7.686
Total		2.265.670	7.686

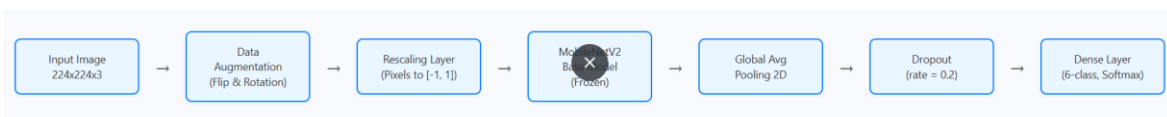


Figure 3. Visualization of the Proposed Model Architecture Pipeline.

The diagram illustrates the data flow, starting from an input image, passing through data augmentation and normalization, feature extraction by the frozen MobileNetV2 base, and finally classification by the custom trainable head to produce output probabilities for the six waste categories.

Training and Experimental Configuration

The model was constructed using the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.001. The utilized loss function was SparseCategoricalCrossentropy, suitable for multi-class classification tasks using integer labels. Accuracy functioned as the primary performance metric during the training procedure. The model was trained for a total of 10 epochs with a batch size of 32. This number was determined empirically based on monitoring the training and validation performance curves during initial experiments (as illustrated in Figure 2). It was observed that after approximately 10 epochs, the validation accuracy began to plateau, and the gap between training and validation loss stopped narrowing, indicating that further training would yield diminishing returns and increase the risk of overfitting. This approach serves as a practical form of early stopping to achieve a good balance between model performance and computational efficiency.

All operations for training and testing the model were executed using a Lenovo consumer laptop (Model 81Y4) with the following specifications:

1. Processor: Intel(R) Core(TM) i7-10750H CPU functioning at 2.60GHz (12 cores)
2. RAM: 16 GB
3. GPU: NVIDIA GeForce GTX 1650 (This study utilized only the CPU using a tensorflow-cpu installation to simulate limited hardware conditions).
4. Operating System: Windows 11 Home Single Language 64-bit.
5. The utilized software consisted of:
 - a. Programming Language: Python 3.10 (inside an Anaconda environment).
 - b. Deep Learning Framework: TensorFlow 2.19, particularly the tensorflow-cpu package, employing the Keras API.
 - c. Supporting Libraries: NumPy for numerical analysis, Matplotlib and Seaborn for data visualization, and Scikit-learn for performance metric evaluation.
 - d. Integrated Development Environment: Spyder.

Metrics for Evaluating Performance

To comprehensively assess model performance, the following standard metrics are calculated based on predictions from the validation dataset:

1. Accuracy: The proportion of photos accurately categorized relative to the total number of images.
2. Precision: The classifier's ability to avoid misclassifying negative samples as positive. Calculated for a particular class as TP divided by the sum of TP and FP.
3. Recall: The classifier's ability to recognize all positive examples. Calculated for a particular class as TP divided by the sum of TP and FN.
4. F1-Score: A singular metric that denotes the harmonic mean of precision and recall, equilibrating both parameters. For a certain class, calculated as $2 * (Precision * Recall) / (Precision + Recall)$.
5. Confusion Matrix: A tabular structure that facilitates the visualization of an algorithm's performance. Each row represents an instance from the actual class, whereas each column indicates an instance from the anticipated class.
6. Computational Efficiency: Total training length, average training time per epoch, aggregate inference time on the validation set, and average inference time per image on the validation set. ($TP = True Positives$; $FP = False Positives$; $FN = False Negatives$)

3. RESULTS AND DISCUSSIONS

This section delineates and examines the quantitative and qualitative results derived from the model training process and its performance assessment on the validation dataset.

Evaluation of the Waste Classification Model Training

The training of the MobileNetV2 model, employing a transfer learning approach across 10 epochs, demonstrates the model's ability to identify visual patterns that differentiate various waste categories. Table 3 delineates the evolution of accuracy and loss metrics during each epoch for both training and validation datasets, together with the requisite computation time.

Table 3. Progression of Accuracy, Loss, and Model Duration During Training (10 Epochs)

Epoch	Training Precision	Training Loss	Validation Precision	Validation Loss	Time pe Epoch (seconds)
1	0.3855	1.6092	0.7545	0.7138	53
2	0.7122	0.7694	0.7901	0.5721	33
3	0.7621	0.6503	0.8396	0.5018	28
4	0.7999	0.5552	0.8337	0.4668	28
5	0.8205	0.5123	0.8495	0.4481	28
6	0.8406	0.4621	0.8475	0.4332	28
7	0.8505	0.4369	0.8515	0.4287	28
8	0.8455	0.4252	0.8535	0.4131	29
9	0.8593	0.4168	0.8495	0.4035	28
10	0.8607	0.3925	0.8673	0.3911	27

Table 3 indicates that the training accuracy increased consistently from 38.55% in the initial epoch to 86.07% in the tenth epoch. A similar trend was observed in the validation data accuracy, which rose from 75.45% to 86.73%. The decrease in loss values for both training and validation datasets indicates that the model successfully reduced its prediction errors during the training phase. Figure 2 illustrates the accuracy and loss trajectories over the course of 10 training epochs.

Although additional experiments, like those utilizing DenseNet121 (Mao et al., 2021), attained an accuracy of 99.6%, and DenseNet169 (Zhang et al., 2021) demonstrated similarly excellent outcomes, these investigations frequently depend on larger, more intricate models and necessitate GPU accelerators for both training and inference. This research demonstrates that the lightweight and efficient MobileNetV2 may attain notable accuracy (86.73%) utilizing solely a CPU. This underscores the primary significance of our work in reconciling accuracy and computing efficiency, an essential factor for practical deployment on resource-limited systems.

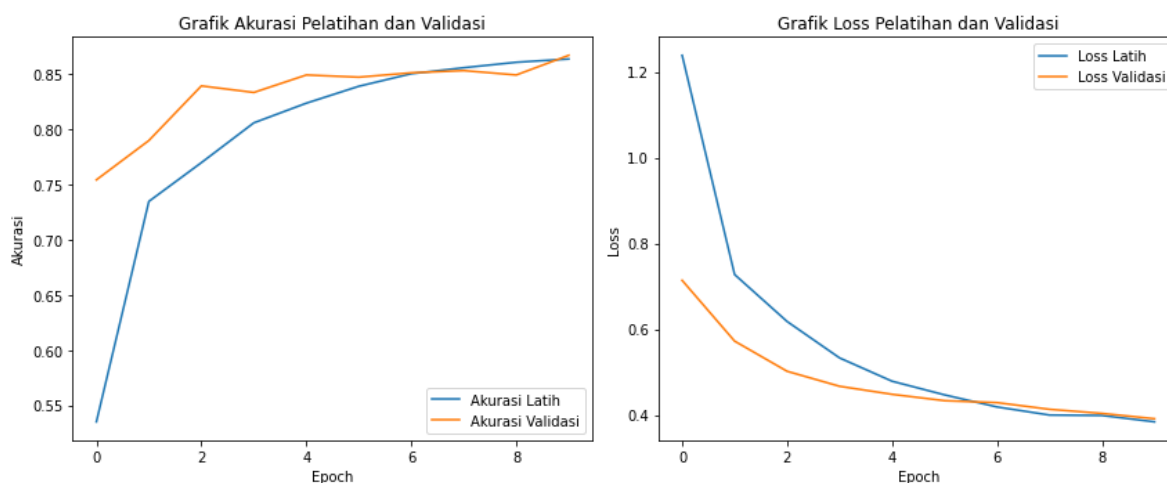


Figure 2. Training Curves for Model Accuracy (left) and Loss (right)

Figure 2 indicates that the validation accuracy curve often corresponds with the ascending trajectory of the training accuracy curve, and both curves appear to converge. Similarly, the validation loss curve decreased in tandem with the training loss curve. This pattern indicates that the model did not experience significant overfitting; it did not merely recall the training data but also successfully adapted its knowledge to fresh data.

Evaluation of Final Classification Performance on Validation Data

After completing 10 training epochs, the final model was rigorously assessed using the entire validation dataset (505 pictures). The outcomes of the classification performance assessment are presented in Table 4 (Classification Report) and Figure 3 (Confusion Matrix).

Table 4. Classification Report on Validation Data for Model Efficacy

Classificaiton of Waste	Precision	Recall	F1-Score	Assistance
cardboard	0.9737	0.8916	0.9308	83
glass	0.8182	0.8738	0.8451	103
metal	0.8875	0.9103	0.8987	78
paper	0.8837	0.9194	0.9012	124
plastic	0.8193	0.7727	0.7953	88
trash	0.7778	0.7241	0.75	29
Accuracy			0.8673	505
Macro Avg	0.86	0.8486	0.8535	505
Weighted Avg	0.8684	0.8673	0.8671	505

Table 4 indicates that the model’s overall accuracy on the validation data is 86.73%. This indicates that approximately 86-87 out of every 100 garbage photos in the validation set can be successfully recognized by the model. Assessing precision and recall for each category provides enhanced understanding:

1. The ‘cardboard’ class exhibits the highest precision (0.9737), indicating that when the model classifies an object as cardboard, the prediction is highly probable to be correct.
2. The ‘paper’ and ‘metal’ classes demonstrate commendable F1-scores (about 0.90), signifying a robust balance between precision and recall for these categories.
3. The categories ‘plastic’ and ‘garbage’ demonstrate relatively lower F1-scores of 0.7953 and 0.7500, respectively. This suggests that the model may encounter greater difficulty in accurately distinguishing between these two classes or identifying all relevant data. The restricted sample size for the ‘trash’ class (29 photos) may adversely affect performance in this category.

Figure 3 illustrates the Confusion Matrix, offering a more explicit depiction of misclassifications among classes.

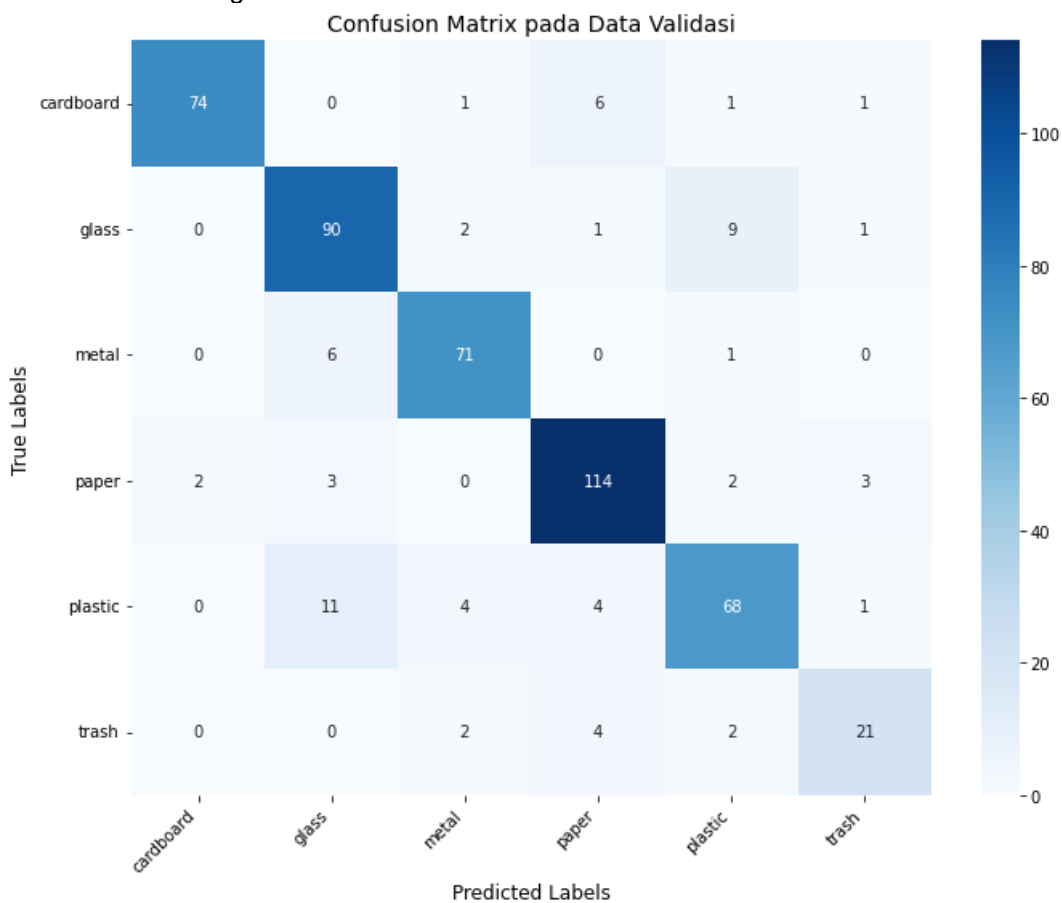


Figure 3. Confusion Matrix Illustrating Model Efficacy on Validation Data

Figure 3 indicates that the models demonstrate varying degrees of efficacy in classifying waste categories. The paper attained the highest performance, accurately categorizing 114 of 124 samples. Subsequently, Metal exhibited robust performance by accurately categorizing 71 out of 78 samples, with only 6 instances misclassified as 'glass' due to slight ambiguity.

Cardboard was correctly identified in 74 out of 83 occurrences, with the predominant error being its misclassification as 'paper' in 6 cases. Glass had commendable performance, accurately identifying 90 out of 103 cases; nevertheless, there was a propensity to misclassify as 'plastic' in 9 instances, a misjudgment that is visually justifiable due to the similarity of the transparent materials.

The category of plastic demonstrated the highest level of confusion, particularly with 'glass' (11 occurrences), with only 68 out of 88 samples correctly categorized. Ultimately, General Waste (Trash) demonstrated the lowest recall (21 of 29 right), frequently misclassified as 'paper' (4 occurrences) and 'metal' (2 instances). The considerable diversity within the 'waste' category and the restricted sample size are likely principal factors contributing to the subpar performance in this classification.

Assessment of Computational Performance

This study examines the model's computational efficiency on a standard laptop CPU to assess the feasibility of its deployment on resource-constrained hardware.

1. Training Duration: The total time required to train the model for 10 epochs was 311.67 seconds (about 5.2 minutes), yielding an average of 31.17 seconds per epoch. The relatively short training period underscores the efficacy of the transfer learning approach and the MobileNetV2 architecture, facilitating rapid iteration and experimentation without the need for a GPU.
2. Inference Duration: The model used 7.31 seconds to provide predictions for all 505 validation photos. The average inference time per image is merely 14.47 milliseconds, enabling processing at around 69 frames per second. This velocity has considerable practical ramifications. The model can be incorporated into a smart bin prototype with a microcontroller (e.g., Raspberry Pi or Arduino) driven by a CPU. Upon detection of an object by a sensor, the model can classify it in under one second, enabling actuators such servo motors to route the waste into the correct compartment (e.g., metal, plastic, paper) instantaneously. This computational efficiency removes the financial and energy obstacles usually linked to GPU utilization, rendering AI technology for trash management broadly accessible. Table 5 summarizes the computational performance.

Table 5. Summary of Model Computational Efficiency on Intel Core i7-10750H CPU

Computational Assesment	Value
Total Training Duration (10 Epochs)	311.67 seconds
Average Training Duration per Epoch	31.17 seconds
Total Inference Duration for 505 Validation Images	7.31 seconds
Average Inference Duration for Each Validation Image	14.47 ms
Potential Inference Frames per Second (FPS)	~69 FPS

The acknowledged computing efficiency, particularly the rapid inference time, is of paramount importance. This indicates that the MobileNetV2 model, although being a deep learning architecture, can provide adequately rapid answers for many practical applications without the necessity for expensive GPU accelerator hardware.

General Discourse

This research reveals that the automated trash classification system, employing MobileNetV2 via transfer learning, achieved an accuracy of 86.73% and notable computational efficiency on a standard laptop CPU. This has significant implications for the development of economical and accessible smart waste management technologies. The model's ability to accurately differentiate among six standard waste categories, coupled with inference speeds suitable for real-time implementation, facilitates practical applications such as smart bins that provide instant feedback to users or compact sorting systems in community recycling centers.

An analysis of the confusion matrix and classification report identified areas for potential model improvement. The confusion between 'glass' and 'plastic,' coupled with diminished

performance in the 'waste' category, suggests that integrating a broader array of data for these classifications or employing more accurate fine-tuning techniques could enhance performance. The relatively limited number of samples in the 'garbage' category, compared to other categories, likely affected its generalization ability.

The principal limitations of this study pertain to the dependence on a dataset that may not fully capture the intricacies of real-world waste (e.g., overlapping or damaged items, or items in suboptimal lighting conditions) and an evaluative focus that is presently confined to quantitative metrics derived from a specified dataset. Further testing under deregulated real-world conditions will improve understanding of the model's robustness.

These results strongly support the proposed innovation: that effective and efficient deep learning techniques for waste classification can be developed and employed on conventional hardware, making this technology more accessible and likely to significantly impact efforts to achieve SDG 12.

4. CONCLUSION

This research has successfully developed, implemented, and assessed an automated waste sorting system utilizing the MobileNetV2 CNN architecture through transfer learning. This work primarily demonstrates the practical viability of implementing an efficient deep learning model on normal CPU-only hardware, addressing a significant gap in research that often emphasizes GPU-dependent systems. Tests conducted on a consumer-grade laptop demonstrated that the model attained a validation accuracy of 86.73% and a real-time inference speed of roughly 69 FPS. These findings provide a strong performance baseline, demonstrating that economical hardware is adequate for developing accessible and scalable smart waste systems. The findings highlight the potential for creating clever bins and small sorting devices that can substantially improve sustainable trash management in accordance with SDG 12. Notwithstanding these encouraging findings, the study possesses shortcomings that offer avenues for subsequent research. The assessment relied on a singular public dataset, and subsequent examination uncovered ambiguity between the 'glass' and 'plastic' classifications. The model's efficacy was significantly reduced in the 'trash' category, presumably due to a constrained sample size and significant variability within that category. To mitigate these limitations, subsequent research should concentrate on utilizing a k-fold cross-validation method for a more reliable performance assessment and augmenting the dataset to encompass a broader range of waste categories under demanding real-world conditions (e.g., partially obscured, deformed, or soiled items). Moreover, subsequent endeavors may encompass the incorporation of the model into a tangible prototype utilizing a single-board computer (such as a Raspberry Pi) and IoT sensors to develop a fully operational smart bin, so further substantiating the effectiveness of transfer learning in practical applications.

REFERENCES

- Ali, M. S., Miah, M. S., Haque, J., Rahman, M. M., & Islam, M. K. (2021). An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. *Machine Learning with Applications*, 5(April), 100036. <https://doi.org/10.1016/j.mlwa.2021.100036>
- Almghraby, M., & Elnady*, A. O. (2021). Face Mask Detection in Real-Time using MobileNetV2. *International Journal of Engineering and Advanced Technology*, 10(6), 104–108. <https://doi.org/10.35940/ijeat.f3050.0810621>
- Dehghani, M. H., Omrani, G. A., & Karri, R. R. (2021). Solid Waste—Sources, Toxicity, and Their Consequences to Human Health. *Soft Computing Techniques in Solid Waste and Wastewater Management, January 2021*, 205–213. <https://doi.org/10.1016/B978-0-12-824463-0.00013-6>
- Dong, Z., Yuan, L., Yang, B., Xue, F., & Lu, W. (2025). Benchmarking computer vision models for automated construction waste sorting. *Resources, Conservation and Recycling*, 213, 108026. <https://doi.org/https://doi.org/10.1016/j.resconrec.2024.108026>
- Fan, H., Dong, Q., Guo, N., Xue, J., Zhang, R., Wang, H., & Shi, M. (2023). Raspberry Pi-based design of intelligent household classified garbage bin. *Internet of Things*, 24, 100987. <https://doi.org/https://doi.org/10.1016/j.iot.2023.100987>
- Fang, B., Yu, J., Chen, Z., Osman, A. I., Farghali, M., Ihara, I., Hamza, E. H., Rooney, D. W., & Yap, P. S. (2023). Artificial intelligence for waste management in smart cities: a review. In *Environmental Chemistry Letters* (Vol. 21, Issue 4). Springer International Publishing. <https://doi.org/10.1007/s10311-023-01604-3>
- Fotovatikhah, F., Ahmady, I., Noor, R. M., & Munir, M. U. (2025). A Systematic Review of AI-Based Techniques for Automated Waste Classification. *Sensors*, 25(10). <https://doi.org/10.3390/s25103181>
- Gao, Z., Tian, Y., Lin, S.-C., & Lin, J. (2025). A CT Image Classification Network Framework for Lung Tumors

- Based on Pre-trained MobileNetV2 Model and Transfer learning, And Its Application and Market Analysis in the Medical Field. *Applied and Computational Engineering*, 133(1), 90–96. <https://doi.org/10.54254/2755-2721/2025.20605>
- George Tchobanoglous, H. T., & Vigil, S. A. (1993). *Integrated Solid Waste Management: Engineering Principles and Management Issues*. McGraw-Hill.
- Gyawali, D., Regmi, A., Shakya, A., Gautam, A., & Shrestha, S. (2020). *Comparative Analysis of Multiple Deep CNN Models for Waste Classification*. <http://arxiv.org/abs/2004.02168>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1). <https://doi.org/10.1186/s40537-022-00652-w>
- Hussain, D., Ismail, M., Hussain, I., Alroobaea, R., Hussain, S., & Ullah, S. S. (2022). Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning. *Wireless Communications and Mobile Computing*, 2022. <https://doi.org/10.1155/2022/1536318>
- Iman, M., Arabia, H. R., & Rasheed, K. (2023). A Review of Deep Transfer Learning and Recent Advancements. *Technologies*, 11(2), 1–14. <https://doi.org/10.3390/technologies11020040>
- Jin, S., Yang, Z., Królczyk, G., Liu, X., Gardoni, P., & Li, Z. (2023). Garbage detection and classification using a new deep learning-based machine vision system as a tool for sustainable waste recycling. *Waste Management*, 162, 123–130. <https://doi.org/https://doi.org/10.1016/j.wasman.2023.02.014>
- Kaza, S., Yao, L. C., Bhada-Tata, P., & Van Woerden, F. (2018). *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050* (Urban Deve). World Bank. <https://doi.org/10.1596/978-1-4648-1329-0>
- Kementerian Lingkungan Hidup dan Kehutanan (KLHK). (n.d.). *Sistem Informasi Pengelolaan Sampah Nasional (SIPSN)*. <https://sipsn.kemenvh.go.id/sipsn/>
- Kim, H. E., Cosa-Linan, A., Santhanam, N., Jannesari, M., Maros, M. E., & Ganslandt, T. (2022). Transfer learning for medical image classification: a literature review. *BMC Medical Imaging*, 22(1), 1–13. <https://doi.org/10.1186/s12880-022-00793-7>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Lu, W., & Chen, J. (2022). Computer vision for solid waste sorting: A critical review of academic research. *Waste Management*, 142, 29–43. <https://doi.org/https://doi.org/10.1016/j.wasman.2022.02.009>
- Ma, R., Wang, J., Zhao, W., Guo, H., Dai, D., Yun, Y., Li, L., Hao, F., Bai, J., & Ma, D. (2023). Identification of Maize Seed Varieties Using MobileNetV2 with Improved Attention Mechanism CBAM. *Agriculture (Switzerland)*, 13(1). <https://doi.org/10.3390/agriculture13010011>
- Mao, W. L., Chen, W. C., Wang, C. T. C., Lin, Y. H., Tian, Y. Y., Feng, Y., Iman, M., Arabia, H. R., Rasheed, K., He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G., Zhu, Z., Lin, K., Jain, A. K., Zhou, J., Kim, H. E., ... Du, L. (2023). Fruit classification using attention-based MobileNetV2 for industrial applications. *Waste Management*, 135(1), 1–16. <https://doi.org/10.1080/01621459.2022.2071278>
- Mao, W. L., Chen, W. C., Wang, C. T., & Lin, Y. H. (2021). Recycling waste classification using optimized convolutional neural network. *Resources, Conservation and Recycling*, 164(August 2020), 105132. <https://doi.org/10.1016/j.resconrec.2020.105132>
- Nowakowski, P., & Pamuła, T. (2020). Application of deep learning object classifier to improve e-waste collection planning. *Waste Management*, 109, 1–9. <https://doi.org/10.1016/j.wasman.2020.04.041>
- Pinto, G., Wang, Z., Roy, A., Hong, T., & Capozzoli, A. (2022). Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. *Advances in Applied Energy*, 5(November 2021), 100084. <https://doi.org/10.1016/j.adapen.2022.100084>
- Shahi, T. B., Sitaula, C., Neupane, A., & Guo, W. (2022). Fruit classification using attention-based MobileNetV2 for industrial applications. *PLoS ONE*, 17(2 February), 1–21. <https://doi.org/10.1371/journal.pone.0264586>
- Sheng, T. J., Islam, M. S., Misran, N., Baharuddin, M. H., Arshad, H., Islam, M. R., Chowdhury, M. E. H., Rmili, H., & Islam, M. T. (2020). An Internet of Things Based Smart Waste Management System Using LoRa and Tensorflow Deep Learning Model. *IEEE Access*, 8, 148793–148811. <https://doi.org/10.1109/ACCESS.2020.3016255>
- Sung, Y. (2022). *VL-A DAPTER: Parameter-Efficient Transfer Learning for Vision-and-Language Tasks*. 2, 5227–5237.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tian, Y., & Feng, Y. (2023). Transfer Learning Under High-Dimensional Generalized Linear Models. *Journal of the American Statistical Association*, 118(544), 2684–2697. <https://doi.org/10.1080/01621459.2022.2071278>
- UN. (2015). *Transforming our world: The 2030 Agenda for Sustainable Development*. <https://sdgs.un.org/2030agenda>

- Vinti, G., Bauza, V., Clasen, T., Medicott, K., Tudor, T., Zurbrügg, C., & Vaccari, M. (2021). Municipal Solid Waste Management and Adverse Health Outcomes: A Systematic Review. *International Journal of Environmental Research and Public Health*, 18(8). <https://doi.org/10.3390/ijerph18084331>
- Yang, M., & Thung, G. (2016). *Classification of the TrashNet dataset* (CS229 Proj). Stanford University.
- Yong, L., Ma, L., Sun, D., & Du, L. (2023). Application of MobileNetV2 to waste classification. *PLoS ONE*, 18(3 March), 1–16. <https://doi.org/10.1371/journal.pone.0282336>
- Zhang, Q., Yang, Q., Zhang, X., Bao, Q., Su, J., & Liu, X. (2021). Waste image classification based on transfer learning and convolutional neural network. *Waste Management*, 135(August), 150–157. <https://doi.org/10.1016/j.wasman.2021.08.038>