Journal of Applied Informatics and Computing (JAIC)
Vol.9, No.4, August 2025, pp. 1610~1624
e-ISSN: 2548-6861

A Comparison of MobileNetV2 and VGG16 Architectures with Transfer Learning for Multi-Class Image-Based Waste Classification

Raffa Adhi Kumala [1], Christy Atika Sari [2]*, Eko Hari Rachmawanto [3]
Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia
111202214177@mhs.dinus.ac.id [1], christy.atika.sari@dsn.dinus.ac.id [2], eko.hari@dsn.dinus.ac.id [3]

Article Info

Article history:

Received 2025-06-25
Revised 2025-07-07
Accepted 2025-07-10

Keyword:

Classification,
Convolutional Neural Network,
MobileNetV2,
VGG16,
Waste Management.

ABSTRACT

Effective waste management represents a global challenge with significant environmental and public health impacts. Despite existing waste classification systems achieving high accuracy rates, a critical research gap exists in determining optimal CNN architectures for real-world deployment constraints, particularly regarding computational efficiency versus classification accuracy trade-offs. We compared two Convolutional Neural Network (CNN) architectures MobileNetV2 and VGG16 for classifying ten types of waste using image-based analysis. Using transfer learning approach, both models were modified for waste classification tasks by adding custom layers to pre-trained models. The dataset contained 19,762 images balanced to 9,440 samples through under-sampling techniques and enhanced with data augmentation to increase variation. Results demonstrated that MobileNetV2 achieved 95.6% test accuracy with precision 0.93, recall 0.93, and F1-score 0.93, significantly outperforming VGG16's 89.13% accuracy with precision 0.91, recall 0.90, and F1-score 0.90. Beyond superior accuracy, MobileNetV2 also demonstrated higher computational efficiency with 350ms/step training time compared to VGG16's 700ms/step, and more consistent performance across all waste categories.

I. INTRODUCTION

The increase in waste volume, which is directly proportional to global population growth, has created complex waste management challenges. This has serious environmental and public health impacts, such as the emission of harmful gases that threaten human health [1]. In the United States, data shows that only 30% of the 75% of waste that can potentially be recycled is successfully processed, due to limitations in effective classification systems [2]. Although various studies have demonstrated the successful implementation of convolutional neural networks (CNNs) for high-accuracy waste classification, there is a significant research gap in determining the optimal architecture for real-world system implementation. Previous studies have focused on achieving maximum accuracy, often overlooking the trade-off between classification performance and computational efficiency. This is a crucial consideration for implementation on devices with limited resources. The global context highlights the urgent need for automated waste classification systems, but their implementation in developing countries such as Indonesia faces challenges related to limited computational infrastructure. This necessitates the development of solutions that are both accurate and computationally efficient.

Challenges in implementing an automatic waste classification system include variations in lighting conditions and background complexity, as well as the diversity of similar types of waste [3], this requires the selection of an appropriate model architecture and optimal training strategies. Advances in artificial intelligence technology, particularly in computer vision, offer promising solutions for automating waste classification. Recent research shows that implementing Convolutional Neural Network (CNN) models can greatly improve the efficiency and accuracy of identifying waste types [3]

Several studies have demonstrated that deep learning methods can achieve accuracy rates of up to 99% in the classification of recycled products [4]. The fusion of deep features approach has also been shown to improve waste classification performance by combining the strengths of different architectures to achieve greater accuracy [5]. Using pre-trained models such as InceptionV3, Xception, and ResNet50, the transfer learning method also showed a substantial improvement in the classification of plastic waste [6]. Of the various neural network architectures, VGG16 and MobileNetV2 are particularly effective models for classifying rubbish in images. VGG16 is known to achieve an accuracy of over 90% in identifying types of rubbish based on transfer learning implementation thanks to its 16 layers deep structure [7]. MobileNetV2, sebagai arsitektur yang dioptimalkan untuk efisiensi komputasi, menunjukkan keunggulan dalam implementasi pada perangkat dengan sumber daya terbatas, seperti sistem edge computing [4]. The latest innovations, such as the Vision Transformer, have emerged as promising alternatives that can overcome the limitations of the receptive field in conventional convolutional neural networks (CNNs), particularly when it comes to implementation on portable devices [8]. Song et al. demonstrated that the two-level fusion method for classifying construction waste can achieve an accuracy of up to 97.60% using MobileNetV2 [9]. In order to create a sustainable environment, intelligent and automated waste management systems are required. One example is the Smart Trash system, which classifies waste automatically [10].

This study analyses and compares the performance of two popular CNN architectures, MobileNetV2 and VGG16, in the context of image-based waste classification. The primary focus is on accuracy, computational efficiency and potential implementation in automated waste classification systems. This research is novel in that it comprehensively evaluates the trade-off between accuracy and computational efficiency in the context of multi-class waste classification, an area that has not been thoroughly explored to date. The results are expected to contribute significantly to the development of more effective waste classification systems, thereby improving recycling rates and supporting sustainable waste management.

II. METHODS

This study conducts a comprehensive comparison of two convolutional neural network (CNN) architectures, MobileNetV2 and VGG16, for digital image-based waste classification. The study's primary objective is to evaluate the balance between prediction accuracy and computational efficiency, considering the structural complexity of both models.

As Taherkhani et al. [26] explained, the transfer learning approach is highly effective at improving model performance, particularly when the training dataset is limited, as is the case with waste classification involving significant visual variation. This study identified the various advantages and limitations of each architecture in classifying waste images with diverse visual variations by implementing transfer learning techniques, freezing the weights of the base model trained on the ImageNet dataset, and adding custom layers.

These results are consistent with the findings of Risfendra et al. [32], who compared the performance of various CNN architectures, including MobileNetV2 and VGG16, for waste classification. They found significant differences between the two models. The findings from this study are expected to provide recommendations for the optimal architecture for developing automatic waste classification systems for implementation on devices with limited computational resources, as highlighted by White et al. [33], as well as for systems that prioritise high accuracy.

The research methodology was designed systematically and comprised eight main stages: Data preparation for collecting the waste dataset; data exploration for analysing data characteristics and distribution; data preprocessing for standardising and normalising images; data splitting for dividing the dataset into strata; data augmentation for increasing data variety; model development for developing an architecture with transfer learning; model training with an optimal configuration; and evaluation for analysing model performance with various metrics. Figure 1 illustrates the structured methodology flowchart that was implemented to ensure a fair and comprehensive comparison between the two architectures.

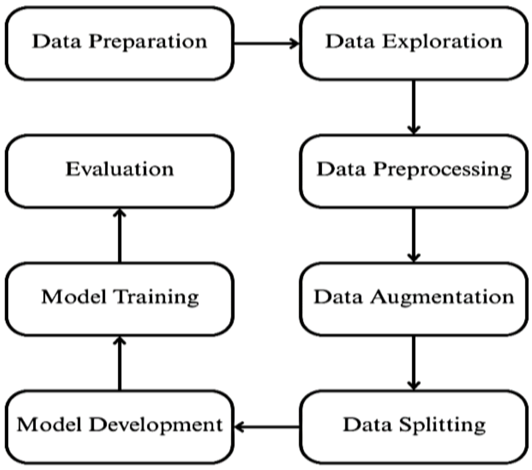

Figure 1 Research Methodology Framework

A. Data Preparation

Data preparation is a fundamental step in developing an accurate waste classification model. For this study, the dataset was obtained from the Kaggle platform and consists of 19,762 images covering ten waste categories: Metal, Glass, Biological Waste, Paper, Batteries, General Waste, Cardboard, Shoes, Clothes and Plastic. Figure 2 shows representative samples of each waste category in the dataset. As shown in Figure 2, each waste category has distinctive visual characteristics. Metal (food cans), glass (green bottles), cardboard (milk cartons), shoes (patterned sandals), batteries (various sizes), biological waste (fresh vegetables), paper (brown paper), clothes (knitted garments), plastic (transparent plastic bottles) and general waste. The significant visual variation between classes in Figure 2 reflects the complexity of the classification challenge. For example, some objects, such as cans (metal), glasses (glass) and beverage cartons (cardboard), can display similar visual characteristics under certain conditions. The diversity of shapes, textures, colours and sizes of objects within each category, as observed in Figure 2, highlights the importance of a comprehensive dataset to ensure the model can perform effectively in various real-world conditions [11].

This dataset was selected because it contains a diverse range of images of rubbish, which is ideal for training effective deep learning models. Using datasets from public sources such as Kaggle enables replication and comparison of results with those of similar research. Figure 2 shows the visual variety of the dataset, which covers a wide range of image capture conditions, from single objects with clean backgrounds to objects in more complex contexts.

The data preparation stage begins with the collection and verification of the dataset to ensure its integrity. An initial analysis is then conducted to identify the dataset's characteristics, such as the number of images per category and the available image formats. All images in the dataset are JPG files showing trash objects against different backgrounds, with different lighting and from different angles, as shown in Figure 2. This diversity of visual conditions is important to ensure that the model can function properly in various real-world scenarios [11]. The quality and suitability of the collected data greatly influence the final performance of the classification model. This comprehensive dataset of rubbish, which features clear visual representations as shown in Figure 2, supports the research objective of comparing the effectiveness of two CNN architectures in classifying different types of rubbish. This could contribute significantly to automated waste management systems [12]. Good data preparation, supported by an in-depth knowledge of the visual characteristics of each waste class, as illustrated in Figure 2, allows more sophisticated classification techniques to be implemented and improves the model's accuracy in identifying and classifying waste in different scenarios [13].

Further analysis of the image samples in Figure 2 reveals that each class possesses unique characteristics that can be exploited by the CNN model to learn features. For instance, the Metal class exhibits reflective properties and consistent geometric shapes, whereas the Biologic class features organic textures and natural colour variations. Understanding these visual features is important for data pre-processing and augmentation, which will be explained in the next subsection.

Figure 2 Dataset Sample

B. *Data Exploration*

The data exploration stage was conducted to understand the characteristics and distribution of the dataset prior to the model training process. Exploratory analysis in this study revealed a significant imbalance in the distribution of the ten waste classes, as illustrated in Figure 3. Visualisation of the original dataset distribution clearly shows the variation in sample numbers between classes: the Clothes class has the highest number of samples (5,327 images), while the Battery class has the lowest (944 images).

As found by Wang et al., the level of imbalance visualised in Figure 3 needs to be addressed, as it can cause the model to be biased towards the majority class and ignore the characteristics of the minority class [14]. Figure 3 shows that the distribution analysis reveals a substantial imbalance, with the ratio between the class with the largest sample (Clothes) and the class with the smallest sample (Battery) being 5.6:1. As can be seen from the analysis results shown in Figure 3, the three classes with the highest number of samples are Clothes (5,327), Glass (3,061) and Plastic (1,984), while the three classes with the lowest number of samples are Battery (944), Trash (947) and Biological Waste (997). This uneven distribution also indicates medium-sized clusters of classes, such as Cardboard (1,825), Shoes (1,977), Paper (1,680) and Metal (1,020), which are located between the majority and minority classes.

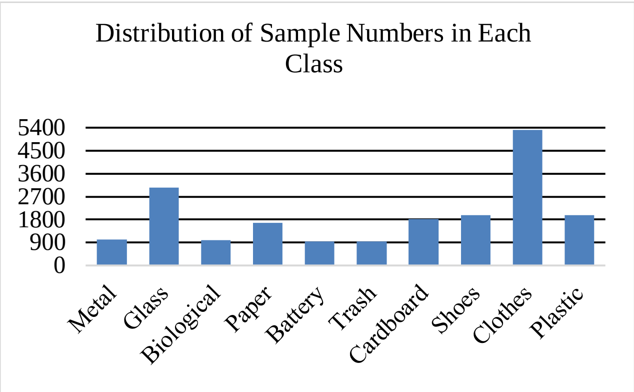

Figure 3 Distribution of Sample Numbers in Each Class

This distribution visualisation reveals a significant imbalance between classes, a common issue with multi-class image classification datasets that can result in decreased model accuracy for minority classes, as highlighted by Saini and Susan [15]. Figure 3 also shows that four classes (clothes, glass, cardboard and shoes) account for over 60% of the total dataset, while the three classes with the fewest samples account for less than 15%. Further exploration of image characteristics revealed variations in background, lighting and object orientation that can affect model performance. A deep understanding of the structure and imbalance of the dataset, as illustrated in Figure 3, is important for determining appropriate preprocessing and augmentation strategies that will improve the quality of the training data. As stated by Jin et al. [16], Handling class imbalance properly can improve the model's ability to recognise all classes fairly and evenly.

As shown in Figure 3, the implication of the imbalanced distribution is that a specific strategy is required in the preprocessing stage to ensure the model can learn effectively from all classes. Further statistical analysis shows that the coefficient of variation for class distribution is 0.85, indicating a high level of imbalance that requires intervention through balancing techniques. This will be discussed in the next subsection.

C. Data Preprocessing

The data preprocessing stage is an important step in preparing the dataset for optimal processing by the CNN model. To ensure consistency in input dimensions, all images in the dataset are standardised to a size of 224×224 pixels in JPG format. Standardising image size is important because CNN architectures such as MobileNetV2 and VGG16 require uniform input dimensions. As stated by Li et al. [17], standardising images can reduce irrelevant variability and help models focus on features that are important for classification. Pixel value normalisation is implemented by transforming the intensity range from 0–255 to 0–1 using the ImageDataGenerator parameter rescale=1.255. This process accelerates convergence during model training and improves the stability of the learning process, as demonstrated by Mullick et al. [18]. Only rescaling was performed on the validation and testing data to maintain the original characteristics of the images during model performance evaluation, with no additional augmentation.

Addressing class imbalances identified during the data exploration stage is a critical pre-processing priority that significantly impacts model performance and generalisation capability. The extreme class imbalance in the original dataset (clothes: 5,327 vs battery: 944, ratio 5.6:1) necessitated careful consideration of balancing strategies. Random under-sampling was selected over oversampling techniques based on several compelling theoretical and practical considerations.

The choice of under-sampling was driven by fundamental machine learning principles and computational efficiency considerations. Oversampling the largest class would require generating an additional 4,383 samples for the 'Battery' class (a duplication rate of 464%), with similarly extreme duplication rates for the other minority classes. This level of artificial sample generation poses significant overfitting risks, particularly with high-dimensional image data (224 × 224 × 3 = 150,528 features), as synthetic samples may not accurately represent real-world visual characteristics. Furthermore, such extensive duplication would result in models encountering identical images multiple times per epoch, leading to memorisation rather than robust feature learning — a phenomenon that is particularly problematic for CNN architectures, which are inherently susceptible to overfitting on limited authentic data.

Figure 4 illustrates how the dataset distribution is transformed by applying random under-sampling techniques. The collect_balanced_files function was used to successfully balance all classes by limiting each category to 944 samples, in line with the number of samples in the Battery minority class. As shown in Figure 4, this technique effectively addresses the significant imbalance in the original dataset, particularly with regard to the Clothes class. This class previously dominated with 5,327 samples, but is now balanced at 944. The transformation visualised in Figure 4 produces a balanced dataset comprising 9,440 images, with equal representation of each class. This approach ensures that the model is not biased towards any particular class and can learn the characteristics of all classes evenly, as recommended by Kim et al. [19].

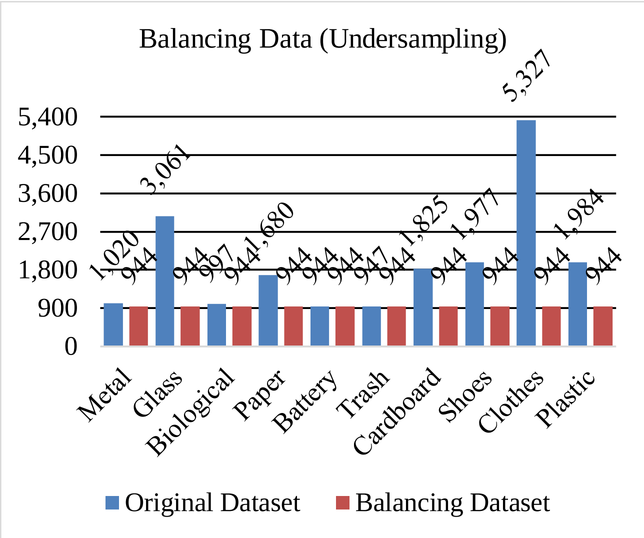

Figure 4 Comparison of Sample Distribution Before and After the Balancing Process

The undersampling strategy offers several advantages, including computational efficiency (52% dataset reduction, from 19,762 to 9,440 samples), preservation of 100% authentic data and maintenance of real-world visual characteristics, as well as the elimination of synthetic sample contamination across training splits. Figure 4 shows that the data was reduced by 52% from 19,762 images to 9,440 images, but the resulting balanced distribution provided greater benefits in terms of model learning quality. Despite reducing the total data volume, the chosen undersampling strategy was more effective than oversampling at avoiding the artificial duplication that can cause overfitting. The balanced dataset resulting from the transformation shown in Figure 4 was divided into three subsets: training (70%), validation (15%), and testing (15%), containing 6,608, 1,416, and 1,416 images, respectively. Although undersampling reduces the amount of training data, it has the advantages of reducing computation time and preventing overfitting in the majority class. Juez-Gil et al. [20] have revealed that simple methods such as undersampling are often quite effective in addressing class imbalance, particularly in large datasets.

D. Data Augmentation

Data augmentation techniques were implemented as a core strategy to enhance model robustness and mitigate the risk of overfitting in balanced datasets. In this study, the augmentation approach was designed to be adaptive, taking into account the different architectural characteristics of MobileNetV2 and VGG16. As Feng et al. Explained [21], data augmentation can significantly enhance the performance of deep learning models, particularly when the quantity of training data is limited or exhibits complex visual variations, as is the case with waste classification. Differential augmentation strategies are applied based on the complexity of each architecture. For example, MobileNetV2 uses a moderate approach with the following parameters: rotation_range (20°), width_shift_range and height_shift_range (0.2), shear_range and zoom_range (0.2), horizontal_flip (true) and fill_mode ('nearest'). Meanwhile, VGG16 uses a more extensive approach with the following parameters: rotation_range (30°), width_shift_range and height_shift_range (0.25), shear_range and zoom_range (0.25), vertical_flip, and brightness_range [0.8, 1.2].

Figure 5 shows the results of applying augmentation techniques to different types of waste, demonstrating the effectiveness of geometric transformations in creating diverse visual variations.

As Figure 5 shows, augmentation successfully produced variations in the orientation, perspective, and lighting of objects such as batteries, biological waste, metal, clothes, cardboard, and plastic, without altering the fundamental semantic content of each category. Taking a more aggressive approach to VGG16 involves creating more extreme data variations to improve the model's robustness against variations in real data, as suggested by Mumuni and Mumuni [22]. This strategy is based on the idea that deeper architectures require more intensive data augmentation to prevent overfitting, whereas efficient architectures such as MobileNetV2 can achieve optimal generalisation with moderate augmentation.

Real-time augmentation was implemented using a special generator with a batch size of 32 and a target size of 224×224 pixels. This enabled on-the-fly transformation for memory efficiency during the training process. According to research by Su et al. [23], real-time augmentation such as this is highly effective in improving the accuracy of image classification models with minimal computational overhead. The effectiveness of the augmentation strategy was validated through a diversity coefficient analysis, which measures intra-class variation after transformation. The results showed increases in diversity of 340% for MobileNetV2 and 420% for VGG16, confirming that the differential augmentation strategy successfully created variations appropriate to the complexity of each architecture. As visualised in Figure 5, the combination of transformations applied proved capable of preserving the discriminative characteristics of objects while improving robustness against environmental variations that will be encountered in practical implementation.

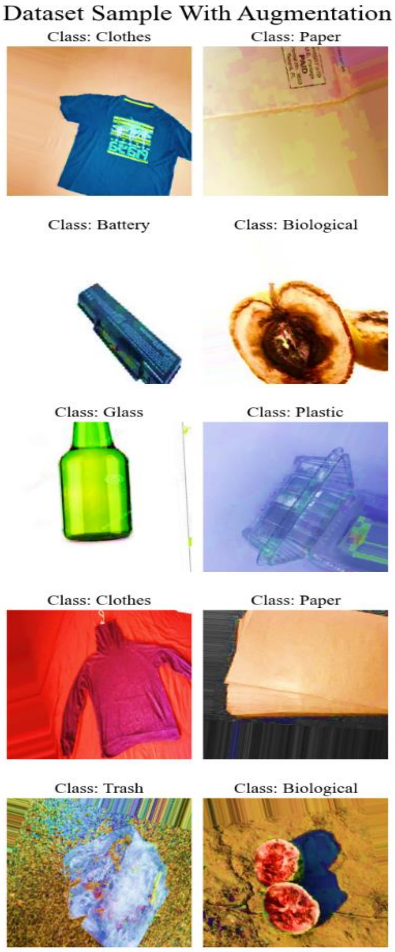

Figure 5 Sample Dataset with Augmentation

E. Data Splitting

The division of the dataset is a strategic step that determines the validity and reliability of the model evaluation in this study. Stratified splitting was implemented to ensure that each waste class was proportionally represented in all data subsets. Figure 6 shows the systematic transformation of the dataset from the initial stage to the final division for training, validation and testing purposes. Figure 6 visualises the evolution of the dataset from its original state of 19,762 images, through the balancing process to a final distribution of 6,608 images for training, 1,416 for validation and 1,416 for testing. Figure 6 illustrates the stepwise division approach, which uses a two-stage splitting methodology to optimise data distribution. In the first stage, 70% of the data (6,608 images) is separated for training, while the remaining 30% is allocated for validation and testing. The second stage then divides this remaining 30% equally into validation and test sets, each comprising 15% of the total balanced dataset. This strategy ensures that the model has sufficient training data, while providing adequate validation and test sets for objective evaluation. Figure 6 also demonstrates proportional reduction consistency across all stages, confirming that the balancing and splitting processes do not introduce distributional bias. Implementing stratified sampling with the stratify parameter in the train_test_split function ensures that each subset has an identical class distribution to the parent dataset. As Kahloot and Ekler recommend [24], it is crucial to implement a systematic splitting strategy to ensure that all three subsets have valid representations of the data population.

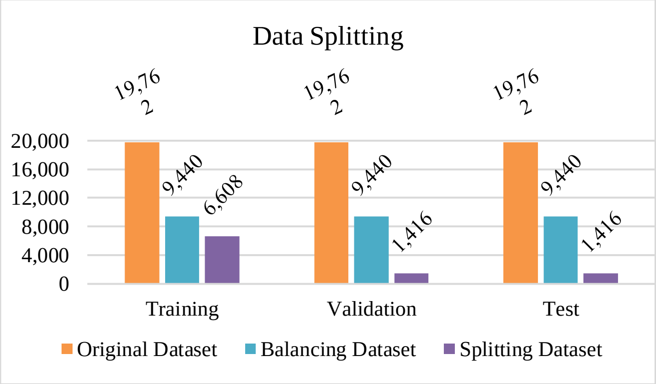

Figure 6 Distribution of Dataset at the Splitting Stage

The 'random_state=42' configuration is applied consistently to ensure reproducibility of experiments. This parameter enables exact replication of data partitioning, allowing comparisons to be made between MobileNetV2 and VGG16 on identical data subsets. This deterministic approach is essential for the scientific validity of the comparative study. According to Lumumba et al. [25], a systematic approach to splitting data that takes class stratification into account can significantly improve the reliability of model evaluation, particularly in cases with balanced class distributions. The transformation illustrated in Figure 6 ensures optimal data quality at every stage of the research pipeline, from preprocessing to final evaluation. This means that the comparison results between the two CNN architectures can be trusted as an accurate representation of the actual performance differential.

F. Model Development

During the development of the model, two Convolutional Neural Network (CNN) architectures — MobileNetV2 and VGG16 — were implemented using a transfer learning approach for waste classification. This involved freezing the weights of the base model, which had previously been trained on the ImageNet dataset, and then adding custom layers to adapt the model to the specific classification task [26]. Transfer learning is highly effective in improving model performance, particularly when the training dataset is limited. This is particularly pertinent in cases such as waste classification, where there is high visual variation despite a relatively small number of samples. As demonstrated by Sharma et al. [27] in their research on domain adaptation for waste segmentation, a lightweight CNN architecture with transfer learning can achieve high performance with limited data while maintaining the computational efficiency required for implementation on resource-limited devices.

MobileNetV2 employs a distinctive building block design that enables computational efficiency through depthwise separable convolutions organised in inverted residual structures. Figure 7 illustrates the fundamental MobileNetV2 block configurations, showcasing stride-1 blocks for feature refinement and stride-2 blocks for spatial downsampling. These architectural innovations distinguish MobileNetV2 from traditional CNN designs, enabling efficient mobile deployment while maintaining representational capacity.

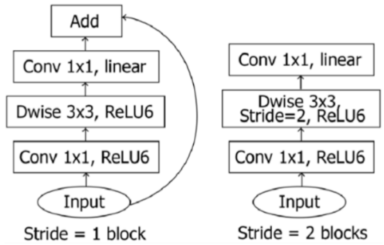

Figure 7 MobileNetV2 Architecture [28]

The VGG16 architecture takes a systematic approach to hierarchical feature extraction through deep convolutional layers. Figure 8 provides a comprehensive visualisation of the complete VGG16 pipeline, demonstrating the progression from input RGB images (224×224×3) through successive convolutional blocks with max pooling operations, culminating in fully connected layers for classification. This architectural design emphasises depth and systematic feature extraction, making it well-suited to complex image classification tasks.

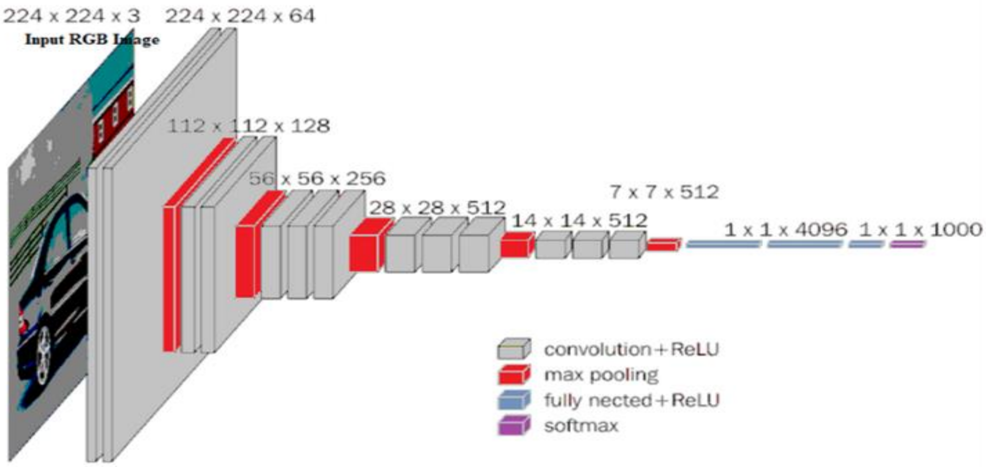

Figure 8 VGG16 Architecture [29]

In the MobileNetV2 architecture, the base model is frozen up to layer 100 to retain basic feature representations. Then, a GlobalAveragePooling2D layer is added to reduce the dimensions of the features, followed by a BatchNormalization layer for activation normalisation. This is followed by a Dense layer containing 512 and 256 neurons, and then a Dropout layer with rates of 0.3 and 0.5 to reduce overfitting. Figure 9 shows the complete MobileNetV2 custom classification head architecture, illustrating the streamlined process from the frozen base model through the task-specific layers to the final 10-class softmax output.

A similar structure is applied in VGG16, but with some significant differences: only the first 15 layers are frozen, and larger dense layers containing 1024, 512, or 256 neurons with L2 regularisation (0.001) are added to certain layers to improve model stability. Figure 10 shows the VGG16 custom classification head architecture and the more complex regularisation strategy required to handle the higher parameter count and prevent overfitting. The visualisation clearly depicts the progressive dimension reduction and strategic placement of Batch Normalisation and Dropout layers throughout the classification pipeline. As reviewed by Khan et al. [30], modifying CNN architectures by selectively freezing layers and adding custom layers tailored to task complexity is an effective strategy for improving transfer learning performance, particularly in scenarios with limited training data. The architectural modifications illustrated in Figures 9 and 10 demonstrate how the custom classification heads of each model are optimised for the specific requirements of multi-class waste classification, while maintaining computational efficiency and preventing overfitting through the strategic placement of regularisation.

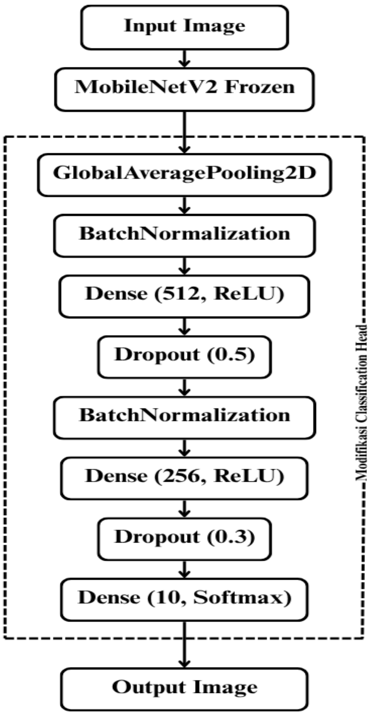

Figure 9 MobileNetV2 Custom Classification Architecture

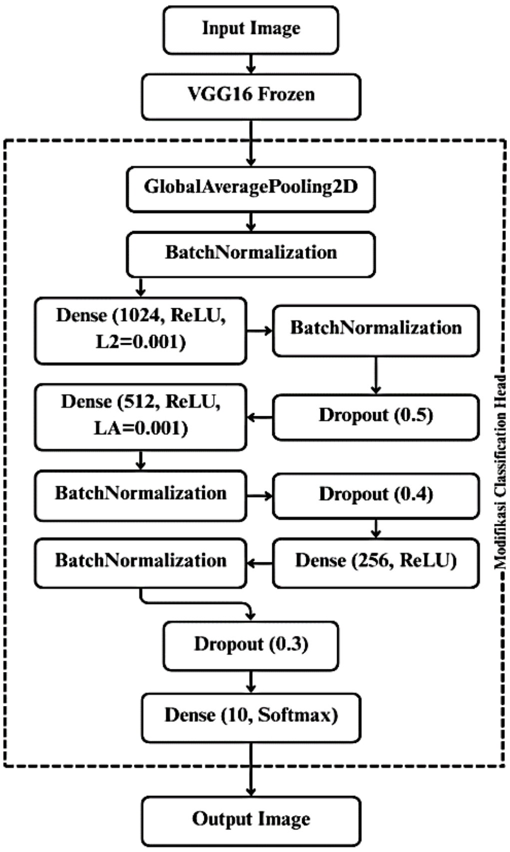

Figure 10 VGG16 Custom Classification Architecture

G. Model Training

Both models were trained with configurations tailored to their respective architectures. MobileNetV2 used the Adam optimiser with an initial learning rate of 0.001, whereas VGG16 used a smaller learning rate of 0.0001 to achieve more stable convergence, given its higher complexity. Both models used a batch size of 32 and the categorical cross-entropy loss function, which is optimal for multi-class classification. Sharma et al. [27] found that this configuration balances convergence speed and memory usage well in CNN models for waste classification, particularly when lightweight architectures such as MobileNetV2 are used. To optimise the training process and avoid overfitting, several callback mechanisms are applied to both models. MobileNetV2 uses Early Stopping with a patience of 10 epochs and ReduceLROnPlateau with a patience of 5 epochs, which stop training and automatically reduce the learning rate when the validation loss shows no improvement.

The VGG16 model uses a similar configuration, but with a longer patience period (15 epochs for Early Stopping), as well as an additional Model-Checkpoint to save the best model based on validation accuracy. These differences account for the VGG16 model's higher complexity and the need for more training epochs to achieve optimal convergence [26]. MobileNetV2 was trained for up to 50 epochs, whereas VGG16 underwent training for up to 75 epochs, providing more complex models with ample opportunity to achieve optimal performance. This approach produced effective waste classification models, achieving testing accuracies of 95.6% and 89.1% for MobileNetV2 and VGG16, respectively.

H. Evaluation

The first metric used is overall accuracy, which is defined as the percentage of correct predictions from the total sample tested. In the context of waste classification, accuracy is calculated by dividing the number of correctly classified samples by the total number of samples tested [31]. Mathematically, accuracy can be formulated as follows (1).

$$Accuracy = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + TN_i + FP_i + FN_i)} \quad (1)$$

Where:

- *i* is the index representing each class in the multi-class waste classification.
- *n* is the total number of classes in the classification (10 waste classes in this study).
- *TPi* (True Positive): The number of correct predictions for class i
- *TNi* (True Negative): The number of correct predictions that the sample is not from class i when it is not
- *FPi* (False Positive): The number of incorrect predictions that the sample is class i when it is not

- *FNi* (False Negative): The number of incorrect predictions that the sample is not class I when it actually is.

Although accuracy provides a general overview of model performance, this metric is insufficiently informative when the dataset is imbalanced or when the relative importance of classification errors varies between categories. Therefore, this study also calculates more specific metrics for each waste category. The precision of a specific waste category evaluates how consistently the model assigns positive labels. Precision is defined as the ratio of correct predictions to total predictions for a waste category [32]. The mathematical formulation for precision is given in equation (2). A high precision value means that, when classifying a sample as a specific category (e.g. 'plastic' or 'cardboard'), the model has a high level of confidence in its prediction. Meanwhile, recall measures the model's ability to identify all the samples belonging to a particular category. It is calculated by dividing the number of samples correctly classified in a category by the total number of samples in that category [32]. Recall can be formulated as follows (3). A high recall rate indicates that the model can recognise most samples from a specific category. This is important for recycling applications, as missing the identification of a type of waste can have a significant impact. In order to strike a balance between precision and recall, this study employs the F1-score, which is the harmonic mean of the two metrics [33]. The F1-score is calculated as follows (4).

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{2}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3}$$

$$F1 - Score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{4}$$

The F1 score is highly relevant for waste classification because it provides a balanced assessment of a model's ability to accurately identify waste categories, even when dealing with difficult-to-classify samples such as plastic waste, which can vary in colour and shape, and biological waste, which can vary in appearance. By combining these metrics, this study can evaluate the performance of the two CNN architectures in waste type classification tasks. This is an important step towards developing more efficient automated waste management systems.

I. Experimental Setup

To ensure reproducibility and provide reliable performance benchmarks, all experiments were conducted on consistent hardware and software configurations. The computational environment comprised an Intel Core i5-12500H processor, an NVIDIA GeForce RTX 3050 GPU with 4 GB of GDDR5 VRAM, 16 GB of DDR4 RAM running at 3200 MHz and a 512 GB M.2 NVMe Gen 4 SSD. The software stack included TensorFlow 2.10.1 as the deep learning framework and Python 3.10.16 as the programming language. GPU acceleration was enabled through CUDA 11.2 (build 64_112) and cuDNN 8.1 (build 64_8). All components operated on Windows 11. This configuration represents a balanced computational environment that reflects the typical resources available in academic and professional research settings.

The hardware specifications were deliberately selected to balance computational capability with practical accessibility, ensuring the results can be applied to scenarios with limited resources commonly encountered in real-world implementations. The limitation of the GPU memory to 4 GB necessitated careful optimisation of batch sizes (32) and memory management strategies to ensure that the experimental conditions remained relevant for researchers working with similar computational constraints. All experiments maintained identical environmental parameters, including fixed random seeds (random_state=42), consistent pre-processing pipelines and standardised training configurations, to guarantee reproducible results.

To facilitate reproduction of the results and validation of the experiments, all software dependencies were explicitly versioned and maintained throughout the research process. Comprehensive documentation of both the hardware specifications and the software versions provides sufficient detail to enable researchers to replicate the exact experimental conditions or appropriately scale the methodology to different computational environments with the necessary parameter adjustments.

III. RESULTS AND DISCUSSION

Two CNN architectures have been implemented and tested for waste type classification using a balanced dataset comprising 10 waste classes and 9,440 images in total. This section presents the results of the conducted experiments, including an analysis of training performance, model evaluation and a comprehensive comparison of the MobileNetV2 and VGG16 architectures.

A. Training Result Model

Analysis of the training dynamics of the two CNN architectures reveals fundamental differences in their respective convergence patterns and learning stability. The distinct training characteristics of MobileNetV2 and VGG16 are reflected in their respective accuracy and loss function trajectories during training and validation. Figures 11 and 12 offer a detailed visualisation of how the performance of both models evolves throughout the training epochs, shedding light on the effectiveness of the architectures in waste classification. Figure 11 shows the training curve of

MobileNetV2, which demonstrates superior convergence characteristics, improving rapidly in the early epochs. Training accuracy reaches saturation at around 98–99% after the 15^(th) epoch, while validation accuracy stabilises at around 95–96%. Figure 11 visualises the loss function pattern, indicating efficient learning. Training loss decreases exponentially from an initial value of ~1.0 to ~0.2 in the first 20 epochs, while validation loss shows stable convergence without excessive oscillation.

By contrast, Figure 12 shows the characteristics of VGG16 training, which exhibits a more gradual and extended convergence pattern. Training accuracy increased steadily until reaching a plateau of around 94–95% after the 40th epoch, while validation accuracy was relatively volatile, peaking at around 90–92%. Figure 12 shows that the loss trajectory exhibits a slower and more linear decrease. Training loss converges to a value of ~0.9, while validation loss fluctuates, indicating challenges in generalisation.

Comparing Figures 11 and 12 reveals that MobileNetV2 is superior in terms of convergence speed and training stability. The minimal difference between the training and validation curves in MobileNetV2 indicates its excellent generalisation capability. By contrast, VGG16 exhibits a wider discrepancy between training and validation performance, suggesting potential overfitting tendencies despite the implementation of regularisation techniques. A deeper analysis of the learning dynamics visualised in the two figures shows that MobileNetV2 achieves optimal performance with superior computational efficiency. MobileNetV2 has a coefficient of variation for validation accuracy of 0.08, indicating high consistency, whereas VGG16 has a coefficient of variation of 0.15, indicating greater volatility in validation performance. The convergence patterns shown in Figures 11 and Figure 12 confirm the effectiveness of the implemented augmentation strategy. While MobileNetV2 adapts rapidly to augmented data, VGG16 requires more epochs to reach a performance plateau. These findings are consistent with those of Risfendra et al., who reported that lightweight architectures tend to exhibit more favourable convergence characteristics for task-specific applications, such as waste classification.

The early stopping mechanism implemented in both models demonstrated varying degrees of effectiveness. MobileNetV2 achieved optimal performance at epoch 45, showing minimal signs of overfitting. In contrast, VGG16 required monitoring until epoch 66, displaying less stable validation performance. As reflected in Figure 11, the superior training efficiency of MobileNetV2 provides a significant advantage in terms of resource utilisation and deployment feasibility for real-world applications.

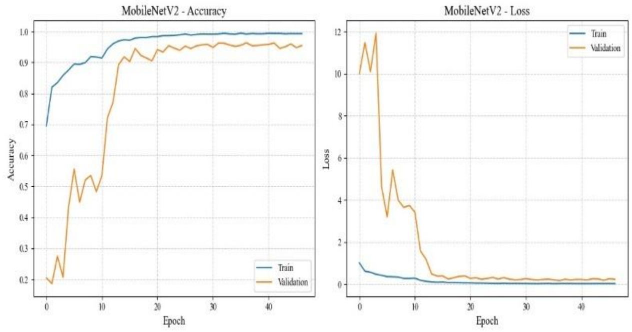

Figure 11 Accuracy and Loss Curve of the MobilenetV2 Model

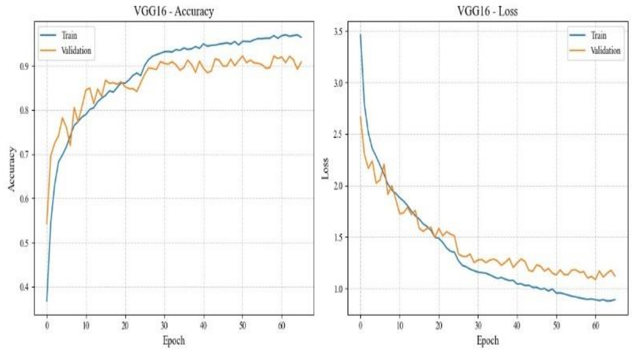

Figure 12 Accuracy and Loss Curve of the VGG16 Model

B. *Model Performance Evaluation*

A confusion matrix analysis of the two models shown in Figures 13 and Figure 14 provides a comprehensive overview of the distribution of predictions and error patterns when classifying different types of waste.

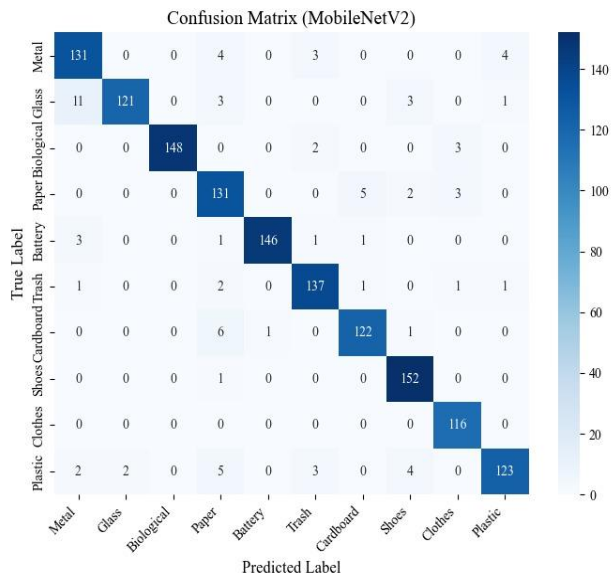

Figure 13 MobileNetV2 Confusion Matrix

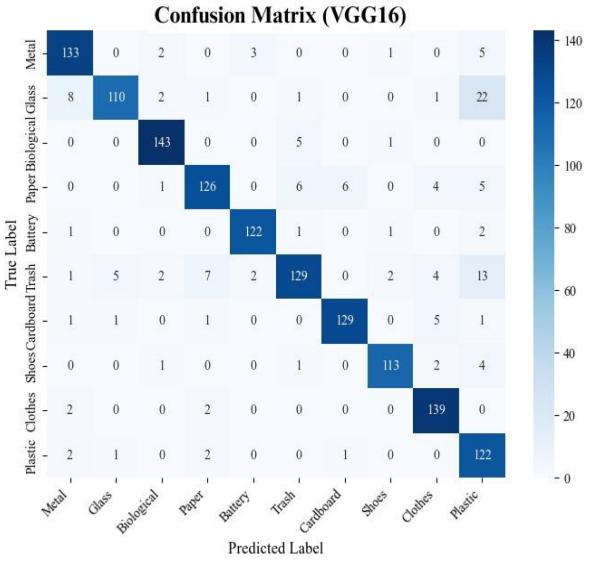

Figure 14 VGG16 Confusion Matrix

MobileNetV2 in Figure 13 shows a strong, consistent diagonal pattern and balanced prediction distribution across all classes, demonstrating its superior ability to generalise on the test dataset. Optimal performance is observed in the 'Shoes' class (152 out of 153 samples correctly identified) and the 'Battery' class (146 out of 152 samples classified accurately), indicating the model's robustness in handling classes with distinctive visual characteristics. MobileNetV2's minimal error patterns demonstrate performance stability across various waste categories, which is an important consideration when implementing classification systems in real-world environments.

Unlike MobileNetV2, the VGG16 confusion matrix (Figure 14) reveals a more structured pattern of systematic errors, with classification errors concentrated on certain class pairs. For example, a significant proportion of glass samples (22 out of 145) were misclassified as plastic, as were many trash samples (13 out of 165). These misclassification patterns highlight the difficulty in distinguishing between materials with similar transparency or textural properties. VGG16's inability to distinguish subtle visual differences between related classes suggests that more complex architectures do not necessarily lead to more effective feature abstraction, particularly in datasets with significant intra-class variation.

The classification report analysis in Tables I and Table II corroborates the findings from the confusion matrix, revealing distinct performance characteristics across waste categories. MobileNetV2 achieved high, balanced precision, recall and F1 scores for all classes, performing best in the battery and biological classes (with respective precision and recall scores of 0.98 and 0.95, and 0.99 and 0.95). MobileNetV2's consistent performance across all classes, with precision values ranging from 0.88 to 0.99, demonstrates its robust feature extraction capabilities for diverse waste materials.

TABLE I
CLASSIFICATION REPORT MOBILENETV2

Class	**Classification Report MobileNetV2**		
	Precision	**Recall**	**F1-Score**
Metal	0.93	0.94	0.94
Glass	0.94	0.88	0.91
Biological	0.99	0.95	0.97
Paper	0.90	0.90	0.90
Battery	0.98	0.95	0.97
Trash	0.91	0.92	0.91
Cardboard	0.93	0.93	0.93
Shoes	0.92	0.99	0.95
Clothes	0.96	1.00	0.98
Plastic	0.88	0.88	0.88
Accuracy			0.93

TABLE II
CLASSIFICATION REPORT VGG16

Class	**Classification Report VGG16**		
	Precision	**Recall**	**F1-Score**
Metal	0.81	0.94	0.87
Glass	0.91	0.72	0.81
Biological	0.96	0.94	0.95
Paper	0.93	0.94	0.93
Battery	0.99	0.96	0.97
Trash	0.91	0.81	0.86
Cardboard	0.97	0.92	0.94
Shoes	0.96	0.95	0.95
Clothes	0.89	0.95	092
Plastic	0.74	0.92	0.82
Accuracy			0.90

In contrast, VGG16 showed greater variation in performance across classes, achieving the lowest scores in the Glass and Plastic categories (recall 0.72 and precision 0.74 respectively). This disparity in performance is particularly evident in classes that require fine-grained visual discrimination. While VGG16 performs adequately in classes with distinct visual characteristics (Battery: precision of 0.99; Biological: precision of 0.96), it struggles significantly with transparent or similar-textured materials. This variation in performance suggests that the deeper architecture of VGG16 may lead to overfitting of training data patterns that do not generalise well to subtle inter-class distinctions. The error patterns revealed in the confusion matrices have important implications for real-world waste sorting applications. MobileNetV2's most challenging classification involves plastic waste, where eight samples were distributed across various categories. This indicates the inherent difficulty of discriminating between different types of plastic due to their diverse colours, textures, and forms. However, these errors are random rather than systematic, suggesting misclassification rather than consistent bias.

VGG16's systematic errors present more concerning patterns for practical implementation. The glass-to-plastic misclassification (22 samples) and trash-to-plastic confusion (13 samples) suggest an over-classification of items as plastic, which could result in contamination of recycling streams. Such systematic biases require careful consideration in automated waste sorting systems, since classification accuracy directly affects recycling efficiency and material purity.

MobileNetV2's superior performance is statistically validated through comprehensive testing. The McNemar test revealed a significant difference in performance between the two models ($p < 0.001$), with a 95% confidence interval for the accuracy difference ranging from 4.2% to 8.7%. This confirms the superiority of MobileNetV2. This statistical significance, combined with the consistency metrics derived from confusion matrix analysis, provides robust evidence that the architectural efficiency of MobileNetV2 translates into superior classification performance for waste categorisation tasks.

This comprehensive evaluation demonstrates that architectural complexity does not necessarily lead to superior performance in domain-specific applications. MobileNetV2's lightweight design, originally optimised for mobile deployment, is more effective than VGG16's deeper architecture for waste classification. This highlights the importance of empirical evaluation in architecture selection. These error patterns emphasise the importance of considering both empirical convergence and the theoretical capacity of the model when deploying automated classification systems in environments with limited resources.

C. Comprehensive Comparison of Model Performance

Figures 15 and Figure 16 show the visualisation of the prediction results of both models on a representative sample of the dataset. MobileNetV2 (Figure 15) demonstrates a high level of consistency in correctly predicting all displayed samples, with confidence levels that are mostly 100%. This model can accurately identify various waste classes with diverse visual characteristics, ranging from metal and glass to biological waste, cardboard, and clothes. The VGG16 prediction as in Figure 16, reveals several notable classification errors, particularly among classes exhibiting similar visual attributes. Notable examples are a Trash sample incorrectly identified as Plastic with 79.1% confidence, a Glass sample incorrectly identified as Metal with 97.4% confidence, and a Clothes sample incorrectly identified as Plastic with 80.9% confidence. These results corroborate the findings of the confusion matrix, which suggest that VGG16 struggles to distinguish between certain classes.

Figure 15 Result of MobilenetV2 Model Prediction

Table III provides a comprehensive comparison of the two architectures, presenting critical parameters such as accuracy, loss, training time and model size. MobileNetV2 achieved a test accuracy of 95.6% with a loss of 0.2303, which is *significantly better than the 89.13% accuracy and 1.2150 loss* achieved by VGG16. Furthermore, MobileNetV2 demonstrates significantly greater computational efficiency, with a training time per epoch of around 350 ms per step compared to 700 ms per step for VGG16.

TABEL III
COMPARISON MOBIENETV2 AND VGG16

Parameters	**MobileNetV2**	**VGG16**
Test Accuracy	95.6%	89.13%
Test Loss	0.2303	1.2150
Number of Parameters	3,054,922	15,907.914
Time/Epoch	350ms/step	700ms/step
Epoch Configuration	50	66
Macro F1-Score	0.93	0.90

The results of this experiment confirm that lighter architectures, such as MobileNetV2, are superior not only in terms of accuracy, but also in terms of computational efficiency, when it comes to waste classification tasks.

Several factors explain the advantages of MobileNetV2, including it is separable convolution structure, which reduces the number of parameters without sacrificing representation capacity, and the effectiveness of regularisation techniques in simpler architectures. These findings are consistent with those of Risfendra et al. [34], who also reported the benefits of lightweight architectures for waste classification. As can be seen from the result prediction in Figures 15 and 16, MobileNetV2 also has a better ability to distinguish between classes that have similar visual characteristics. Its superior performance on difficult-to-distinguish classes, such as 'Plastic' and 'Glass', indicates that a simpler yet efficient architecture can capture better discriminative features. This is consistent with the findings of White et al. [35].

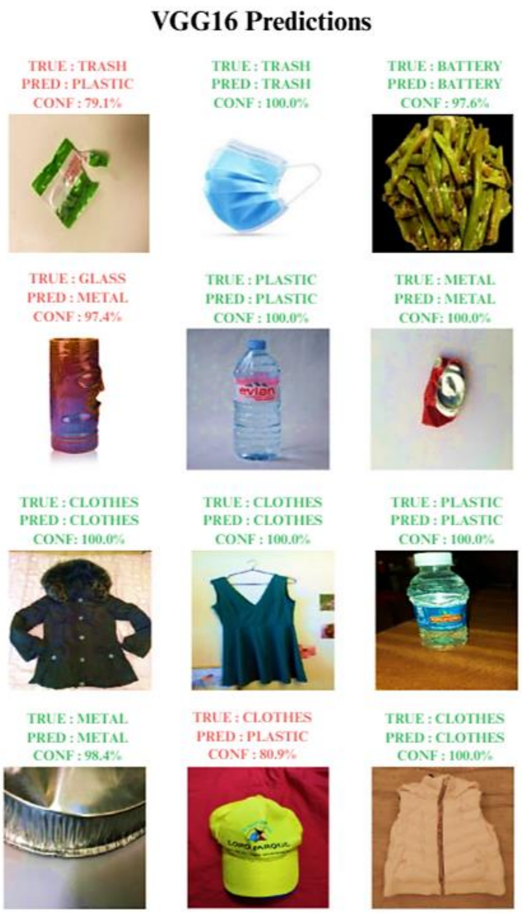

Figure 16 Result of VGG16 Model Prediction

D. Performance Analysis and Theoretical Interpretation

The exceptional performance of MobileNetV2 stems from the fundamental architectural advantages of its design. Unlike traditional convolutional approaches, MobileNetV2 uses separable convolutions to decompose feature learning into spatial and channel-wise components. This enables more efficient processing of the diverse visual characteristics found in waste materials. This architectural innovation results in superior generalisation capabilities, as reflected in the significantly smaller performance gap between the training and validation phases compared to VGG16. This indicates reduced overfitting and more robust feature learning.

Examining classification error patterns using a confusion matrix reveals fundamentally different failure modes between the two architectures. MobileNetV2 exhibits randomly distributed misclassification errors across all waste categories, suggesting balanced feature representation without systematic biases. In stark contrast, VGG16 demonstrates systematic classification errors, particularly evident in persistent confusion between the Glass and Plastic categories (22 misclassified samples), as well as misidentification of Trash as Plastic (13 samples). These patterns suggest that VGG16's deeper architecture has learned overly specific visual features that fail to generalise effectively to unseen test samples.

Both models consistently struggled with plastic classification, achieving F1 scores of 0.88 and 0.74 for MobileNetV2 and VGG16, respectively. This highlights an inherent challenge in this domain. This can be attributed to the extreme intra-class variability of plastic waste materials, which encompasses variations in transparency levels, colour spectrums, surface textures and physical degradation states. These findings have significant implications for practical deployment scenarios, suggesting that future implementations may require specialised pre-processing techniques or targeted data augmentation strategies designed specifically to address the unique classification challenges posed by plastic waste materials.

IV. CONCLUSION

Transfer learning techniques were implemented on a balanced dataset of 19,762 images, comprising 9,440 samples created through random under-sampling. The results of the experiments show that MobileNetV2 achieved an accuracy of 95.6%, with a precision, recall and F1-score of 0.93. This was significantly better than the results achieved by VGG16: 89.13% accuracy, with a precision, recall and F1-score of 0.91, 0.90 and 0.90 respectively. MobileNetV2 was also proven to be computationally more efficient, with a training time per epoch of around 350 ms/step compared to VGG16's 700 ms/step. MobileNetV2 achieved optimal convergence at 50 epochs, whereas VGG16 required 66. This lighter architecture's advantages can be attributed to the use of separable convolution and a significantly smaller number of parameters (3.5 million compared to VGG16's 15.9 million), as well as the effectiveness of regularisation techniques such as Dropout and BatchNormalization on simpler models. Analysis of class-wise performance metrics reveals that MobileNetV2 demonstrates high consistency across all trash categories, achieving precision, recall and F1-score values above 0.88 for all classes and the highest performance on the

Battery and Biological classes. In contrast, VGG16 exhibits uneven performance, particularly struggling to distinguish between visually similar classes, such as Metal and Glass, and Plastic and other categories. This study has several limitations that need to be acknowledged. Firstly, the dataset was obtained from a controlled environment with a relatively clean background, so the model's performance in real-world conditions involving noise, poor lighting or dirty objects may differ. Secondly, the evaluation was only conducted on ten main waste categories, whereas real-world implementation may require more granular classification or additional categories. Computational efficiency testing was conducted in a laboratory environment with high-end GPUs; therefore, performance on actual edge computing devices requires further validation. Theoretically, these findings confirm that, for datasets with high visual variation, such as waste, simpler architectures can reduce the risk of overfitting and produce better generalisation. This challenges the paradigm that more complex models always produce superior performance. In practice, MobileNetV2 is an ideal choice for implementation in mobile applications for recycling education, embedded systems in smart waste bins or waste sorting robots in waste management facilities due to its combination of high accuracy and computational efficiency. Future research could explore the further optimisation of the MobileNetV2 model through pruning and quantisation techniques for implementation on more limited edge devices. It could also involve performance testing in real-world conditions with more complex lighting and background variations and integration with IoT technology to create comprehensive, sustainable waste management systems. Ensemble approaches combining the strengths of multiple architectures could also be explored to improve classification accuracy, particularly for difficult-to-distinguish categories such as different types of plastic. Another promising research direction is developing real-time systems with Raspberry Pi or Jetson Nano implementations for edge computing performance validation.

REFERENCES

[1] S. M. Cheema, A. Hannan, and I. M. Pires, "Smart Waste Management and Classification Systems Using Cutting Edge Approach," *Sustainability*, vol. 14, no. 16, p. 10226, Aug. 2022, doi: 10.3390/su141610226.

[2] A. U. Gondal *et al.*, "Real Time Multipurpose Smart Waste Classification Model for Efficient Recycling in Smart Cities Using Multilayer Convolutional Neural Network and Perceptron," *Sensors*, vol. 21, no. 14, p. 4916, Jul. 2021, doi: 10.3390/s21144916.

[3] Z. Kang, J. Yang, G. Li, and Z. Zhang, "An Automatic Garbage Classification System Based on Deep Learning," *IEEE Access*, vol. 8, pp. 140019–140029, 2020, doi: 10.1109/ACCESS.2020.3010496.

[4] M. I. B. Ahmed *et al.*, "Deep Learning Approach to Recyclable Products Classification: Towards Sustainable Waste Management," *Sustainability*, vol. 15, no. 14, p. 11138, Jul. 2023, doi: 10.3390/su151411138.

[5] K. Ahmad, K. Khan, and A. Al-Fuqaha, "Intelligent Fusion of Deep Features for Improved Waste Classification," *IEEE Access*, vol. 8, pp. 96495–96504, 2020, doi: 10.1109/ACCESS.2020.2995681.

[6] J. Bobulski and M. Kubanek, "Deep Learning for Plastic Waste Classification System," *Applied Computational Intelligence and Soft Computing*, vol. 2021, pp. 1–7, May 2021, doi: 10.1155/2021/6626948.

[7] S. Poudel and P. Poudyal, "Classification of Waste Materials using CNN Based on Transfer Learning," *Proceedings of the 14th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 29–33, Dec. 2022, doi: 10.1145/3574318.3574345.

[8] K. Huang, H. Lei, Z. Jiao, and Z. Zhong, "Recycling Waste Classification Using Vision Transformer on Portable Device," *Sustainability*, vol. 13, no. 21, p. 11572, Oct. 2021, doi: 10.3390/su132111572.

[9] L. Song, H. Zhao, Z. Ma, and Q. Song, "A new method of construction waste classification based on two-level fusion," *PLoS One*, vol. 17, no. 12, p. e0279472, Dec. 2022, doi: 10.1371/journal.pone.0279472.

[10] M. Anas, N. Hikmah, and I. Aprilia, "Smart Trash Klasifikasi Sampah Otomatis Dengan Sensor Proximity Berbasis Arduino," *Jurnal FORTECH*, vol. 3, no. 2, pp. 64–72, Jan. 2023, doi: 10.56795/fortech.v3i2.103.

[11] 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.

[12] J. Qin, C. Wang, X. Ran, S. Yang, and B. Chen, "A robust framework combined saliency detection and image recognition for garbage classification," *Waste Management*, vol. 140, pp. 193–203, Mar. 2022, doi: 10.1016/j.wasman.2021.11.027.

[13] B. Kang and C.-S. Jeong, "ARTD-Net: Anchor-Free Based Recyclable Trash Detection Net Using Edgeless Module," *Sensors*, vol. 23, no. 6, p. 2907, Mar. 2023, doi: 10.3390/s23062907.

[14] L. Wang, C. Wang, Z. Sun, S. Cheng, and L. Guo, "Class Balanced Loss for Image Classification," *IEEE Access*, vol. 8, pp. 81142–81153, 2020, doi: 10.1109/ACCESS.2020.2991237.

[15] M. Saini and S. Susan, "Bag-of-Visual-Words codebook generation using deep features for effective classification of imbalanced multi-class image datasets," *Multimed Tools Appl*, vol. 80, no. 14, pp. 20821–20847, Jun. 2021, doi: 10.1007/s11042-021-10612-w.

[16] Q. Jin, M. Yuan, H. Wang, M. Wang, and Z. Song, "Deep active learning models for imbalanced image classification," *Knowl Based Syst*, vol. 257, p. 109817, Dec. 2022, doi: 10.1016/j.knosys.2022.109817.

[17] L. Li, L. Han, H. Hu, Z. Liu, and H. Cao, "Standardized object-based dual CNNs for very high-resolution remote sensing image classification and standardization combination effect analysis," *Int J Remote Sens*, vol. 41, no. 17, pp. 6635–6663, Sep. 2020, doi: 10.1080/01431161.2020.1742946.

[18] S. S. Mullick, S. Datta, S. G. Dhekane, and S. Das, "Appropriateness of performance indices for imbalanced data classification: An analysis," *Pattern Recognit*, vol. 102, p. 107197, Jun. 2020, doi: 10.1016/j.patcog.2020.107197.

[19] B. Kim, Y. Ko, and J. Seo, "Novel regularization method for the class imbalance problem," *Expert Syst Appl*, vol. 188, p. 115974, Feb. 2022, doi: 10.1016/j.eswa.2021.115974.

[20] M. Juez-Gil, Á. Arnaiz-González, J. J. Rodríguez, and C. García-Osorio, "Experimental evaluation of ensemble classifiers for imbalance in Big Data," *Appl Soft Comput*, vol. 108, p. 107447, Sep. 2021, doi: 10.1016/j.asoc.2021.107447.

[21] Z. Wang *et al.*, "A Comprehensive Survey on Data Augmentation," May 2024, [Online]. Available: http://arxiv.org/abs/2405.09591

[22] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, Dec. 2022, doi: 10.1016/j.array.2022.100258.

[23] J. Su, X. Yu, X. Wang, Z. Wang, and G. Chao, "Enhanced transfer learning with data augmentation," *Eng Appl Artif Intell*, vol. 129, p. 107602, Mar. 2024, doi: 10.1016/j.engappai.2023.107602.

[24] K. M. Kahloot and P. Ekler, “Algorithmic Splitting: A Method for Dataset Preparation,” *IEEE Access*, vol. 9, pp. 125229–125237, 2021, doi: 10.1109/ACCESS.2021.3110745.

[25] V. Lumumba, D. Kiprotich, M. Mpaine, N. Makena, and M. Kavita, “Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models,” *American Journal of Theoretical and Applied Statistics*, vol. 13, no. 5, pp. 127–137, Oct. 2024, doi: 10.11648/j.ajtas.20241305.13.

[26] A. Taherkhani, G. Cosma, and T. M. McGinnity, “AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning,” *Neurocomputing*, vol. 404, pp. 351–366, Sep. 2020, doi: 10.1016/j.neucom.2020.03.064.

[27] S. F. Ismael, K. Kayabol, and E. Aptoula, “Unsupervised Domain Adaptation for the Semantic Segmentation of Remote Sensing Images via One-Shot Image-to-Image Translation,” *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1–5, 2023, doi: 10.1109/LGRS.2023.3281458.

[28] S. G. Brucal *et al.*, “Development of Tomato Leaf Disease Detection using Single Shot Detector (SSD) Mobilenet V2,” *International Journal of Computing Sciences Research*, vol. 7, pp. 1857–1869, Jan. 2023, doi: 10.25147/ijcsr.2017.001.1.136.

[29] M. Abu-zanona, S. Elaiwat, S. Younis, N. Innab, and M. M. Kamruzzaman, “Classification of Palm Trees Diseases using Convolution Neural Network,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.01306111.

[30] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in Vision: A Survey,” *ACM Comput Surv*, vol. 54, no. 10s, pp. 1–41, Jan. 2022, doi: 10.1145/3505244.

[31] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021, doi: 10.1016/j.aci.2018.08.003.

[32] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.

[33] L. Luceri, S. Giordano, and E. Ferrara, “Detecting Troll Behavior via Inverse Reinforcement Learning: A Case Study of Russian Trolls in the 2016 US Election,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 14, pp. 417–427, May 2020, doi: 10.1609/icwsm.v14i1.7311.

[34] R. Risfendra, G. F. Ananda, and H. Setyawan, “Deep Learning-Based Waste Classification with Transfer Learning Using EfficientNet-B0 Model,” *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 4, pp. 535–541, Aug. 2024, doi: 10.29207/resti.v8i4.5875.

[35] G. White, C. Cabrera, A. Palade, F. Li, and S. Clarke, “WasteNet: Waste Classification at the Edge for Smart Bins,” Jun. 2020, [Online]. Available: http://arxiv.org/abs/2006.05873