Applied Environmental Research
- RESEARCH FOR SUSTAINABLE PLANET -

Research Article

Multi-class Waste Classification Using Convolutional Neural Network

Yanfahmi Rayhan, Achmad Pratama Rifai*

Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia

*Correspondence Email: achmad.p.rifai@ugm.ac.id

Abstract

This study explores a method of waste classification using deep learning, specifically employing the Convolutional Neural Network (CNN). This research involves the creation of a unique dataset, a hybrid of publicly accessible data and a newly compiled collection of images across 13 waste classes: paper, glass, wood, metal, clothes, PCB e-waste, non-PCB e-waste, PET, HDPE, LDPE, PP, PVC, and PS. The development of the CNN model was approached in two ways: transfer learning and full learning. In the transfer learning approach, two pre-trained models, MobileNetV2 and DenseNet121, were utilized. While in the full learning approach, the architecture is constructed using the sequential method. The experimental results indicated that the DenseNet121 transfer learning model outperformed others, achieving an impressive accuracy of 95.2% and an average F-1 score of 0.95 on test data. This was closely followed by the MobileNetV2 transfer learning model, which attained an accuracy of 92% and an average F-1 score of 0.92. In comparison, the full learning model reached an accuracy of 65% and an average F-1 score of 0.65. Generally, transfer learning models yielded more optimal results than those full learning model. This efficiency can be attributed to the pre-existing knowledge in the transfer learning models, which eliminates the need to learn input patterns from the ground up. However, it's important to note that the dataset size of 4586 images across 13 classes may not be sufficient for developing a robust machine learning model from scratch.

ARTICLE HISTORY

Received: 25 Jan. 2024
Accepted: 2 Apr. 2024
Published: 25 Jun. 2024

KEYWORDS

Solid waste;
Deep learning;
Convolutional neural network;
Classification;
Transfer learning;
Full learning

Introduction

Every human activity is likely to generate waste. With economic development, high urbanization rates, and improved living standards, the amount of waste is also rapidly increasing. According to data from the National Waste Management Information System (SIPSN) of the Indonesian Ministry of Environment and Forestry, in 2022, Indonesia produced approximately 36.4 million tons of waste annually [1]. This figure marked an increase from the previous year, where the waste generation was around 28.5 million tons. The rise in waste production correlates with the growing population of Indonesia each year. Figure 1 presents the composition of waste based on types in Indonesia in 2022.

The most significant types of waste produced include food remnants (40.54%), wood/branches (13.09%), paper /cardboard (11.29%), plastic (17.89%), metal (3.08%), and fabric (2.6%). However, this study will focus on detecting waste that still holds value and can be recycled or repurposed. The detection of those types of waste is important, because concentrating on sorting waste that can be recycled or repurposed is integral to developing a sustainable, economically viable, and environmentally responsible approach to waste management. This focus not only addresses immediate waste disposal challenges but also contributes to the broader goals of resource conservation and sustainable development.

https://doi.org/10.35762/AER.2024021

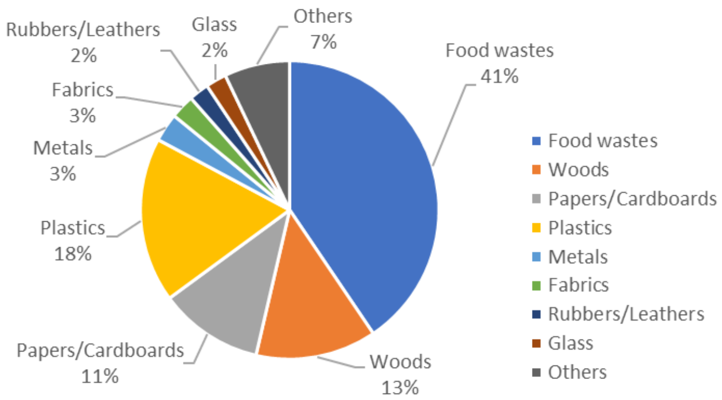


Figure 1 Waste composition in Indonesia in 2022 [1].

From the types of waste mentioned earlier, plastic waste, being the second largest in volume, deserves special attention due to its alarming rate of growth. By 2050, the global volume of plastic waste is projected to reach 1.1 billion tons [2]. Consequently, appropriate handling of plastic waste according to its type is necessary. Plastics come in several types, each with unique characteristics and common uses:

- Polyethylene Terephthalate (PET): This lightweight, durable, disposable plastic should not be used with hot liquids. It's typically used for plastic drink bottles, cooking oil packaging, and food wrappers.
- High-Density Polyethylene (HDPE): A strong, rigid plastic not suitable for reuse. Common uses include liquid soap and shampoo bottles.
- Polyvinyl Chloride (PVC): A hard plastic, PVC is not recommended for food and beverage packaging due to chemicals that can disrupt the digestive system. It is commonly used in electronic cable coatings and pipes.
- Low-Density Polyethylene (LDPE): This type of plastic is low-density, flexible, and transparent, often used for plastic bags, sauce bottles, etc.
- Polypropylene (PP): Easily mouldable at high temperatures, flexible, hard, and oil-resistant, PP is commonly used for plastic furniture, food packaging containers, and straws.
- Polystyrene (PS), commonly known as Styrofoam, is easily moldable at high temperatures, lightweight and rigid at room temperature. It is typically used for toys and Styrofoam food containers.

Each type of plastic has different properties such as density, durability, and heat resistance, making them suitable for various applications [3]. Therefore, proper identification and segregation based on these characteristics are crucial for effective recycling and waste management strategies.

Electronic waste (e-waste) presents a significant environmental challenge that needs urgent attention. It can be broadly divided into two categories: items with and without Printed Circuit Boards (PCBs). PCBs are essential components found in numerous electronic devices, including smartphones and computers. They are constructed from insulating materials like fiberglass or plastic and are designed with conductive pathways on the surface to facilitate the flow of electricity and connect various electronic components. The array of e-waste encompasses a wide range of items. Among those containing PCBs are communication and IT devices such as smartphones, GPS devices, and PCs [4]. Additionally, e-waste includes cooling appliances like refrigerators and air conditioners, screens and monitors including TVs and laptops, lighting devices such as neon lights and LED lamps, as well as large and small appliances ranging from washing machines and dishwashers to calculators and electric shavers [5]. This diversity of e-waste highlights the complexity of managing and recycling these materials effectively to mitigate their environmental impact.

According to data from the Indonesian Ministry of Environment and Forestry, Indonesia generated approximately 33,683.781 tons of e-waste in 2021, and this number is expected to increase annually [1]. It is estimated that by 2028, Indonesia will produce about 49,627,917 tons of e-waste, with an average annual growth rate of 14.91% [6]. To overcome such problem, one crucial step in waste management is the process of waste sorting. Proper segregation and handling of e-waste are vital to prevent environmental pollution and to facilitate recycling and safe disposal of hazardous materials. Effective e-waste management not only mitigates negative environmental impacts but also opens avenues for resource recovery and recycling, contributing to a circular economy approach.

In Indonesia, waste materials with resale value, such as plastic, glass, paper, and metal, are typically sorted by the informal sector, including scavengers.

These materials are either recycled or sold. However, sorting waste by scavengers poses significant health risks, such as disease contamination, injury risks from sharp objects like glass or needles, falls, and inadequate personal protective equipment [7].

To address waste management issues, an effective waste sorting method is necessary so that different types of waste can be processed appropriately. Inefficient waste sorting methods lead to poorly sorted waste accumulation, resulting in uncontrolled landfill growth and rapid overloading of waste processing facilities.

There are various waste sorting methods, including manual sorting and automated sorting. Manual sorting relies on human labor and skills, while automated sorting uses equipment or software to make the process more efficient [8]. The fourth Industrial Revolution has introduced waste sorting methods utilizing machine learning based on image data, replacing manual sorting methods. Several machine learning methods for waste classification include Bayesian networks [9], Artificial Neural Networks (ANN) [10], K-Nearest Neighbor, random forest, and Gaussian naive bayes [11].

Among all mentioned methods, deep learning using Convolutional Neural Networks (CNN) excels in image classification. CNN, a type of Neural Network, comprises several layers commonly used for image detection and is applied in image classification, object tracking, image segmentation, etc. [12]. Research conducted by Sami et al. [13] compared various machine learning and deep learning methods in waste sorting. The methods used in this study include Support Vector Machine (SVM), Random Forest, Decision Tree, and CNN. The study found that CNN achieved the highest accuracy rate of 90%, while other methods like SVM achieved 85%, and random forest and decision tree reached 55% and 65%, respectively. Based on this comparison, CNN is chosen for application in this study.

Additionally, based on the types of waste mentioned earlier, this study will investigate 13 data classes, separating plastic classes by type and including new classes not explored in other studies, such as PVC and PS, and differentiating e-waste classes based on the composition of the printed circuit board. The study will also add wood and clothing classes, which have not been extensively researched in other studies.

Literature review

Currently, several CNN architectures are used to build waste classification models, including AlexNet, ResNet, DenseNet, VGGNet, among others. Each of these models achieves relatively high accuracy. However, for waste classification, there is a lack of large datasets currently available [14]. A few public datasets like Trashnet are available for waste classification. Developed by Yang et al. [15], Trashnet is a publicly accessible dataset with 2527 images across six classes: 594 papers, 501 glass, 482 plastics, 410 metals, 403 cardboards, and 137 trashes. Trashnet is commonly used in waste classification research, as seen in studies by Bircanoglu et al. [16], Yujie et al. [17], and Ziouzios et al. [18].

In the research by Bircanoglu et al. [16], the Trashnet dataset was used to classify six image classes. This study employed various CNN models with a train-from-scratch method, including ResNet50, MobileNetv2, Inception ResNet, InceptionV4, DenseNet121, Xception, and the RecycleNet model, a modification of DenseNet 121 with altered skip connection patterns in DenseNet121's dense block. Additionally, various optimizers were experimented with, including Adam and Adadelta. The experiment with DenseNet121 using transfer learning and the Adam optimizer achieved an accuracy of 95%, while the RecycleNet model trained from scratch reached an accuracy of 81%, with training times of 15.9ms (GPU) and 352ms (CPU) over 200 epochs.

Yujie et al. [17] experimented with two modified AlexNet models: one using softmax activation in the fully-connected layer with categorical cross-entropy loss function and the other using SVM in the fully-connected layer with categorical hinge loss. The experiments, which included partial data augmentation techniques (30 epochs with data augmentation followed by 30 epochs without), yielded an accuracy of 79.94% with the second model.

Ziouzios et al. [18] applied cloud computing for waste sorting at recycling centers using the transfer learning method with the MobileNetV2 CNN architecture. MobileNetV2 was chosen for its ability to achieve high accuracy with few hyperparameters. The experiment conducted over 20 epochs and an optimal learning rate of 1.66 x 10^{-3} achieved an accuracy of 96.57%. Common errors in this study included misclassification of metal, plastic, and glass due to the transparent and reflective attributes of these materials.

Mao et al. [19] used the Trashnet dataset for their research, optimizing the DenseNet model with a genetic algorithm for fine-tuning hyperparameters in DenseNet121's fully-connected layer. The optimization of neuron numbers and dropout rate was the primary focus for improving the DenseNet121 model. The optimized DenseNet121 achieved an accuracy of 99.60%, the highest compared to other CNN models for the Trashnet dataset, with a training time of 5542 seconds over 40 epochs.

Although widely used, the Trashnet dataset has several weaknesses as outlined by Zhang et al. [20].

These include a limited amount of data, uneven distribution of waste types, and homogenous backgrounds in the images that do not accurately represent real-world conditions, hindering the model's generalization ability. To address these limitations, Zhang et al. [20] used the NWNU-dataset, comprising 18,911 images across five classes: paper, glass, plastic, metal, and fabric. This study utilized transfer learning with a pre-trained Densenet169 model, fine-tuned to the dataset. The resulting accuracy was 82.8%, with a testing time of 22.56 sec.

To achieve more comprehensive classification results, additional data is required. Patrizi et al. [21] and Huynh et al. [22] added organic waste classes to their datasets. Patrizi et al. [21] used the Compostnet dataset, an extension of Trashnet with 175 organic waste images and 49 trash images. They employed the BackRep augmentation technique, changing image backgrounds to various conditions to reflect real-world scenarios. The experiment conducted in controlled conditions (conveyor -belt) and natural environments showed that BackRep-enhanced InceptionV4 models performed better in natural scenarios, achieving 44.2% accuracy compared to 40.9% with the standard dataset.

Huynh et al. [22] expanded the Trashnet dataset by adding 4,113 images sourced from Google, increasing the total to 6,640 images across seven classes. They used ResNet101, achieving an accuracy of 92.43%. Majchrowska et al. [23] combined 10 publicly available datasets, reaching 75% accuracy using the EfficientNet-D2 model. Karthikeyan et al. [24] merged four datasets, including Trashnet, using an eight-layer CNN architecture, achieving 98% accuracy.

Karthikeyan et al. [24] also included a wood class in their dataset, a rarity in previous studies. Their total dataset comprised 7,000 images across seven classes, including wood. They utilized the DDR-net model, a modification of the ResNext model, implementing double fusion and regularization techniques. The DDR-net model achieved 97.3% accuracy, higher than the tested ResNext101 model with 93% accuracy.

Previous studies on plastic waste classification have not differentiated between plastic types. To enhance the success of plastic waste management, systems capable of sorting plastics by type are needed. Bobulski et al. [25] developed a CNN architecture to classify four types of plastic waste: HDPE, PET, PP, and PS. The model, with 15 convolutional layers, achieved 99% accuracy in training but dropped to 74% in testing.

Chazhoor et al. [26] used transfer learning with six pre-trained models, including AlexNet, ResNet-50, ResNeXt, MobileNetV2, DenseNet, and SqueezeNet, to classify plastic waste. The highest accuracy was achieved by ResNeXt, with 87.44% in training and a 13.11-min training time over 20 epochs. MobileNetV2 was noted for its slightly lower accuracy but faster training time. In addition, inspired by the gaining attention of E-waste due to its increasing volume, Baker et al. [27] classified 12 types of smartphones from six brands using AlexNet, achieving 98% accuracy with the Stochastic Gradient Descent with momentum (SGDM) optimizer.

According to the previous literatures, most studies only focused on developing models based on a specific dataset. Meanwhile, certain classes are only available in specific datasets, while not present in others. Therefore, this study aims to combine several publicly accessible datasets and add new classes like wood, clothes, and e-waste. The plastic and e-waste classes will be further differentiated. The e-waste will be divided into PCB-containing and non-PCB waste, an aspect not previously explored. This research will classify waste into 13 classes, including PET, HDPE, LDPE, PP, PS, PVC, e-waste PCB, and e-waste non-PCB. The potential development of waste sorting using CNN will be explored with two methods: transfer learning using DenseNet121 and MobileNetV2, and building from scratch. The choice of DenseNet121 is based on its high accuracy in previous studies, while MobileNetV2 is selected for its small size and relative high accuracy, making it suitable for application in smaller devices.

Data and method

1) Datasets

The data sources for this research include several public datasets such as Trashnet, Trashbox, the Clothing dataset, the Garbage Classification dataset, and waste from a sushi restaurant. Relevant image classes for the study are selected from these datasets. For image classes not available in these datasets, data collection is performed by downloading images from Google. Following this, a series of data preprocessing steps will be conducted on these datasets to produce image data ready for use in the training and testing phases of the CNN model. This approach ensures a comprehensive and diverse dataset, enhancing the model's ability to generalize and perform accurately across various types of waste materials.

Once the data is collected, the next step involves data reduction to avoid class imbalance, which can lead to model bias towards classes with dominant data, thereby reducing the model's ability to generalize. Another reason for data reduction is the limitation of computational performance of the device. Large datasets require devices with high computational power. The parameter for data reduction is to decrease the number of images in larger classes to match the class with the least number of images, which in this case is the HDPE class. After data reduction, the total number of images in the dataset is adjusted, with the detailed breakdown of the revised dataset as presented in Table 1.

2) Data preprocessing

2.1) Splitting and shuffling data

In this research, the data is divided into three parts: training, validation, and test sets. Splitting is crucial for segregating the data, while shuffling randomizes the order to prevent bias. Both splitting and shuffling are conducted simultaneously using scikit-learn library. The splitting is executed by determining the ratio between training, validation and test. A 70:15:15 ratio is used for training, validation, and test sets, respectively, as substantial training data is necessary for the model to recognize features in the input [16, 18]. Specifically, the training set contains 3,210 images, while the validation and test sets each have 688 images. This distribution ensures a balanced approach to model training and evaluation, enhancing the model's performance and generalizability. In addition, the data shuffling is performed randomly to prevent bias, improve model robustness, and ensure that the experiments are reproducible.

2.2) Rescale dan resize

The processes of rescaling and resizing in this research are performed using the ImageDataGenerator library from Keras. Rescaling is crucial as it multiplies the data before any other processing. Since RGB image data has coefficients ranging from 0-255, these values are too high for the model to process effectively. Therefore, rescaling is done by multiplying the data values by 1/255 to normalize them to a range between 0 and 1. This normalization aids in more efficient and stable model training.

On the other hand, resizing alters the dimensions of the input data to ensure uniformity across all inputs. For this study, an input image size of 300 x 300 pixels is used. Consistent image size is essential for the model to process the data uniformly, which helps in maintaining the integrity of the image features and facilitates more accurate predictions by the model. By implementing both rescaling and resizing, the study ensures that the input data is optimally prepared for processing by the CNN model.

2.3) Data augmentation

The augmentation process in this study is carried out using the same library as in the rescale and resize stages, namely ImageDataGenerator from Keras. Data augmentation transforms the input data to generate and add new variations, thereby increasing the amount of data. This is achieved by employing built-in parameters of the ImageDataGenerator library for data transformation. Several parameters are utilized for data augmentation in this research, including:

1. Height shift range:

This process shifts the image vertically, both upwards and downwards, in proportion to the image's height. In this study, the maximum range used is 0.2, meaning the maximum vertical shift is 20% of the image's height. Figure 2 presents the images before and after height shift range.

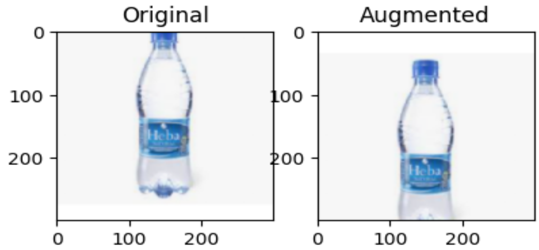


Figure 2 Result of height shift range.

2. Shear range:

This process skews the image along the x and y axes, providing a distorted perspective in the image data. The attribute used for this augmentation process is set to 0.2, causing skewing along both the x and y axes by 0.2. Figure 3 presents the shear range result.

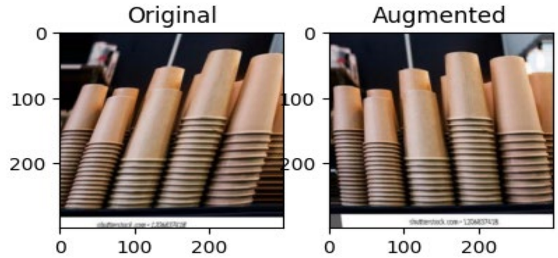


Figure 3 Result of shear range.

Table 1 Number of images per class

Class	HDPE	LDPE	PET	PP	PS	PVC	Clothes	E waste non PCB	E waste PCB	Glass	Metal	Paper	Wood
Number of images	340	368	931	350	561	366	7295	389	655	5170	2941	3699	1049

3. Zoom range:

This process zooms into the image, either by enlarging it or by adding pixels around the image to make it appear larger. The zoom range attribute is set at value of 0.2, meaning the image will be zoomed in or out by 20% of its original size, as illustrated in Figure 4.

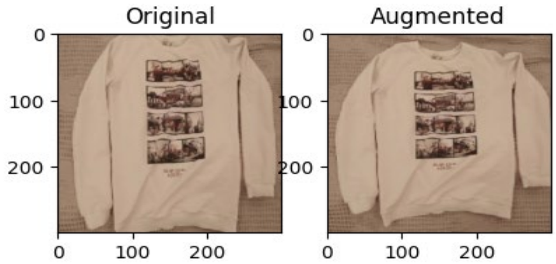


Figure 4 Result of zoom range.

4. Horizontal flip:

The horizontal flip attribute is set to true to generate new variations of the image data with a horizontal orientation different from the original image. Figure 5 presents an example of horizontal flip.

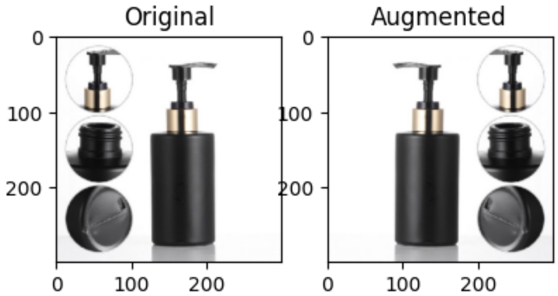


Figure 5 Results of horizontal flip.

These augmentation techniques effectively increase the diversity of the dataset, allowing the model to learn from a broader range of data variations and thereby enhancing its generalization ability.

3) Pretrained model

The selection of pretrained models in this study refers to those available in the Keras Applications module, which offers a variety of pretrained models along with their weights trained on ImageNet. These models can be used for predictions, feature extraction, and fine-tuning. In this research, pretrained models are employed specifically for feature extraction.

There are 38 pretrained Keras models available that can be chosen based on the research requirements. From these 38 models, attributes of each model are considered in the selection process, such as the model size (in MB), Top-1 Accuracy, Top-5 Accuracy, the number of parameters, and the processing time per iteration. Top-1 Accuracy and Top-5 Accuracy refer to the model's performance on the ImageNet validation dataset. Based on these attributes, two pretrained models have been selected for the transfer learning process in this study: MobileNetV2 [28] and DenseNet121 [29].

MobileNetV2 is selected due to its small size relative to other available models. This compact size results in fewer parameters compared to other pretrained models, positively influencing the model's iteration time. Consequently, MobileNetV2 boasts the fastest iteration time among its counterparts. Additionally, its Top-1 Accuracy and Top-5 Accuracy are superior to those of MobileNetV1, its predecessor, making it a more efficient choice.

DenseNet121, on the other hand, is chosen for its greater number of parameters compared to MobileNet V2. This increased complexity allows DenseNet121 to extract features more effectively, resulting in higher Top-1 and Top-5 Accuracies compared to earlier models. However, the larger number of parameters also means a larger model size, leading to slower processing times with a time difference of approximately 1.6 ms per iteration.

These selections highlight a strategic trade-off between computational efficiency and model complexity. MobileNetV2 offers rapid processing suitable for applications where speed and model size are critical, while DenseNet121 provides more detailed feature extraction, beneficial for tasks requiring higher accuracy and deeper learning capabilities.

4) Full learning model

The next architecture to be analyzed in this study is one that utilizes the full learning method. The stages involved are similar to the previous method, starting with the development of the full learning architecture, followed by training, validation, and testing phases. The architecture that undergoes training, validation, and testing is hereafter referred to as the model. The key difference from the previous model is that in this method, the model is constructed from the ground up, where the entire architecture and model parameters are manually determined. Figure 6 presents the architecture of the proposed full learning model.

The full learning model architecture in this study is meticulously constructed using the sequential method. This architecture comprises five convolutional layers, each designed to capture complex patterns in the dataset. The first layer contains 16 filters of size 7x7, the second layer has 32 filters of size 5x5, while the third, fourth, and fifth layers have 64, 128, and 256 filters, respectively, all with a filter size of 3x3. Each layer utilizes the Rectified Linear Unit (ReLU) activation function. Following every convolutional layer, there is a max pooling layer of size 2x2, which extracts the maximum value from each grid to reduce the spatial dimensions of the feature maps.

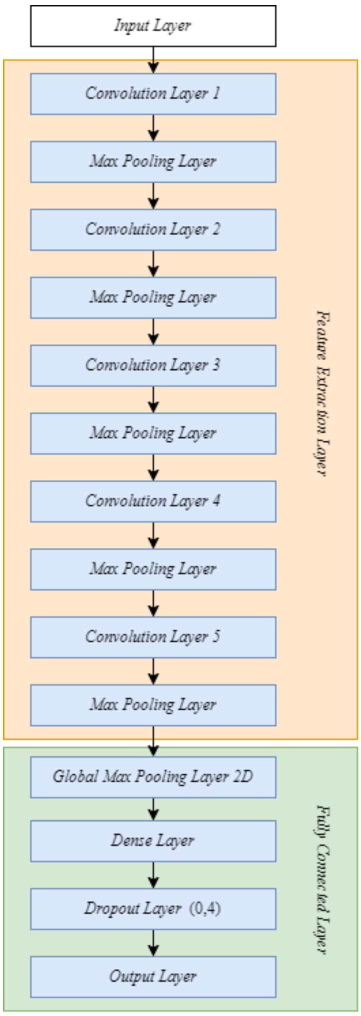
Subsequently, a Global Max Pooling layer is employed to further reduce the dimensions of the feature maps from three dimensions to one, making them suitable for the classification layers. The architecture includes two dense layers and one dropout layer for the classification process. The first dense layer comprises 512 neurons with the ReLU activation function. A dropout layer with a rate of 0.4 follows the first dense layer to reduce model complexity. The second dense layer has 13 neurons, aligning with the 13 classes in the dataset, and uses the softmax activation function, suitable for multi-class classification problems.

This architecture results in a total of 540,973 trainable parameters. The training process is conducted over 100 epochs to achieve optimal results. This methodical approach to building the architecture from scratch allows for fine-tuning and adapting each layer to the specific requirements of the dataset, potentially leading to a more effective model for the given classification task.

Building a model from scratch allows for complete customization and control over the model's architecture, making it possible to tailor the model specifically to the dataset's characteristics and the research objectives. This method offers the advantage of a deeper understanding of how each component of the model contributes to its overall performance. However, it requires careful consideration and expertise in designing the architecture and selecting appropriate parameters to ensure optimal performance. This approach is particularly beneficial for exploring innovative or non-standard architectures that might not be available in pretrained models.

5) Hyperparameter setting

The training phase is conducted to extract features from the input images. During this process, several hyperparameters are set and remain constant throughout the training and validation stages. These hyperparameters are crucial as they significantly influence the learning process and the overall performance of the model. The specific hyperparameters used in this research are detailed in Table 2.




```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 294, 294, 16)      2368

 max_pooling2d (MaxPooling2  (None, 147, 147, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 143, 143, 32)      12832

 max_pooling2d_1 (MaxPoolin  (None, 71, 71, 32)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 69, 69, 64)        18496

 max_pooling2d_2 (MaxPoolin  (None, 34, 34, 64)        0
 g2D)

 conv2d_3 (Conv2D)           (None, 32, 32, 128)       73856

 max_pooling2d_3 (MaxPoolin  (None, 16, 16, 128)       0
 g2D)

 conv2d_4 (Conv2D)           (None, 14, 14, 256)       295168

 max_pooling2d_4 (MaxPoolin  (None, 7, 7, 256)         0
 g2D)

 global_max_pooling2d (Glob  (None, 256)               0
 alMaxPooling2D)

 dense (Dense)               (None, 512)               131584

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 13)                6669

=================================================================
Total params: 540973 (2.06 MB)
Trainable params: 540973 (2.06 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 6 Full learning model.

Table 2 Hyperparameters setting

No	Hyperparameter	Value
1	Learning rate	0.0001
2	Loss function	Categorical crossentropy
3	Evaluation metrics	Accuracy, precision, recall
4	Image size	300 x 300
5	Batch Size	32
6	Epochs	100

The learning rate is a parameter that controls how much the model adjusts its parameters during the training process. In this study, a relatively small learning rate of 0.0001 is used. This slow rate of parameter updates minimizes the loss, allowing for more precise model adjustments. The loss function employed is categorical crossentropy, which is suitable for multi-class classification problems. The metrics used are accuracy, precision, and recall, serving to display the evaluation values of these three metrics at each epoch during the training process. Additionally, these metrics are used for generating accuracy, precision, and recall graphs. During the training, the same hyperparameters setting are set for both transfer learning and full learning models, except the image size which follows the input size of respective architecture, to ensure fair comparison.

The image size used in full learning model is 300 x 300 pixels, chosen to align with the computational capabilities of the equipment used. A larger image size would require more parameters in the training process, thus increasing the computational power needed. The study uses 100 epochs, meaning that the training process iterates 100 times. A batch size of 32 is employed, meaning that 32 images are processed in each training epoch. With a total of 3210 training images and a batch size of 32, each epoch involves approximately 100 training iterations.

After completing the training and validation processes, testing is conducted using the trained model on the test set. Testing is performed on new data that the model has not previously encountered to assess how well the model classifies data accurately. The batch size for testing is also set at 32, and no data augmentation is done during this process. The results of the testing process include an evaluation of accuracy, precision, recall, F-1 score, and a confusion matrix. This methodology ensures a thorough evaluation of the model's performance, providing insights into its strengths and areas for improvement, especially in terms of its ability to generalize and accurately classify new data.

Result and discussion

1) Training results

1.1) MobileNetV2

In the transfer learning process of this study, the model undergoes training and testing phases using the weights of MobileNetV2, with a key step being the freezing of layers in the pretrained model. The purpose of freezing layers is to retain the weights in the feature extraction layer without retraining them. This approach ensures that the pretrained model's learned patterns are preserved, leveraging its prior knowledge for the current dataset.

Additionally, modifications are made to the fully connected layer to suit the research needs. These modifications include adding a dropout layer with a rate of 0.2 to reduce the model's complexity, thereby decreasing the risk of overfitting. Overfitting occurs when a model learns the training data too well, including the noise and outliers, which can negatively affect its performance on new, unseen data. The dropout layer helps in preventing this by randomly dropping units from the neural network during training, which forces the model to learn more robust features. Furthermore, the output layer is altered from 1000 classes to 13 classes, aligning with the number of classes in this study's dataset. This change tailors the model more specifically to the task at hand.

The final architecture results in a total of 3,583,053 parameters, of which 1,325,069 are trainable, and 2,257,984 are non-trainable. This division of parameters allows for a balance between leveraging the learned features from the MobileNetV2 model and fine-tuning the model to fit the specific data and classification tasks of this study. This strategic combination of pretrained knowledge and task-specific adaptation is key to the success of transfer learning methodologies.

The model is evaluated both before and after the training phase using the validation set to measure accuracy and loss values pre- and post-training. This approach provides a clear insight into the model's performance improvements due to training. The model demonstrates a significant increase in accuracy and a decrease in loss value after the training process, as indicated in Figure 7.

Based on Figure 7, both the training and validation accuracy curves are close together and plateau at a high level, which indicates good generalization. There is no significant gap between the two, suggesting that the model is not overfitting. The training accuracy starts high and remains stable throughout the epochs, which could indicate that the model had a good initialization, possibly from effective pretraining or an easy-to-learn dataset.

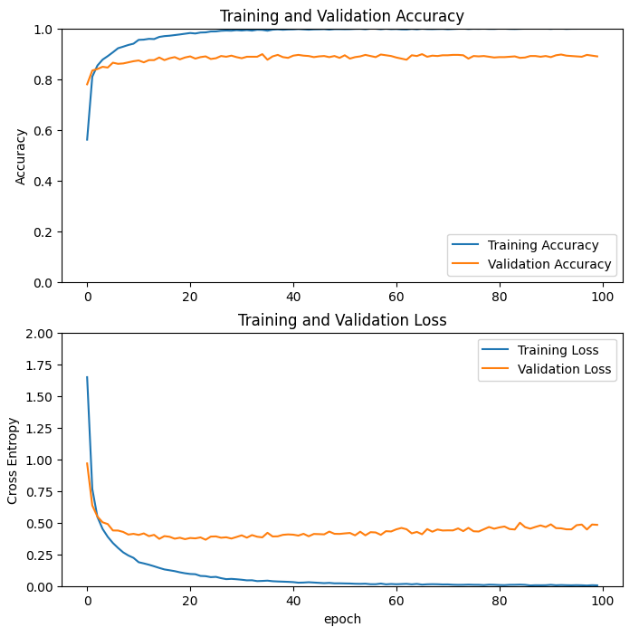


Figure 7 Training progress of MobileNetV2 model.

Both the training and validation loss curves show a sharp decline initially and then level off, which is typical and desirable during the training of a neural network. This suggests that the model quickly learned to reduce errors on the training set and validation set. The validation loss closely follows the training loss, which again indicates that the model generalizes well to unseen data. There is no sign of divergence between the two curves, which would suggest overfitting.

Overall, the figure suggests that the model has trained successfully, achieving high accuracy and low loss on both the training and validation sets, indicating a well-fitting model. The close convergence of training and validation lines further suggests that the model should perform reliably when making predictions on new, similar data.

The improvement in accuracy and reduction in loss is an indication that the model has successfully learned from the training data, adjusting its weights and biases to better predict the validation set. Such evaluations are essential for iterative model development, allowing researchers to fine-tune training parameters and model architecture to achieve optimal performance.

1.2) DenseNet121

The training process of DenseNet121 transfer learning models follows the same procedure of MobileNetV2 training with freezing layers to retain the weights in the top layers. Similarly, the model is trained with 100 epochs. Figure 8 presents the training progress of DenseNet121 transfer learning model in the perspective of accuracy and loss of training and validation datasets.

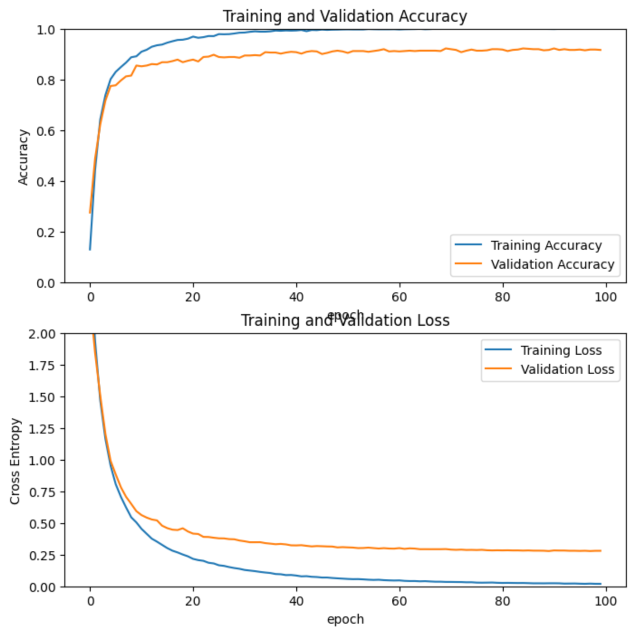


Figure 8 Training progress of DenseNet121 model.

Similar to the training of MobileNetV2 models, the accuracy curves for both training and validation start high and plateau, indicating that the model quickly reaches a high level of accuracy and maintains it, which is indicative of effective learning. The close proximity of the training and validation accuracy lines suggests that the model generalizes well and is not overfitting to the training data.

The loss curves for both training and validation sharply decrease at the beginning and then level off, which is typical of neural network training. This indicates that the model rapidly minimizes the loss function at the start of training. The validation loss closely mirrors the training loss throughout the training process, which again suggests the model has good generalization capabilities. As compared to the training of MobileNetV2 models, the gap between training and validation loss of DenseNet121 models are narrower, which indicates that it has better generalization capabilities of the model.

1.3) Full learning model

Unlike the transfer learning models, during the learning process, all layers and parameters of full learning model are trained. The training is executed using the hyperparameters value set in Table 2. The graph of accuracy and loss for training and validation datasets of full learning model during the training process are presented in Figure 9.

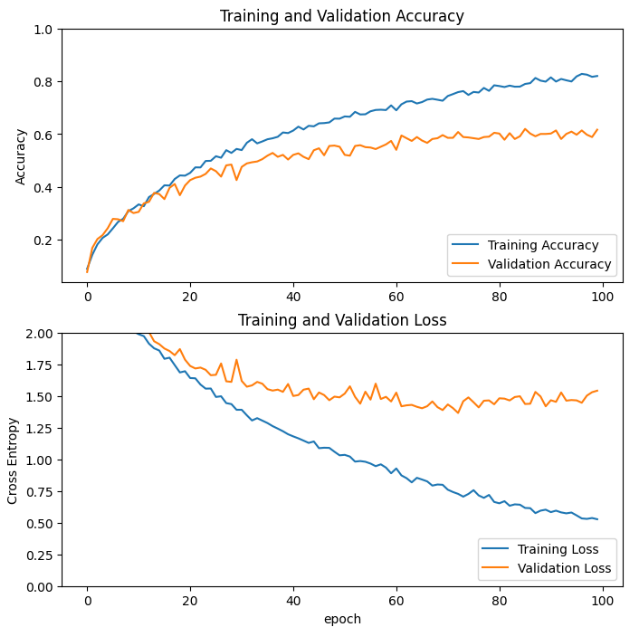


Figure 9 Training progress of full learning model.

Based on Figure 9, the training accuracy steadily increases over the 100 epochs, which indicates that the model is learning from the training data. The validation accuracy also increases, but it plateaus and fluctuates more than the training accuracy. This fluctuation could suggest the beginning of overfitting or a model that is less stable on the validation data.

The training loss decreases over time, showing that the model is increasingly fitting the training data well. The validation loss decreases initially but then starts to fluctuate and even slightly increase towards the later epochs. This behavior is a potential indicator of overfitting, as the model starts to learn the noise in the training data rather than generalizable patterns.

Unlike the previous models, where training and validation lines were close and stable, this figure shows more separation between training and validation lines, especially in the loss graph. The previous graphs showed a model that had very little change in validation loss and accuracy, maintaining a high level of performance across epochs. In contrast, this figure suggests that the model's performance on the validation set is less stable and possibly overfitting towards the end of the training epochs. While the model is still learning and improving its performance on the training set, the divergence in the validation metrics suggests that it may not generalize as effectively to new data. The increased fluctuation in the validation metrics implies that the model may be capturing noise rather than underlying patterns in the later stages of training.

1.4) Training results comparison

Figure 10 presents the comparison of indicators obtained by the models after training. Based on the evaluation results of the accuracy, precision, and recall matrices across all epochs used on the training and validation data for each type of constructed model, it can be concluded that the transfer learning DenseNet121 model produces the best results among the other models used. The DenseNet121 model shows the best outcomes on both training and validation data, with a positive upward trend during training and no indication of fluctuating data, suggesting that there are no signs of overfitting.

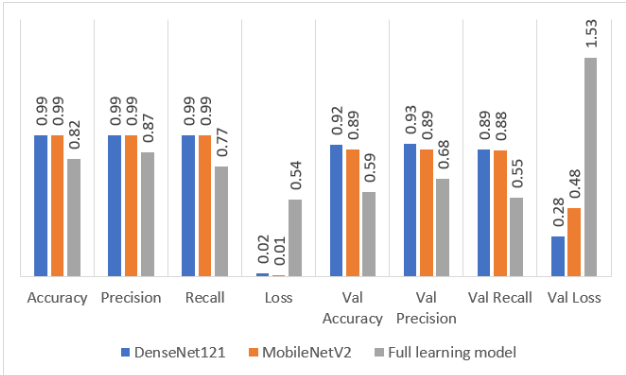


Figure 10 Comparison of training results.

The image above illustrates a comparison among the models based on accuracy, loss, precision, and recall metrics for training and validation data. This comparative analysis helps to highlight the relative strengths and weaknesses of each model and underscores the superior performance of the DenseNet121 model in terms of consistent learning and generalization without overfitting to the training data. Such a comparison is vital for selecting the most effective model for deployment in real-world applications.

In Figure 10, it is observed that the transfer learning DenseNet121 model generally achieves the most optimal results compared to the other models, except in the loss metric where DenseNet121 has a slightly higher loss of 0.0193 compared to MobileNetV2's loss of 0.0053. However, this difference in loss is not significant, and overall, the DenseNet121 model performs better in other metric aspects, making it the best model of those tested.

The MobileNetV2 model also displays commendable results, although its performance is slightly less favorable than that of DenseNet121. Despite this, both the DenseNet121 and MobileNetV2 transfer learning models have achieved good results and are deemed implementable, having reached a satisfactory point of convergence.

On the other hand, the full learning model is considered less optimal compared to the two other models because it exhibits inferior results across all evaluation metrics used. The loss metric, in particular, draws attention in the full learning model, with a significant difference from the other models, registering a loss value of 0.54. This indicates a need for further adjustments and improvements in the full learning model to enhance its performance and bring it closer to the standards achieved by the pretrained models.

In the perspective of efficiency, the model that requires the shortest training time is the transfer learning MobileNetV2, clocking in at 173.95 min. In contrast, the DenseNet121 model requires the longest training time, amounting to 189.87 min. This difference is attributable to the varying number of parameters used in each model.

Considering the comparisons across various factors, it can be concluded that the transfer learning Dense Net121 model delivers the most optimal results. The DenseNet121 model is deemed successful in addressing the challenges posed in this research, demonstrating the most favorable outcomes with the constructed dataset. Therefore, based on the training phase evaluations, the DenseNet121 model is the best choice for solving multi-class waste classification problems due to its highest accuracy rate.

2) Test results

The testing phase is conducted using the test set to evaluate the model's accuracy after training with new image data that the model has not previously encountered during the training process. This step is critical as it assesses the model's generalization capabilities and its effectiveness in classifying unseen data. It serves as a measure of how well the model can be expected to perform in real-world scenarios, where it will encounter images that were not part of its training dataset. Ensuring a robust testing phase is essential for validating the practical applicability of the model. Based on the testing results using the same test set for each model, a comparison of accuracy on the test set data was conducted among the three constructed models. The outcomes of this comparison can be viewed in Table 3.

According to Table 3, all models perform well on the test data in terms of accuracy. Both transfer learning models achieved an accuracy above 0.90, and the full learning model reached an accuracy of 0.65. High accuracy across all data classes indicates that the transfer learning models can predict correctly with a high degree of reliability.

In addition to accuracy, the precision values for the three models show that the transfer learning models yield satisfying results, with the DenseNet121 model displaying a precision of 0.95 and the MobileNetV2 model showing a precision of 0.92. On the other hand, the full learning model has less satisfactory precision, with a value of 0.65. Precision indicates how often the model accurately predicts a given class when it predicts that class.

The next metric, recall, shows that the highest recall value is from the DenseNet121 transfer learning model at 0.95, followed by the MobileNetV2 model at 0.91, with the full learning model at 0.64. Recall measures how often the model correctly identifies an actual class A as class A. The final metric, the F1 Score, which is the harmonic mean of precision and recall, shows the DenseNet121 transfer learning model with the highest value of 0.95, followed by the MobileNetV2 model at 0.92. The full learning model has the lowest F1 Score at 0.65. The F1 Score is a balanced metric that considers both false positives and false negatives, giving a more comprehensive view of the model's performance.

These metrics suggest that the DenseNet121 model is the most robust among the three, offering the best balance between precision and recall, which are critical in a multi-class classification setting. It implies that DenseNet121 is not only accurate but also consistent in its predictions across various classes. The overall performance makes DenseNet121 the preferred model for tasks requiring high reliability in predictions. Further, the detailed analysis of the model's performance in each class is presented in Figure 11.

Based on Figure 11, it is apparent that the accuracy of test data among the three models is reported. Both models employing transfer learning techniques perform exceedingly well, with each achieving accuracy rates above 0.90 in most data classes. However, the DenseNet121 model edges out with superior accuracy on the test data, particularly in certain classes such as LDPE and paper, where it outperforms the Mobile NetV2 transfer learning model.

Conversely, the full learning model shows lower accuracy compared to the transfer learning models, with the highest accuracy for this model being 0.81 for PCB e-waste, while the lowest being 0.29 for metal. The higher accuracy in the transfer learning models is likely due to the pretrained knowledge about general features relevant to the data classes involved in this research.

Transfer learning models benefit from the knowledge gained from large, diverse datasets like ImageNet, which often results in a model that is better at extracting and generalizing the essential features from the input data. This pre-existing knowledge base allows the model to quickly adapt to the specific tasks of the current research, leading to higher performance levels on test data. The detailed performance in each class is depicted in the confusion matrix of the three models, presented in Figures 12–14.

Tabel 3 Comparison of testing results

Model	Accuracy	Precision	Recall	F-1 Score	Training time (min)
DenseNet121 Transfer learning model	0.95	0.95	0.95	0.95	189.87
MobilenetV2 Transfer learning model	0.92	0.92	0.91	0.92	173.95
Full learning model	0.65	0.65	0.64	0.65	183.89

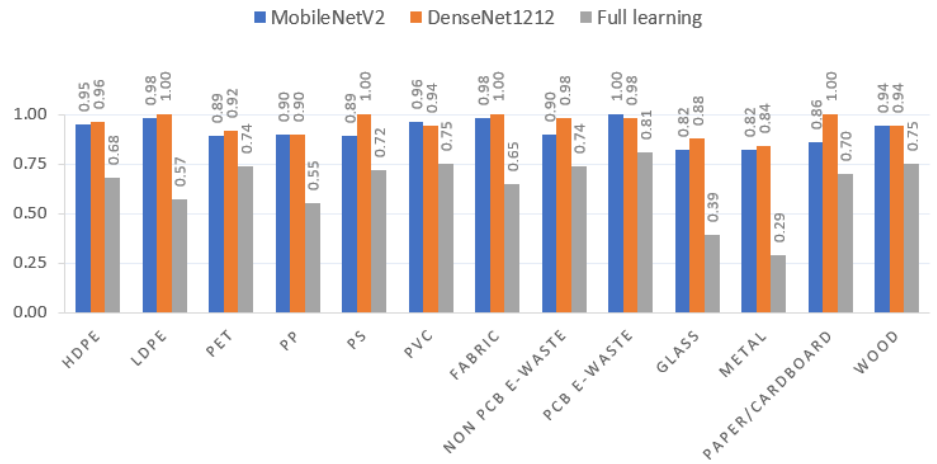


Figure 11 Accuracy comparison across all classes.

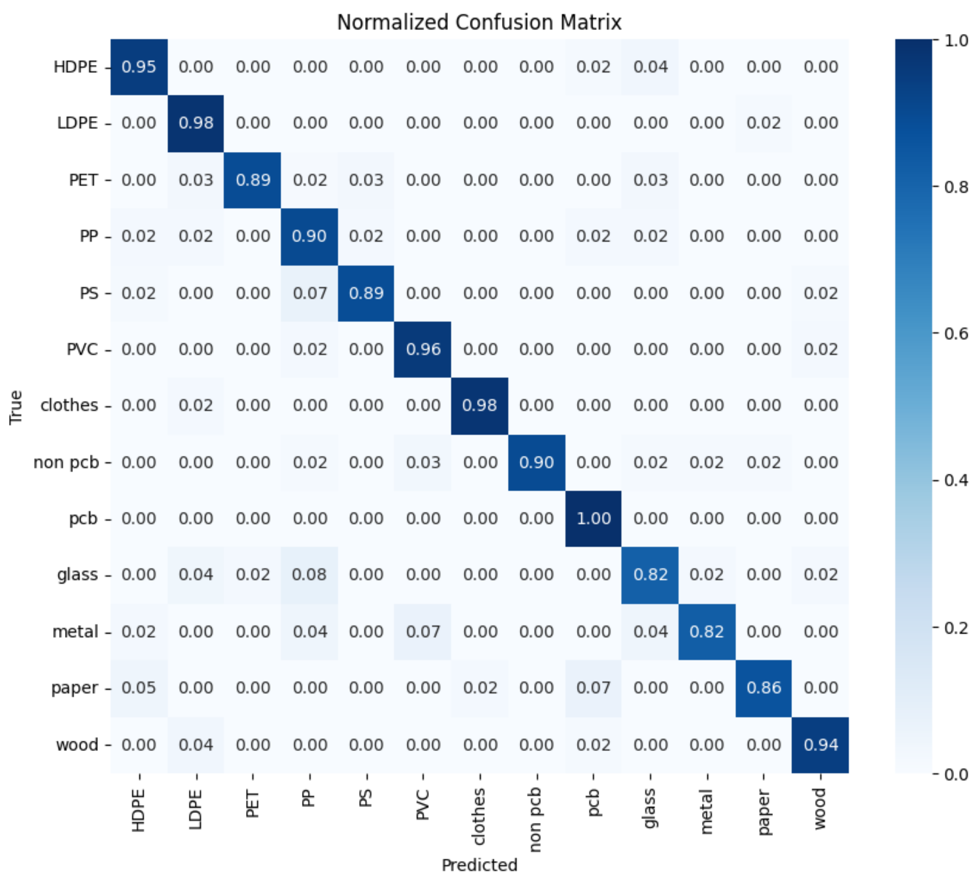


Figure 12 Normalized confusion model MobileNetV2 transfer learning matrix.

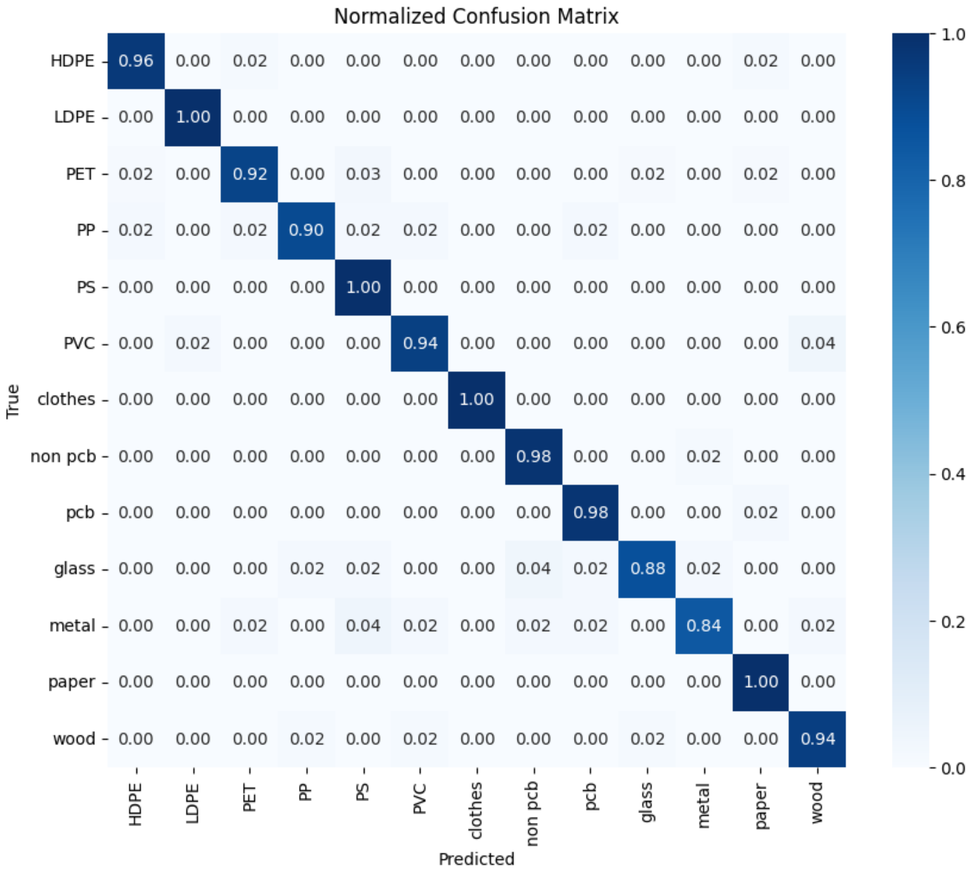


Figure 13 Normalized confusion matrix DenseNet121 transfer learning model.

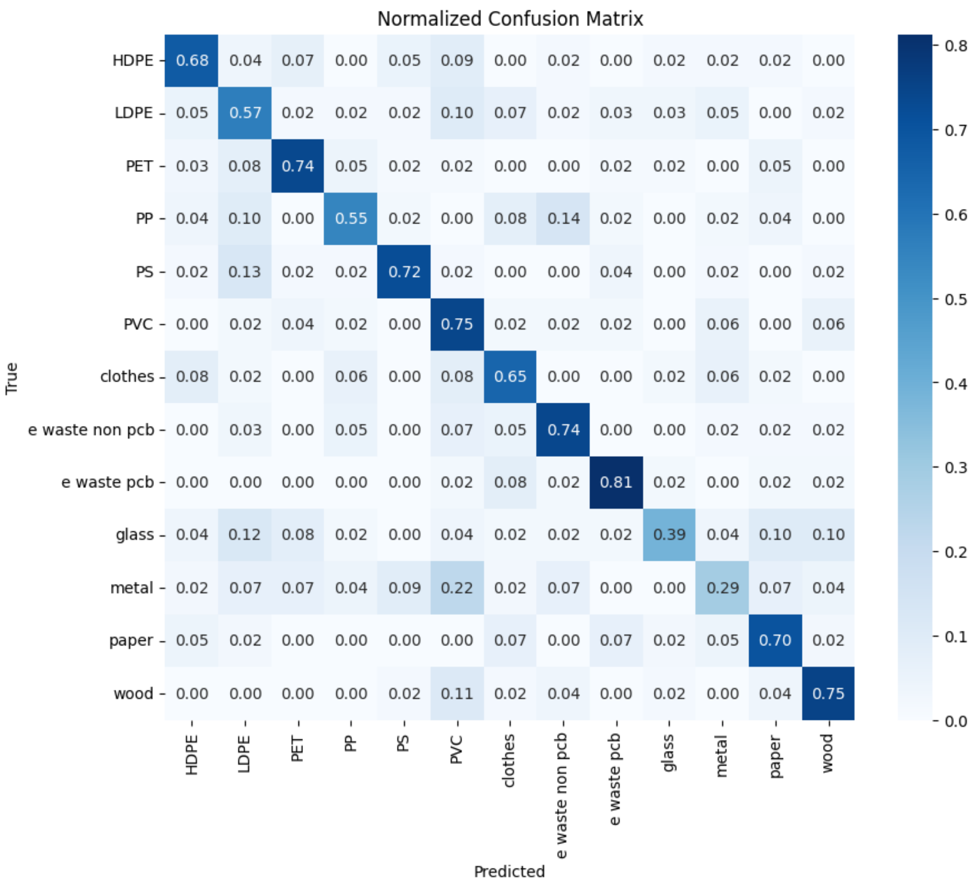


Figure 14 Normalized confusion matrix full learning model.

Based on Figure 12, the waste classes that show suboptimal performance in MobileNetV2 model compared to other classes are glass and metal. The accuracy for the glass and metal classes did not exceed 0.82. This could be due to the model's difficulty in distinguishing between the glass and PP classes, as both have similar characteristics, such as transparency, leading to a higher number of false negatives where actual positive instances are incorrectly predicted as negative by the CNN model. In addition, the model may struggle to differentiate between metal and PVC, as they both can exhibit similar features, like a gray color, which can confuse the model and affect its classification accuracy.

A similar pattern is also recorded for the Dense Net121 model shown in Figure 13 in which the lowest accuracy was obtained by both glass and metal classes. Some of the false negatives encountered within the glass waste class occur when the model incorrectly predicts glass as non-PCB e-waste, PP, PS, PCB-containing e-waste, or metal. For metal classes, the model exhibits some errors in identifying the metal class, mistakenly detecting it as other classes, including PET, PVC, PS, non-PCB e-waste, and wood. This misclassification may arise because these materials share visual properties with glass, such as shininess, rigidity, or color.

Meanwhile, for the full learning model, the accuracy in each class is significantly lower than the transfer learning models. Certain classes like glass and metal show significant confusion with other materials, as indicated by the lower diagonal values (0.39 for glass and 0.29 for metal) and higher misclassification rates with other classes. Classes like HDPE and LDPE also show some confusion with each other, which is understandable given their material similarities. In overall, improvements are required to increase the performance of full learning model. Some techniques for improvement might include collecting more diverse data for these classes, implementing class-specific data augmentation, or adjusting the model architecture to better capture the distinguishing features of these materials.

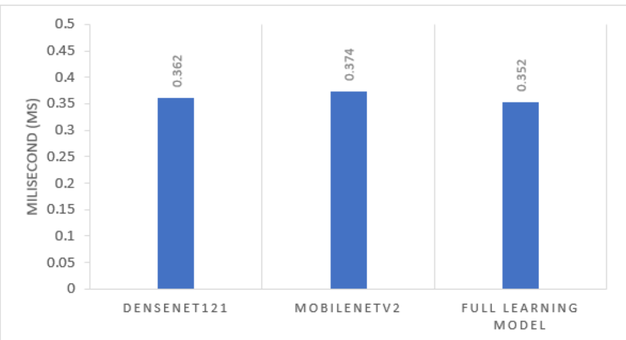


Figure 15 Inference time per sample.

The comparison of testing results also takes into account the testing/inference time for each model used in this research. The comparative testing times can be viewed in Figure 15. According to this figure, the full learning model has the shortest testing time at 0.352 milliseconds, followed by the DenseNet121 transfer learning model at 0.362 milliseconds, and the MobileNetV2 transfer learning model has the longest testing time at 0.374 milliseconds.

The full learning model's shortest testing time can be attributed to its lower complexity relative to the other models, which likely contain more hidden layers and parameters. Fewer layers and parameters can lead to faster inference times, as there are less computations for the model to process when making predictions.

While shorter testing times are advantageous, especially in real-time applications or when processing large datasets, it's important to balance speed with accuracy and robustness of the model. Although the full learning model is faster, it may not necessarily perform as well in terms of accuracy when compared to more complex models like DenseNet121 and MobileNetV2. Therefore, when selecting a model for deployment, the specific application requirements must be considered, including the acceptable trade-off between speed and performance.

3) Discussion, implication, and limitation

Based on the comparison of training and testing results, it can be concluded that the best-performing model is the DenseNet121 transfer learning model. This is supported by the performance on both training and testing data for the DenseNet121 model, which generally surpasses the other models. The Mobile NetV2 transfer learning model also performs well, with results that are competitive with DenseNet121. However, despite its good performance, the overall outcomes from the Mobile NetV2 model are not better than those of the Dense Net121 transfer learning model.

Furthermore, the full learning model shows inferior results compared to the other models used. One possible reason for this could be that the hyperparameters employed were not ideally suited for the problem addressed in this research. To optimize results, comprehensive experimentation on the hyper-parameters, such as the number of layers, number of filters, filter sizes, number of neurons, activation functions, regularization, learning rate, and batch size, would be necessary.

A too small number of epochs could also contribute to suboptimal model results. However, using a large number of epochs can risk overfitting, where the model learns the training data too well but struggles to generalize to new data, leading to a significant discrepancy between training and testing accuracy.

Additionally, full learning models require a large amount of data to effectively learn patterns and features from the input images from scratch, thus classifying inputs well. It is possible that the amount of data per class used in this research has not been sufficient for the full learning model to achieve high accuracy. On the other hand, transfer learning methods have an advantage in this regard, as those models have already been trained on large and diverse datasets. Therefore, transfer learning models are more adept at learning and recognizing features and patterns in new images because they have prior knowledge of potential features in each class, making classification easier. To address this issue, next study can develop a large size database for waste images which can be used for training full learning models effectively.

This research has the capability to classify waste types more intricately by introducing new waste classes not covered in previous studies, such as electronic waste, clothing, and wood. Generally, this study has the potential to advance the field of waste management. The expected benefits from this research include increased recycling efficiency, integration with automated sorting systems, development of applications, machine learning enhancements, and reduction of environmental pollution. The limitations of this study could provide opportunities for future research development, which include:

- Object detection improvement: Since the current technique is image classification, the model requires input images to contain only one object per data class for accurate detection. If multiple objects are present in a single input image, the model may struggle to detect the correct class. To overcome this issue, object recognition techniques can be employed, which can detect multiple objects within a single input image.

- Interpretability enhancement: This study utilizes a form of CNN that is unexplainable, meaning the internal decision-making process of the model is not transparent to the researchers. Future research could focus on explainable artificial intelligence (XAI) to understand how the model processes and classifies input images.

- Data volume for full learning models: The number of data points per class is considered insufficient for creating a robust full learning model, which requires a significant amount of data. This limitation was due to the researcher's resource constraints. Future studies with access to larger datasets could enhance the model's performance.

Class representation: The waste classes used in this study do not represent all the waste types produced. Classes such as organic waste, cardboard, rubber, etc., were not included. Future studies could expand on the types of waste classified to create a more comprehensive waste management solution.

In addition, for further investigation, it would be beneficial to explore a range of convolutional neural network architectures beyond MobileNetV2 and DenseNet121 to enrich the comparative analysis and potentially improve classification accuracy. State-of-the-art architectures such as ResNet50, MobileNetV3, and EfficientNet could offer unique advantages due to their varying depths, complexities, and approaches to handling convolutional operations. ResNet50, known for its residual learning framework, could enhance the learning of deeper networks without the vanishing gradient problem. MobileNetV3, optimized for mobile and edge devices, might offer a more efficient inference time while maintaining high accuracy. EfficientNet, which scales model size in a more structured way to achieve better accuracy and efficiency, could provide a balance between accuracy and computational resources.

Incorporating these improvements into future research could lead to more advanced, accurate, and interpretable models for waste classification and management, contributing to environmental sustainability and the efficiency of recycling processes in developing countries, such as in Indonesia.

Conclusion

This study examines urban waste management identification using a deep learning approach based on image classification, employing a new dataset compiled from publicly accessible collections. The author aggregated relevant datasets, including Trashnet, Trashbox, Clothing, Garbage Classification, Waste from Sushi Restaurants, and Most Common Recyclable and Non-Recyclable Objects, resulting in a new dataset of 4,586 images across 13 classes. This dataset was then divided into training (3,210 images), validation (688 images), and testing (688 images) subsets. Model development, training, and testing yielded several insights. Three models were built using two distinct techniques: transfer learning with MobileNetV2 and DenseNet121, and a build-from-scratch approach. The DenseNet121 transfer learning model excelled in training and testing, achieving 0.95 accuracy, while MobileNetV2 also performed well with 0.92 accuracy. The full learning model did not converge satisfactorily by epoch 100 and showed signs of overfitting, with a test accuracy of 0.65. Overall, the DenseNet121 transfer learning model surpassed the other models in most evaluation metrics and showed the best performance in classifying the 13 waste classes. Despite DenseNet121's longer training time due to its larger number of trainable parameters (8,018,829), it was identified as the superior model for waste classification. The testing times for the three models are closely matched, averaging around 0.35 milliseconds per sample. This minimal variance suggests that each model can efficiently process data in near real-time, making them all suitable for real-time waste identification applications where quick decision-making is crucial.

References

[1] Indonesian Ministry of Environment and Forestry. National waste management information system, 2023. [Online] Available from: https://sipsn.menlhk.go.id/sipsn/public/data/komposisi [Accessed 15 July 2023].

[2] Lai, Y.Y., Lee, Y.M. Management strategy of plastic wastes in Taiwan. Sustainable Environment Research, 2022, 32(1), 1–10.

[3] Hidayat, Y.A., Kiranamahsa, S., Zamal, M.A. A study of plastic waste management effectiveness in Indonesia industries. AIMS Energy, 2019, 7(3), 350–370.

[4] Shamkhalichenar, H., Bueche, C.J., Choi, J.W. Printed circuit board (PCB) technology for electrochemical sensors and sensing platforms. Biosensors, 2020, 10(11).

[5] Balde, C.P., Forti V., Gray, V., Kuehr, R., Stegmann, P. The global e-waste monitor – 2017. United Nations University (UNU), International Telecommunication Union (ITU) & International Solid Waste Association (ISWA), Bonn/Geneva/Vienna, 2017. [Online] Available from: https://collections.unu.edu/eserv/UNU:6341/Global-E-waste_Monitor_2017__electronic_single_pages_.pdf [Accessed 20 July 2023].

[6] Santoso, S., Zagloel, T., Yuri M., Ardi, R., Suzianti, A. Estimating the amount of electronic waste generated in Indonesia: Population balance model. In IOP Conference Series: Earth and Environmental Science, 2019, 219(1).

[7] Cruvinel, V., Marques, C., Cardoso, V., Novaes, M., Araъjo, W., Angulo-Tuesta, A., …, Da Silva, E. Health conditions and occupational risks in a novel group: Waste pickers in the largest open garbage dump in Latin America. BMC Public Health, 2019, 19(1), 1–15.

[8] Shah, J., Kamat, S. A Method for waste segregation using a convolutional neural networks. In 2022 2nd International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT, 2022, 1–5.

[9] Liu, C., Sharan, L., Rosenholtz, R. Exploring features in a Bayesian. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, 239–246.

[10] Mohammed, M.A., Abdulhasan, M., Kumar, N., Abdulkareem, K., Mostafa, S., Maashi, M., …, Chopra, S. Automated waste-sorting and recycling classification using artificial neural network and features fusion: A digital-enabled circular economy vision for smart cities. Multimedia Tools and Applications, 2023, 82, 39617–39632.

[11] Hanbal, I.F., Ingosan, J., Oyam, N., Hu, Y. Classifying wastes using random forests, Gaussian Naпve Bayes, support vector Machine and multi-layer perceptron. In IOP Conference Series: Materials Science and Engineering, 2020, 803(1).

[12] Wang, T., Chen, Y., Qiao, M., Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. International Journal of Advanced Manufacturing Technology, 2018, 94(9–12), 3465–3471.

[13] Sami, K.N., Amin, Z.M.A., Hassan, R. Waste management using machine learning and deep learning algorithms. International Journal on Perceptive and Cognitive Computing, 2020, 6(2), 97–106.

[14] Peng, H., Shen, N., Ying, H., Wang, Q. Factor analysis and policy simulation of domestic waste classification behavior based on a multiagent study—Taking Shanghai's garbage classification as an example. Environmental Impact Assessment Review, 2021, 89.

[15] Yang, M., Thung, G. Classification of Trash for Recyclability Status. Project Report CS229, 2016, 940–945. [Online] Available from: https://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf [Accessed 21 July 2023].

[16] Bircanoglu, C., Atay, M., Beser, F., Genc, O., Kizrak, M. RecycleNet: Intelligent waste sorting using deep neural networks. In Proceeding of 2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA, 2018, 1–7.

[17] Yujie H., Qinyue G., Maguo S. Trash classification using convolutional neural networks project category: Computer vision. CS230: Deep Learning, 2020. [Online] Available from: http://cs230.Stanford.edu/projects_spring_2020/reports/38847029.pdf [Accessed 21 July 2023].

[18] Ziouzios, D., Tsiktsiris, D., Baras, N., Dasygenis, M. A distributed architecture for smart recycling using machine learning. Future Internet, 2020, 12(9).

[19] Mao, W.L., Chen, W., Wang, C., Lin, Y. Recycling waste classification using optimized convolutional neural network. Resources, Conservation and Recycling, 2021, 164, 105132.

[20] Zhang, Q., Yang, Q., Zhang, X., Bao, Q., Su, J., Liu, X. Waste image classification based on transfer learning and convolutional neural network. Waste Management, 2021, 135, 150–157.

[21] Patrizi, A., Gambosi, G., Zanzotto, F.M. Data augmentation using background replacement for automated sorting of littered waste. Journal of Imaging, 2021, 7(8).

[22] Huynh, M.H., Pham-Hoai, P., Tran, A., Nguyen, T. Automated waste sorting using convolutional neural network. In Proceedings of the 7th NAFOSTED Conference on Information and Computer Science, NICS, 2020, 102–107.

[23] Majchrowska, S., Miko ajczyk, A., Ferlin, M., Klawikowska, Z., Plantykow, M., Kwasigroch, A., Majek, K. Deep learning-based waste detection in natural and urban environments. Waste Management, 2022, 138, 274–284.

[24] Karthikeyan, S., Sivakumar, M., Jeysiva, A., Maheshkumar, C. Application of deep learning for solid waste trash classification using deep CNN. In Proceedings of ACM/CSI/IEEECS Research & Industry Symposium on IoT Cloud for Societal Applications, 2021, 126–131.

[25] Bobulski, J., Kubanek, M. Deep learning for plastic waste classification system. Applied Computational Intelligence and Soft Computing, 2021, 1–7.

[26] Chazhoor, A.A.P., Edmond, H., Gao, B., Woo, W. Deep transfer learning benchmark for plastic waste classification. Intelligence & Robotics, 2022, 2, 1–19.

[27] Baker, N., Szabo-Muller, P., Handmann, U. Transfer learning-based method for automated e-waste recycling in smart cities. EAI Endorsed Transactions on Smart Cities, 2018, 5(16), 169337.

[28] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 4510–4520.

[29] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 2261–2269.